Syntactic and Referential Choice in Corpus-based Generation: Modeling Source, Context and Interactions

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Philosophie (Dr. phil.) genehmigte Abhandlung

Vorgelegt von

Sina Zarrieß

aus Stendal

Hauptberichter:	Prof. Dr. Jonas Kuhn
Mitberichter:	Prof. Dr. Sebastian Padó
Mitberichter:	Prof. Dr. Alexander Koller

Tag der mündlichen Prüfung: 21.05.2015

Institut für maschinelle Sprachverarbeitung
Universität Stuttgart

2016

Erklärung

Hiermit erkläre ich, dass ich, unter Verwendung der im Literaturverzeichnis aufgeführten Quellen und unter fachlicher Betreuung, diese Dissertation selbstständig verfasst habe.

(Sina Zarrieß)

Contents

Deutsche Zusammenfassung

Natürlich-sprachliche Sätze aus einer abstrakten Repräsentation einer kommunikativen Absicht zu generieren, ist ein Prozess, der einer gewissen Variabilität unterliegt, was bedeutet, dass typischerweise mehrere sprachliche Ausdrucksmöglichkeiten für einen nicht-sprachlichen Fakt verfügbar sind. Diese Variabilität liegt auf allen Ebenen der sprachlichen Realisierung vor, zum Beispiel in der Satzstruktur, in lexikalischen Entscheidungen oder der Wortstellung, und viele dieser Realisierungsmöglichkeiten interagieren.

Aus der Perspektive des Sprachgebrauchs erfüllen Phänomene wie Wortstellungsvarianten eine Funktion: sie dienen dazu, eine sprachliche Äußerung an ihren Kontext anzupassen. Beispielsweise würden menschliche Sprecher, wenn sie eine Frage wie in Beispiel (1) beantworten, stark dazu tendieren, die Antwort (1-b) der Antwort (1-a) vorzuziehen. Beide Sätze drücken den gleichen Inhalt aus, aber die sprachliche Realisierung des Inhalts ist im Kontext der vorhergehenden Frage in Satz (1-b) wesentlich angemessener.

(1) Was ist die Funktion eines Lysosoms?

   a. Polymere werden in den Lysosomen von eukaryotischen Zellen verdaut.
   b. Die Funktion eines Lysosoms ist die interzellulare Verdauung von Polymeren in einer eukaryotischen Zelle.

Diese Doktorarbeit untersucht statistische Modelle, die ein Ranking zwischen verschiedenen Realisierungsmöglichkeiten einer Generierungseingabe im Hinblick auf ihre Adäquatheit im Diskurskontext vorhersagen. Wir übernehmen dazu bestimmte Annahmen und Methoden aus dem Paradigma der korpusbasierten Generierung: die Modelle benutzen tatsächlich vorkommende Korpussätze als Instanzen sprachlicher Realisierungsvarianten und die vorhergehenden Sätze als ihren Kontext. Wir setzen Analysewerkzeuge wie Grammatiken und Parser ein, um eine abstrakte Repräsentation eines Satzes

zu bestimmen. Diese Repräsentation stellt den Ausgangspunkt für den Generierungsprozess dar. Das Generierungssystem bildet die Ausgangsrepräsentation auf eine Kandidatenmenge von Realisierungen ab und gewichtet diese mit Hilfe von Merkmalen, die aus dem Kontext berechnet werden. Die Ausgabe des Generierungssytems ist der am besten bewertete Satz, der gegen den originalen Korpussatz evaluiert werden kann.

Diese Arbeit geht von zwei etablierten Teilgebieten der Generierung aus, nämlich der Oberflächenrealisierung und der Generierung für referierende Ausdrücke (REG). In beiden Fällen bekommt der Generator typischerweise eine recht detaillierte Eingaberepräsentation, die bereits einige Generierungsentscheidungen vorwegnimmt. Dabei besteht die Aufgabe der Oberflächenrealisierung darin, aus einer syntaktischen Repräsentation, die grammatische Funktionen und lexikalische Realisierung von Konstituenten aber nicht ihre lineare Reihenfolge spezifiziert, ein Satz zu generieren. Generierung für referierende Ausdrück konzentriert sich auf das Problem, die lexikalische Realisierung von referierenden Nominalphrasen zu bestimmen, zum Beispiel zu entscheiden, ob das Subjekt in einem Satz als Pronomen oder definites Nomen realisiert wird.

Unsere ersten Experimente greifen auf F-Strukturen, wie sie in der Lexikalisch-Funktionalen Grammatik (LFG) definiert werden, und einen grammatik-basierten Generator zurück, der verschiedene Realisierungen, hauptsächlich Wortstellungsvarianten, aus der F-Struktur eines Korpussatzes produziert. We trainieren ein statistisches Rankingmodell, das alternative Konstituentenfolgen eines Satzes bewertet und den Kontext mit Hilfe von Merkmalen wie den vorherigen Erwähnungen der Konstituente, ihrer syntaktische Funktion etc. repräsentiert. Außerdem führen wir ein REG-Experiment durch, bei dem alle referierenden Ausdrücke, die auf denselben Referenten in einem Korpustext verweisen, extrahiert und als Kandidaten für das Ranking betrachtet werden. We trainieren ein Modell, das lernt, verschiedene Realisierungen eines referierenden Ausdrucks den Erwähnungen eines Referenten im Text zuzuweisen. In beiden Experimenten stellt sich heraus, dass Kontextfaktoren, die aus den vorhergehenden Sätzen extrahiert werden, mit jenen Merkmalen interagieren, die aus der detaillierten Repräsentation der Generierungseingabe bestimmt werden. Somit ist die morpho-syntaktische and lexikalische Realisierung von Konstituenten (wie zum Beispiel ihre Definitheit) sehr aussagekräftig für Entscheidungen auf Ebene der Wortstellung, während die syntaktische Position der Erwähnung eines Referenten sehr akkurate Kontextmerkmale für die Generierung der referierenden Ausdrücke liefert.

Aus diesen Beobachtungen ergibt sich die Frage, wie Kontext in einem Generierungssytem modelliert werden kann, das eine abstraktere Repräsentation als Eingabe bekommt, die weniger Entscheidungen vordefiniert. In diesem Fall produziert der Generator eine größere Menge von Realisierungsvarianten und das Kontextmodell kann auf weniger detaillierte Informationen aus der Eingabererpäsentation zurückgreifen.

Um diese Frage zu untersuchen, erweitern wir den LFG-basierten Ansatz der Oberflächenrealisierung für eine semantische Eingaberepräsentation, die Alternationen des Genus verbi and der Wortstellung als Realisierungsvarianten erlaubt. In einer Pilotstudie gehen wir von einem eher naiven Verfahren aus, um F-Strukturen auf semantische Repräsentationen abzubilden, bei dem im Wesentlichen die grammatischen Funktionen normalisiert werden. Es zeigt sich, dass die daraus resultierende Repräsentation für die Generierung oft nicht geeignet ist, um eine erweiterte Kandidatenmenge für das Ranking zu erzeugen, da sie eine Reihe von kontextuellen, syntaktischen und lexikalischen Merkmalen spezifiziert, welche die wohlgeformte Realisierung der Alternation blockieren. Im Besonderen zeigt sich hier das Problem, dass die Repäsentation nicht die impliziten Erwähnungen von Referenten in einem Text erfässt, und damit nicht verwendet werden kann, um Kandidatenmengen zu erzeugen, auf denen das Rankingmodell kontextuelle Präferenzen für das Passiv lernen kann. Wir implementieren eine regelbasiertes Verfahren, das kontextuelle Merkmale mit Hilfe von Heuristiken aus der Eingaberepräsentation entfernt und diese außerdem mit bestimmten Typen von impliziten Referenten anreichert. Unsere Experimente zeigen, dass aus einer solchen Eingaberepräsentation für die Generierung bessere Kandidatenmengen erzeugt werden können, um kombinierte Realisierungsvarianten von Genus verbi und Wortstellung zu lernen und vorherzusagen.

Schließlich entwickeln wir ein datengetriebenes, dependenzbasiertes Szenario, in dem bestehende Ansätze der Oberflächenrealisierung und Generierung referierender Ausdrücke zusammengeführt werden. Die grundlegende Idee dieses Ansatzes ist es, eine Generierungsaufgabe zu definieren, die es erlaubt, systematisch die komplexen Interaktionen zwischen syntaktischen, morpho-syntaktischen und lexikalischen Realisierungsvarianten zu untersuchen. Wir erzeugen einen Datensatz, der manuelle Annotationen von expliziten und impliziten Erwähnungen bestimmter Referenten eines Textes mit dependenzbasierten Repräsentationen kombiniert, die von der morpho-syntaktischen Realisierungen von Prädikaten abstrahieren. Wir entwickeln eine flexible, modulare Generierungsarchitektur, die eine variable Organisation der einzel-

nen Module ermöglicht, wobei die Kontextmodelle der einzelnen Komponenten von den Informationen und Generierungsentscheidungen abhängen, die von den vorhergehenden Modulen der Pipeline-Architektur bestimmt werden. Wir finden heraus, dass syntaktische und referentielle Realisierungsmöglichkeiten eng zusammenhängen, indem wir zeigen, dass die Akkuratheit einzelner Kontextmodelle stark von Fehlern interagierender Module beeinflusst wird. Wir stellen eine sogenannte revisionsbasierte Architektur vor, die Interaktionen zwischen Wortstellung und referierenden Ausdrücken explizit modelliert und dabei bessere Ergebnisse als eine typische Pipeline-Architektur erzielt.

Zusammmenfassed zeigt diese Doktorarbeit, dass das Phänomen der interagierenden Realisierungsmöglichkeiten alle Dimensionen eines korpus-basierten Generierungsverfahrens betrifft: die Definition and das Ableiten einer Eingaberepräsentation, die angemessene Kandidatenmengen der zugrundeliegenden Variabilität erfassen muss, der Aufbau einer Generierungsarchitektur, die Abhängigkeiten zwischen einzelnen Rankingmodellen festlegt, und die Merkmale der Rankingmodelle, die verschiedene Aspekte des Kontextes repräsentieren. Es ist der Forschungsbeitrag dieser Arbeit, die korpusbasierten Verfahren für Oberflächenrealisierung und referierende Ausdrücke für eine breitere Menge an Realisierungsvarianten zu erweitern und dabei die in der Eingaberepräsentation verfügbaren Informationen und die entsprechenden Kandidatenmengen systematisch zu manipulieren. Mit Hilfe dieser Methodologie können wir eine detaillierte Untersuchung von interagierenden Realisierungsvarianten vorlegen.

Abstract

The process of generating natural language sentences from an abstract representation of some communicative intent involves choice, meaning that there will usually be several linguistic means of expression to realize a non-linguistic fact. Choices exists on all levels of the linguistic realization, e.g. sentence structure, predicate-argument structure or word order, and many of these choices interact.

From the perspective of language use, choice phenomena like word order variation fullfil a function: they serve to adapt linguistic utterances to their context. For instance, when answering a question like (2), human speakers would strongly prefer answer (2-b) over (2-a). Both sentences express the same content, but the linguistic realization of the content in (2-b) is much more appropriate in the context of the preceding question.

(2) What is the function of a lysosome?
   a. Polymers are digested in the lysosomes of eukaryotic cells.
   b. The function of a lysosome is intercellular digestion of polymers in a eukaryotic cell.

This thesis investigates statistical models that predict choice by ranking alternative realizations of a generation input according to their naturalness in a particular discourse context. We adopt some basic assumptions and methods from the corpus-based regeneration paradigm: the models are built using naturally occuring corpus sentences as instantiations of linguistic choices and surrounding sentences as their context. We exploit analysis tools such as parsers and grammars to derive an abstract representation of the sentence. This representation constitutes the underlying source of the generation process. The generator maps the source to a set of alternative surface realizations and ranks them based on a set of features computed from the context. The final generation output is the top ranked sentence that can be

evaluated against the original corpus sentence.

We start out from two established tasks in the field, namely surface realization and referring expression generation in context (REG). In both cases, the generator typically gets a relatively detailed input representation that predetermines a range of generation decisions. A surface realisation system has to generate a sentence from a syntactic representation that defines grammatical functions and lexical choice for constituents but not their linear order. REG focuses on the problem of determining the lexical realization of referring noun phrases, for instance, on deciding whether the subject of sentence should be realized as a pronoun or definite noun.

Our first experiments rely on F-structures defined by Lexical Functional Grammar (LFG) as generation inputs and a grammar-based generator that produces surface realizations, i.e. mainly word order variants, from the F-strucure of a corpus sentence. We train a statistical ranker that scores alternative orders of constituents in a sentence and represents context in terms of properties such as previous mentions of the constituent, its syntactic function etc. We also carry out an REG experiment where alternative realizations of noun phrases referring to a discourse referent, e.g. pronouns, definite and indefinite nominals that mention a particular person, are extracted from a text. We train a model that learns to assign realizations of referring expressions to entity mentions in a corpus text. In both settings, we find that contextual factors extracted from the previous sentences (e.g. previous mentions, distance to previous mention) are closely interacting with features extracted from the fine-grained representation of the generation input. Thus, the morpho-syntactic and lexical realization of constituents, such as their definiteness, are highly predictive for decisions at the level of word order while the syntactic position of a mention of a discourse entity in its sentence-internal context provides highly accurate contextual factors for REG.

The observations raise the question how context can be modelled in a generation system that gets a more abstract input representation where less decisions are predetermined. In this case, the generator will produce a bigger set of realization candidates and the context model has access to less detailed information in the input representation.

For being able to investigate this question, we extend the LFG-based surface realization framework to a more abstract semantic representation that triggers choice in terms of voice alternations and word order variation. In a pilot study, we adopt a rather naive way of mapping F-structures to semantics by mainly normalizing syntactic functions. We find that the resulting

generation source is often not appropriate for obtaining an extended candidate set in the generator output as the representation specifies a range of contextual, syntactic or lexical, cues that block a well-formed realization of an alternation. In particular, we encounter the problem that the representation that does not account for implicit mentions of referents in a text, cannot be used to generate meaningful candidate sets for learning contextual preferences for passives. We implement a rule-based approach that heuristically removes contextual cues from generation inputs and enriches representations with certain types of implicit referents. Our ranking experiments show that these representation constitute a more useful source for learning to predict combined choices of voice and word order.

Finally, we develop a data-driven, dependency-based scenario for integrating previous approaches to surface realization and REG. The main idea of this setting is to define a generation task where the intricate interactions between syntactic and referential choice can be systematically studied. We create a data set with manual annotations of mentions of referents, including implicit mentions, and abstract dependency-based representations for surface realizations. We develop a flexible, modular generation architecture that allows for a variable organization of modules where the context models corresponding to the single components vary with the amount of information determined by the previous modules in a pipeline architecture. We find that syntactic and referential choices interact closely by showing that the accuracy of the single contextual models is severely affected by mistakes made by interacting modules. We present a revision-based generation architecture that explicitly models interaction between word order and referential choice, outperforming a standard pipeline.

More generally, this thesis shows that the phenomenon of interacting choices affects all dimensions of a corpus-based generation framework: the definition and derivation of an input representation, which has to trigger appropriate candidate sets of some underlying choice, the set-up of a generation architecture, which determines dependencies between contextual models, and the feature design of a ranking models, which represent different aspect of the context. Our contribution is to systematically extend the corpus-based frameworks for surface realization and REG to a wider range of choices while controlling the information available in the generation input and the corresponding candidate sets. With the help of this methodology, we provide a detailed investigation of the interaction between choices.

List of Tables

List of Figures

Acknowledgements

A huge thank you to my *Doktorvater* Jonas Kuhn. He provided an endless list of positive contextual factors for this thesis: enthusiasm, visions and big pictures, ever encouraging feedback, always motivating and inspiring discussions. He immediately found these interesting bits in the most rudimentary ideas, gave me confidence in my work, gave me freedom, wrote amazingly smooth introductions to some of my papers, took me to South Germany, seemed to truly believe in *Vereinbarkeit*, gave me mental and practical support in some important real-life situations (like Raki and all of its long-term consequences). All this made my life as a PhD student as painless and, at the same time, insightful as it can be.

I was really lucky to work with Aoife Cahill in our SFB project. She did not only share the historically rich Pargram office with me, but many details and insights about the XLE generator, surface realization with LFG as well as extremely useful scripts and data. This thesis benefited a lot from her knowledge and experience.

The IMS and the SFB 732 have been a great environment for doing a PhD. It is a lot of fun to work in such a big group of researchers that pursue so many different and interesting topics, and are always open for discussion and exchange. In particular, Wolfgang Seeker, Anders Bjoerkelund, Kyle Richardson, Florian Laws, Jagoda Bruni, Ozlem Cetinoglu, Arndt Riester and Bernd Bohnet have been great friends and colleagues. Thanks for always being there to talk and the relaxed atmosphere around the coffee machine!

Finally, I have to thank some people that have cared so much for me that it is almost impossible to thank them. Danke, Mama und Papa, zum Beispiel dafür, dass ihr wolltet, dass ich etwas richtig Tolles (und nicht einfach Lehramt) studiere. Danke Matthias, dass du immer die richtigen Dinge, manchmal ein Zauberwort und manchmal auch gar nichts, sagen kannst.

The work reported in this dissertation was conducted in SFB 732 'Incremental Specification in Context', project D2 'Combining contextual information sources for disambiguation in parsing and choice in generation' supported by the Deutsche Forschungsgemeinschaft (DFG).

Chapter 1

Introduction

1.1 Motivation

People use natural language to communicate ideas. The production of a linguistic utterance can be seen as a conversion process that translates an idea into a sequence of words or sounds (Bock, 1987). Computational systems that implement such a translation process are called Natural Language Generation (NLG) systems. While it is impossible to come up with a symbolic representation for the ideas in people's heads which could be empirically evaluated, an NLG system takes an abstract representation of an idea, some non-linguistic content, as input and maps it to a linguistic sentence, text or speech signal (Reiter and Dale, 1997).

In a range of established NLG applications, the abstract input is retrieved from a database that contains knowledge about entities and events. Examples are illustrated in (1-a) and (2-a), showing sets of triples taken from a database that represents facts from a biology textbook (cf. Banik et al. (2013)). This representation defines entities (e.g. `Polymer`), events (e.g. `Intracellular-Digestion`) and relations between them (e.g. `base`). Sentences (1-b) and (2-b) correspond to the desired generation output.

(1) a. Input_1:

```
( (|Intracellular-Digestion04| |object| |Polymer20|)
(|Intracellular-Digestion04| |base| |Eukaryotic-Cell03|)
(|Intracellular-Digestion04| |site| |Lysosome02|)
(|Eukaryotic-Cell03| |has-part| |Lysosome02|))
```

b. Output_1:

Polymers are digested in the lysosomes of eukaryotic cells.

(2) a. Input$_2$:
```
(|Intracellular-Digestion54| |object| |Nucleic-Acid51|)
(|Intracellular-Digestion54| |base| |Eukaryotic-Cell70|)
(|Intracellular-Digestion54| |agent| |Lysosome60|)
(|Lysosome60| |has-function| |Intracellular-Digestion54|))
```
b. Output$_2$:
The function of a lysosome is intercellular digestion of nucleic acid in a eukaryotic cell.

In order to automatically compute a mapping between formal input and linguistic output, many NLG systems divide the problem into several subtasks and make use of intermediate syntactic and semantic representations. In these stages, an overall sentence structure is planned and aggregated, words are chosen, a syntactic structure is computed, the order of constituents is determined. These modules often have a reverse counterpart in Natural Language Understanding (NLU) applications that take a linguistic sentence as input and map it to some the syntactic and semantic structure.

A key challenge for many NLU applications is the ambiguity of natural language, meaning that linguistic sentences often have several syntactic and semantic interpretations. This problem has an equally challenging counterpart in NLG: choice in natural language production, the fact that an abstract content can be phrased by several linguistic outputs.

The two input-output pairs in Example (1) and (2) nicely illustrate that there can be substantial variation in the mapping between abstract input and linguistic output. Although both inputs express ideas about the same type of event (an `Intracellular-Digestion`), the corresponding sentences are structured and lexicalized in different ways. In the first output, the event is lexicalized as the verb *digest* in passive voice which forms the main predicate of the sentence. In the second output, the main verb of the sentence is the copula and the event is expressed via a nominalization. In the first output, the `base` relation is not overtly realized, as the entity `Eukaryotic-Cell` is also the argument of a `has-part` relation. In the second output, the `base` relation is realized via a prepositional phrase headed by *in*.

Thus, when developing NLG systems, we cannot expect to find deterministic and unique mappings between relations and entities in a formal representation on the one and words and linguistic structures on the other hand. In all generation stages that we have mentioned above, certain parts

of the formal representation can be translated to several possible linguistic realizations, i.e. lexical words, syntactic structures, constituent orders, etc. This means that every type of generation process can be conceived of as a procedure that subsumes at least two conceptual steps: the determination of possible candidate realizations that correspond to a formal input and the selection of the optimal linguistic realization.

Looking at practical implementations of NLG systems and how they deal with the determination and selection of possible realizations, we find a whole spectrum of different approaches: On one end of the scale, there are so-called template-based systems that map each generation input to a piece of prefabricated text. In this case, determination and selection of candidates is done manually by the developer. He has to think about all possible use cases of his system, and fixes the output to a constrained range of sentences. On the other side of the spectrum, we find systems that are designed independently of a domain or use case. Some of them use large grammars to first determine all grammatically possible candidates for a given generation input. In a second step, they apply (mostly) statistical ranking models to select the optimal candidate from a set of given realizations. Such systems adopt a *generate-and-rank* architecture that directly implements the two-step procedure for determining and selecting candidates. The first well-known generator of this type was proposed by Langkilde and Knight (1998).

A key advantage of generate-and-rank systems (and other comparable approaches) over template-based systems is that they are able to produce varied output. Whereas a template-based system will always output the same text if its system is in a particular state, more complex generation systems try to adapt their output to a whole range of context factors.

Why is it desirable for a generation system to produce varied output? Looking again at Examples (1) and (2), we could apply the same generation steps from Example (2) to Example (1) and produce the following, competing generation candidates for the input in (1-a):

(3) a. Polymers are digested in the lysosomes of eukaryotic cells.
b. The function of a lysosome is intercellular digestion of polymers in a eukaryotic cell.

Sentences (3-a-b) are both grammatical and express the same factual content. However, native speakers of English would use them in very different contexts. Sentence (3-a) seems to express a fact about *polymers* and is appro-

priate in a context that has introduced this entity beforehand. In contrast, the topic of Sentence (3-b) is *lysosome* and it would not sound natural in a paragraph that puts emphasis on *polymers*:

(4) a. Let's talk about polymers today. Polymers are digested in the lysosomes of eukaryotic cells.
b. ?Let's talk about polymers today. The function of a lysosome is intercellular digestion of polymers in a eukaryotic cell.

Note that what we have informally called "topic" or "aboutness" can be related to concrete linguistic structures in the generation candidates. Due to the passive realization of *digest* in Sentence (3-a), the argument of the `object` relation (*polymers*) is realized sentence-initially as the syntactic subject. In Sentence (3-b), where *digest* is nominalized, the object is realized as a prepositional phrase following the predicate such that another entity has more emphasis in the sentence.

These observations show that generation systems should produce varied output and consider several linguistic choices for expressing some content in order to be able to adapt candidate selection to the context, modeling some notion of contextual appropriateness or naturalness. In the following Chapters of this thesis, we will be much more concrete about what constitutes this context and how we can represent it. For the moment, we focus on our main point, the relation between choice and context in generation:

- Generation involves choice, several linguistic realizations can be used to express some abstract idea.
- Choice affects various levels of linguistic realization, e.g. sentence structure, predicate-argument structure or word order
- Choices of one kind, affecting some part of the underlying content, e.g. the passivization or nominalization of a verbal predicate, are likely to interact with other choices such as linearization.
- Depending on the selected combination of choices realized in a generation output, it is more or less appropriate in a particular context.

This thesis investigates statistical models that predict choice by ranking alternative realizations of a generation input according to their naturalness in a particular discourse context. The key challenge that we address is to design

generation frameworks that capture multiple, possibly interacting choices for phrasing a certain content.

1.2 State of the Art

In the previous Section 1.1, we have introduced the fields of NLG and NLU as counterparts of each other. While the input to the NLU process are invariably surface utterances, which provides a natural basis for comparison across different approaches and application contexts, the input to NLG could by any kind of abstract representation of some content. This complementary input-output relation has far-reaching consequences for the research carried out in the two fields: It has often been said that the NLG community is smaller and has developed at a slower pace as the inputs of practical NLG systems are very diverse and generally harder to obtain (McDonald, 1993; Dale et al., 1998). Depending on whether a system generates a weather report from some sensor data or a description of an object located in a museum and described in a database, it has to account for quite different translation processes between an external reality and its abstract representation on the one hand, and its linguistic description on the other hand.

One possible response to the input problem in NLG is to use representations as inputs that are not coming from a concrete real-world application, but have the status of a linguistic annotation, similar to annotations used as outputs in NLU. This method has become more and more popular, especially with the rise of statistical techniques in the NLG community. This trend was basically initiated by Langkilde and Knight (1998)'s generate-and-rank system mentioned above. In this work, the authors use a computational grammar to generate sentences from a meaning representation. A statistical ranker scores these candidates based on n-gram probabilities calculated on corpus data.

Figure 1.1 displays the idea of a generate-and-rank system in a corpus-based setting. It shows that the actual generation step is split into two stages: a candidate generation and a ranking step. Another important component of the setting is the analysis process which derives generation inputs from sentences occurring in a corpus. These sentence can also be used for evaluation by comparing the generated output against the original source of the meaning analysis.

A concrete input-output pair from Langkilde and Knight (1998)'s gen-

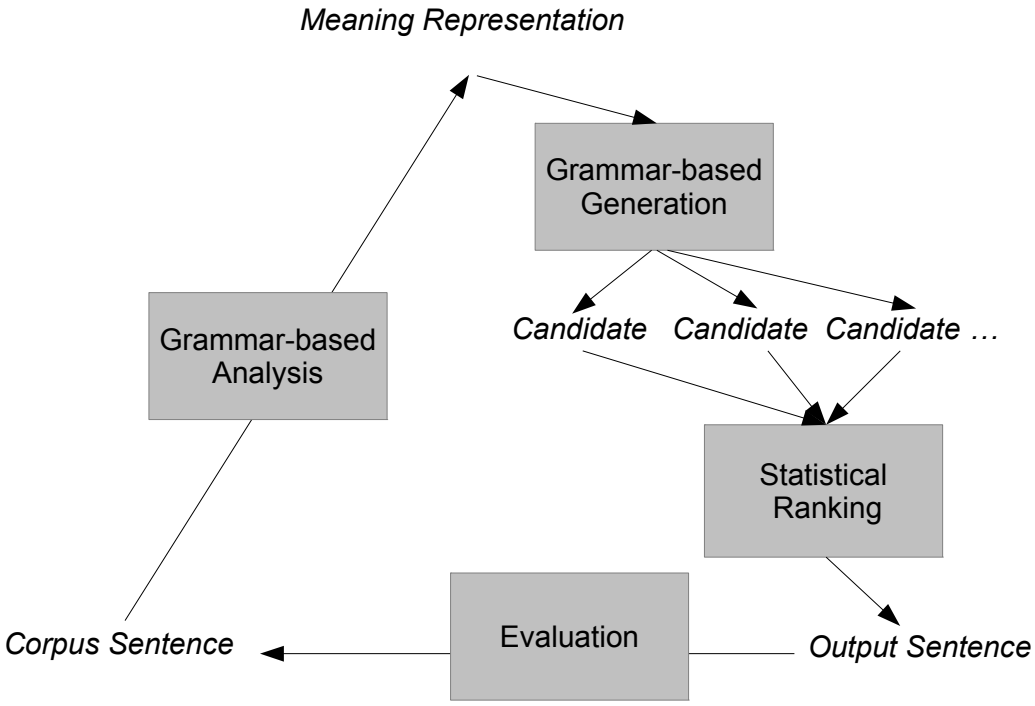


Figure 1.1: The general set-up for a corpus-based generate-and-rank architecture

erator is given in Figure 1.2. Compared to the database input discussed in Section 1.1, the input representation specifies the abstract content in a linguistically structured way: a verbal predicate is the head and its arguments are specified as semantic roles. In contrast to Examples (1) and (2), the generator does not need to decide about the basic sentence structure. Nevertheless, the system generates a considerable number of output candidates, which are efficiently represented in a word lattice.

While Langkilde and Knight (1998) used simple n-gram statistics to rank their generation candidates, a whole range of more elaborate and linguistically informed models for candidate selection have been developed since (Corston-Oliver et al., 2002; Velldal and Oepen, 2005; Cahill et al., 2007a; de Kok, 2010). In these surface realizers, the production of generation candidates is typically assumed to be given through an underlying grammar. These grammars are typically able to produce high-quality, grammatical output which means that they output sets of meaning-equivalent, syntactically well-formed paraphrases or candidates for an abstract input. For instance, Cahill and Riester (2009) use a large and reversible broad-coverage grammar for German to produce mainly word order variants from an syntactic representations annotated on a German treebank. A typical candidate set in their data set looks as follows:

(5) a. Man hat aus der Vergangenheitsaufarbeitung gelernt.
One has from the dealing with the past learned.
b. Aus der Vergangenheitsaufarbeitung hat man gelernt.
c. Aus der Vergangenheitsaufarbeitung gelernt hat man.
d. Gelernt hat man aus der Vergangenheitsaufarbeitung.
e. Gelernt hat aus der Vergangenheitsaufarbeitung man.

Thus, in cases like Example (5), the choice that the generation process has to deal with is constrained to one particular linguistic phenomenon or processing stage such as word order. Note that despite the fact that the inputs specify the major linguistic structure and the lexical realization of the sentences, candidate sets can be large especially when the language allows free worder like German (see e.g. Cahill et al. (2007a)). The focus of these linguistically-informed approaches to surface realization lies on the selection phase of the generation process: the aim is to develop statistical models that capture relative naturalness among generation candidates in a given context.

The key question that is addressed in these approaches is why humans

Input:

```
(m7 / |eat,take in|
    : time present
    :agent (d / |dog,canid|
        : quant plural)
    :patient (b / |os,bone|
        : quant sing))
```

Output:

```
(S (or (seq (or (wrd "the") (wrd "*empty*"))
            (wrd "dog") (wrd "+plural")
            (wrd "may") (wrd "eat")
            (or (wrd "the") (wrd "a")
                (wrd "an") (wrd "*empty*"))
            (wrd "bone") )
       (seq (or (wrd "the") (wrd "a")
                (wrd "an") (wrd "*empty*"))
            (wrd "bone") (wrd "may") (wrd "be")
            (or (wrd "being") (wrd "*empty*"))
            (wrd "eat") (wrd "+pastp") (wrd "by")
            (or (wrd "the") (wrd "*empty*"))
            (wrd "dog") (wrd "+plural")) ) )
(NP (seq (or (wrd "the") (wrd "a")
             (wrd "an") (wrd "*empty*"))
         (wrd "possibility") (wrd "of")
         (or (wrd "the") (wrd "a")
             (wrd "an") (wrd "*empty*"))
         (wrd "consumption") (wrd "of")
         (or (wrd "the") (wrd "a")
             (wrd "an") (wrd "*empty*"))
         (wrd "bone") (wrd "by")
         (or (wrd "the") (wrd "*empty*") )
         (wrd "dog") (wrd "+plural") )) )
(S-GER . . . )
(INF . . . )
```

Figure 1.2: Input-output example from Langkilde and Knight (1998)'s generator

would prefer a certain generation candidate, i.e. a certain way of phrasing or realizing an abstract input, over another in a given context. In state-of-the-art surface realizations models, a multitude of competing, contextual factors that guide candidate selection in generation can be accounted for. Experiments are carried out on relatively large treebanks from the newspaper domain. Thus, if generators are provided with a fairly informed input that specifies the basic syntactic and lexical decisions for the desired output, grammatical output that captures relative naturalness between possible generation candidates can be modeled quite well.

In this respect, surface realization is an interesting empirical test-bed for linguistic theories that aim at modeling choice in human language production. The task focusses on specific phenomena, i.e. syntactic variation, that is known to be sensitive to a range of subtle contextual factors. Finding empirically stable operationalizations of these contextual factors is still a major challenge for research on language use, information structure and discourse.

But, in addition to that theoretical motivation, large-scale, domain-independent surface realization systems could be highly useful as off-the-shelf modules in practical NLG applications – if they provide a suitable modularization of the modeling aspects and are based on representations that can be easily interfaced with. In many real-world NLG domains, where abstract inputs like sensor data have to be processed, available data sets are too small to provide enough clues for fluent, locally coherent NLG. In particular, these are not suitable to train models that are sensitive to the complex interaction of a range of alternation phenomena in language use. As a result, the design principles of many applied NLG systems turn out to be very domain-specific and less flexible than large-scale statistical models that can account for a large number of context factors more naturally. Consequently, these systems do not generalize well to other domains, and produce less varied output. However, contextual variability and flexibility are central properties of a system that generates sentences in order to convey a communicative message in a particular context.

Thus, the research perspectives and approaches change quite fundamentally, when we look at "deeper" generation settings where the input representations are more abstract and contain less hints at the output realization. In Figure 1.3, a meteorological data file which is used as input for weather forecast generation is given. For translating such data into linguistic output, elaborate domain knowledge about meaningful ways of structuring and aggregating relevant pieces of information from the potentially huge set of facts

is needed. Traditionally, such generation problems have been treated in rule-based systems that have the priority to produce coherent, reasonable text from non-linguistic data (see Goldberg et al. (1994) for the first well-known forecast generator).

In many data-to-text generation applications, adaptation to a particular context and variability of the generation output are less central since the overall task involves a range of other complexities. This holds for the linguistic output describing the weather forecast data in Figure 1.3, where single, relatively short and syntactically simple sentences summarizing the most important aspects from the big data file have to produced. As an extreme case, Reiter et al. (2005) even show that an automatic generator which implements the mapping between meteorological data to forecast text in a lexically uniform way is preferred over more varied human-produced text by users. However, this result has to be considered against the specific background of forecast generation where relatively technical and uniformally structured facts have to be communicated. In the more general case, humans seem to have very fine-grained, individual preferences for particular phrasings of a content (Paiva and Evans, 2005; Walker et al., 2007; Dale and Viethen, 2009; Dethlefs et al., 2014).

A major challenge for generators in realistic domains with very abstract input data is to handle requirements and knowledge of various types and from various sources or modules. For instance, generators need to plan and organize content on a textual and pragmatic level while taking into account the possible means for sentence-level linguistic realization (Hovy, 1990; De Smedt et al., 1996). If it is left to the generator to decide about the basic document content, its structure and its style, the possible combinations with syntactic realizations are typically so large that is impossible to compute the entire space of candidates. To get a grasp for the possible variation, Figure 1.4 illustrates two example texts taken from Hovy (1990) describing the same event from two different perspectives. The divergences in selected content, document structure, sentence ordering as well as syntactic realizations are considerable.

Generation systems dealing with such complex tasks typically do not represent candidates exhaustively such as the word lattice output from Figure 1.2, but employ mechanisms that iteratively constrain the search space. The dilemma for these mechanisms is basically the following: If constraints from various sources and modules are handled simultaneously, i.e. in integrated architectures and representations, the systems are hard to maintain, often very

Input, meteorological data file:

```
-------------------------------------------
O1, O2 AND O3 OIL FIELDS (EAST OF SHETLAND)
10-08-01

10/18  WNW  11  13  17  1.7  2.7  NW  1.5  7
10/21  W     8  10  12  1.5  2.4  NW  1.4  7
11/00  W     7   8  10  1.4  2.2  NW  1.4  7
11/03  SW    7   8  10  1.4  2.2  NW  1.3  7
11/06  SW    7   8  10  1.3  2.1  NW  1.3  7
11/09  SSW  10  12  15  1.3  2.1  NW  1.2  7
11/12  S    14  17  21  1.5  2.4  NW  1.2  7
11/15  S    20  25  31  1.8  2.9  WNW 1.3  7
11/18  S    22  27  34  1.9  3.0  SW  1.5  8
11/21  S    24  30  37  2.3  3.7  S   1.7  8
12/00  S    28  35  43  3.0  4.8  S   1.9  8
12/03  S    28  35  43  3.0  4.8  S   1.9  8
12/06  SW   27  33  41  3.0  4.8  S   2.0  8
12/09  WSW  26  32  40  2.9  4.6  SSW 2.0  8
12/12  WSW  25  31  39  2.9  4.6  SW  2.0  8
12/15  WSW  25  31  39  2.9  4.6  WSW 2.0  8
12/18  WSW  24  30  37  3.1  5.0  WSW 2.1  8
12/21  SW   23  28  35  2.9  4.6  WSW 2.2  9
13/00  SW   21  26  32  2.8  4.5  WSW 2.3  9
13/03  SW   19  23  29  2.4  3.8  WSW 2.1  8
13/06  SSW  19  23  29  2.2  3.5  SW  2.0  8
13/09  SSW  20  25  31  2.2  3.5  SSW 1.9  8
13/12  SSW  21  26  32  2.4  3.8  SSW 1.8  8
-------------------------------------------
```

Output, textual weather report:

```
-------------------------------------------
2.FORECAST 1500 GMT FRI 10-Aug,TO 0600GMT SAT
11-Aug 2001
=====WARNINGS:        NIL             =======
WIND(KTS)  CONFIDENCE: HIGH
  10M:     WNW-NW 12-15 BACKING W'LY 05-10 BY
           MIDNIGHT, THEN SW-SSW BY MORNING
  50M:     WNW-NW 15-18 BACKING W'LY 06-12 BY
           MIDNIGHT, THEN SW-SSW BY MORNING
-------------------------------------------
```

Figure 1.3: Input-output example from weather forecast generation due to Belz (2005)

(6) On April 4, students at Yale built a symbolic shantytown to protest their school's investments in companies doing business in South Africa. The college ordered the shanties destroyed. The police arrested 76 protesters when the shantytown was torn down. Local politicians and more than 100 faculty members criticized the action. A week after it had ordered the removal of the shantytown–named Winnie Mandela City, after the South African foe of apartheid–the shantytown was reconstructed and the Administration agreed to allow it to remain standing. Concurrently, Yale announced that its trustees, the Yale Corporation, would soon send a fact-finding mission to South Africa to investigate the actions of corporations in which it owns between $350 million and $400 million of stock.

(7) Some students erected a shantytown to protest Yale's investments in companies that have operations in South Africa. The University tore it down and arrested several of them. The students continued to demonstrate and finally the University said they could put up the shantytown again. The University said it would investigate its investments in South Africa.

Figure 1.4: Generation output examples from Hovy (1990)

domain-specific and the final generation decisions are hard to reconstruct. On the other hand, if different types of constraints are handled completely separately, interactions cannot be captured in a satisfactory way. Our generation example from Section 1.1, repeated here, perfectly illustrates such an interaction:

(8)
```
( (|Intracellular-Digestion36204| |object| |Polymer36220|)
(|Intracellular-Digestion36204| |base| |Eukaryotic-Cell36203|)
(|Intracellular-Digestion36204| |site| |Lysosome36202|)
(|Eukaryotic-Cell36203| |has-part| |Lysosome36202|))
```

a. Polymers are digested in the lysosomes of eukaryotic cells.
b. The function of a lysosome is intercellular digestion of polymers in a eukaryotic cell.

Depending on whether the generation output is supposed to talk about *polymers* or *lysosomes*, the generator should choose different realizations for the `Intracellular-Digestion` event: if it is realized as a verbal passive, *polymers* will figure in a sentence-initial position, if it is nominalized, other constituents can be fronted. Due to such complexities, NLG systems that focus on candidate determination from very abstract representations often focus

on algorithms, architectural issues and resources for generation frameworks.

Roughly speaking, we could say that NLG research falls into two main areas with respect to the underlying simplifying assumptions made such that subproblems of the complex overall task can be addressed systematically: On the one hand, the problem of selecting natural and appropriate candidates based on a detailed account of context is mostly investigated in settings that restrict choice to very specific types. On the other hand, the problem of producing well-formed candidates that describe a particular content in a meaningful way, by combining several, possibly interacting, choice phenomena is mostly investigated in settings where generation inputs are more abstract.

1.3 Research Questions

The major goal of this thesis is to build and investigate generation settings that bridge the gap between contextually informed but very controlled candidate selection on the one hand and complex, highly interactive candidate generation mechanisms from very abstract inputs. Our contribution is to systematically extend the corpus-based surface realization framework to a wider range of choices. In these extended realization frameworks, we provide a detailed investigation of the interaction between choices, building statistical models that predict the preference for a particular grammatical realization of a sentence in a given discourse context.

Our investigations will focus on German, a relatively free word order language. There is a considerable body of work in the generation literature that looks at the automatic prediction of word order. Typically, these approaches assume that the realization of the sentence structure and the lexical realization of its constituents (and referents) is specified in the input. Example (9) illustrates such a case with a simplified generation input in (9-a) and the resulting candidate set, sentences (9-b-e):

(9) a. Input:
`schicken(subject:er, object:Buch, recipient:sie)`

b. Er hat ihr das Buch geschickt.
He has her the book sent.

c. ?Er hat das Buch ihr geschickt.

d. Ihr hat er das Buch geschickt.
e. *Ihr hat das Buch er geschickt.
f. Das Buch hat er ihr geschickt.
g. *Das Buch hat ihr er geschickt.

Although word order in the German *Mittelfeld*, the position between the finite and infinite verb, is often free, there are specific constraints on the precedence of pronouns and nominal realizations of NPs (Uszkoreit, 1987). Consequently, this factor, or interacting choice, can often determine or substantially constrain the set of fluent word orders. Half of the candidate set in (9) would actually be ruled out due to non-natural orders of pronominal and nominal referent realizations. A corpus-based generator that is provided with features concerning the morphological realization of NPs is very likely to predict these patterns as it never or rarely observes them in corpus data.

The task is fundamentally more complex if the generation input does not specify the realization of referents such as in Example (10), where, in addition to the syntactic representation, candidates for referents are given. With these additional choices, all word orders are possible (Sentences (10-b-e)) given a particular instantiation of the referent choice (much more permutations with referring expressions and word orders are possible, but not listed).

(10) a. Input:
`schicken(subject:ref`$_1$`, object:ref`$_2$`, recipient:ref`$_3$`)`
`ref`$_1$`[er, der Junge, ein 15Jähriger]`
`ref`$_2$`[das Buch, es]`
`ref`$_3$`[sie,seiner Freundin]`

b. Er hat ihr das Buch geschickt.
He has her the book sent.
c. Ein Junge hat es seiner Freundin geschickt.
d. Ihr hat er das Buch geschickt.
e. Seiner Freundin hat es der Junge geschickt.
f. Das Buch hat er ihr geschickt.
g. Das Buch hat ihr der Junge geschickt.

In the theoretical linguistic literature, both the referential choice of a pronoun for a referent and its position in the sentence have been correlated with its discoursive prominence. When modeling one or the other choice in isolation, the controlled factor is often a good predictor: if a constituent is

pronominalized, it tends to precede other constituents in the sentence. If a constituent precedes other constituents, it tends to be pronominalized.

Corpus-based generation provides a natural setting for going beyond the controlled study of choice restricted to a particular type and investigating interacting choice phenomena. Given that a quite mature state-of-the-art in corpus-based modeling of controlled choices on the level of syntax and referring expressions has been reached, an obvious next step is to look at generation settings that allow for more flexible manipulations of these types of choices. Instead of generating from a source that abstracts from a single choice, we develop systems that deal with a wider range of candidates generating from a more abstract source.

Our final goal is to be able to empirically test candidate selection mechanisms for combined choices, by extending and putting together existing modules in a corpus-based generation architecture. In applied rule-based generation systems, the standard architecture for combining modules is the pipeline where specific components are devoted to a certain task or phenomenon, which, on the one hand, seems to be a transparent and well-structured set-up for dealing with complex inputs, but, on the other hand, implies that interacting constraints cannot be modeled in a straightforward way. For instance, linearization is typically considered to be the last stage in an NLG pipeline, meaning that all the previous modules, e.g. referring expression generation or syntactic realization, do not have access to linear precedence information. Such architectural decisions seem to be a notorious problem for pipeline systems (Mellish et al., 2000), as e.g. the realization of pronouns depends to a large extent on the constituent's position in the surface sentence.

The dimension of the architecture implemented by a particular selection mechanism is intricately related to the representation and modeling of contextual factors: A generation pipeline that first realizes referring expressions and then deals with syntax and linearization cannot access factors related to linear order and surface syntax in its model for referring expressions. A generator that is provided a surface corpus text where just pronouns and definite descriptions are missing can condition its decisions on a wider range of factors and integrate surface cues in its model.

But before we can look at selection mechanisms and contextual models that deal with multiple, interacting choices produced from a more abstract generation input, we have to construct an appropriate source that will actually yield the desired candidates. Here, it turns out that the limits of existing

methods for corpus-based generation are quickly reached. Despite the fact that the input representation or source is usually recognized as a crucial ingredient of a generation framework (see Section 1.2), there has not been a lot of corpus-based research that systematically looks at effects of manipulating and deriving the source of an underlying choice. Thus, when dealing with choice processes that relate to variation in surface order, the underlying source can be reconstructed by relatively trivial means or by exploiting existing syntactic annotation tools from the NLU domain. For data-driven surface realizers, a common input representation is an unordered constituency or dependency tree, where the order of the nodes, words or phrases has been removed. An even simpler strategy is remove the order of words from a tokenized sentence, such that the source of the choice process is a bag of words (see e.g. Wan et al. (2009))

Issues related to the definition of the input representation for surface realization are also highly relevant when we consider these representations to be the interface that the surface realizer provides for deeper components in a practical generation system. As mentioned above, a typical assumption is that surface realizers should be applicable as off-the-shelf, large-scale components, trained on syntactico-semantic representations that are generally agreed upon and can be easily produced by a smaller-scale, domain-specific system. However, as it turns out, syntactic annotations that are more or less generally agreed upon in the NLU community do not necessarily serve as a reasonable basis in NLG.

The commonly used analysis procedures in corpus-based generation do not always yield the desired abstractions for other surface realization decisions. Consider the dialogue in (11) where Speaker B described the robbery event with a passive construction for *steal* that does not mention an agent:

(11) Speaker A: What happened to your bike?
Speaker B:
a. It was stolen.
b. `steal(?,bike)`

A semantic representation which is automatically derived for the Sentence (11-a) is likely to resemble the structure (11-b) that does not specify the agent of *steal*. Consequently, a generator cannot produce an active sentence for this input as this would require an agent entity. Now, consider a slightly different realization of the sentence where this agent entity is present:

(12) Speaker A: What happened to your bike?
Speaker B:

a. It was stolen by somebody.
b. `steal(somebody,bike)`

Having the relation (12-b) as an input, a generator could produce the active candidate sentence *Somebody has stolen the bike*. This sentence would also be a natural active paraphrase for the passive in (11), but it cannot be generated since the meaning representation does not account for the underlying interaction with argument realization, i.e. it does not hide the fact that the sentence did not realize the agent of the verb as on overt phrase.

Another related example in German is shown in (13), where a pronominal subject of an active verb cannot be turned into an oblique agent of the passive due to morpho-syntactic restrictions exhibited by the arbitrary reference pronoun *man*. If a very similar pronoun like *jemand* is chosen, the passive paraphrase is perfectly well-formed.

(13) Speaker A: Was ist mit deinem Fahrrad passiert?
Speaker A: What is with your bike happened?
Speaker B:

a. Man hat es mir gestohlen.
One has it me stolen.
b. `stehlen(man,Fahrrad)`

c. *Es wurde mir von man gestohlen.
It was me by one stolen.
d. Es wurde mir von jemandem gestohlen.
It was me by somebody stolen.

These examples clearly show that interacting choices constitute a serious issue for deriving generation inputs. If the derivation process is not aware of these potential interactions, the set of realizations considered for candidate selection will be restricted by a range of idiosyncratic, biasing factors. This is a clearly undesirable situation for an empirical study of voice alternations and their contexts.

These examples also point to some undesirable implications for practical NLG applications that would rely on a corpus-based surface realization component to map a syntactic representation produced by a microplanning

component to a surface sentence: Using the above syntactic structures as intermediate representations in the generation process would ultimately mean that the "deeper" modules of the system, concerned with lexicalisation and content selection, would already predetermine most of the contextually relevant, subtle realization choices.

In this thesis, we will show that is essential for a corpus-based generation framework to adress these issues and account for interactions between choices in order to obtain plausible candidate selection models that go beyond word order variation and scale to more complex phenomena like syntactic alternations. We will focus on developing a general account of syntactic and referential choice phenomena in NLG, plausible representations of generation inputs and contextual factors.

From a broader perspective, this thesis investigates and extends the basic foundations of surface realization as a well-defined component of the generation process, which is relevant for practical systems as well. As it stands, previous attempts to integrate existing surface realizers as off-the-shelf tools in external systems or tasks have often been moderately succesful, or have imposed substantial additional manual effort on system developers. Whereas such problems have been mostly treated as technical idiosyncrasies of specific surface realization systems, we show that they shed light on some common, implicit assumptions in surface realisation approaches which need to be rethought theoretically, methodologically and practically.

Generally, the phenomenon of interacting choice affects several dimensions related to the set-up of a generation system:

- the input representation, defining the source of some underlying choice
- the architecture, defining the organization of candidate generation and selection mechanisms
- the context model, representing the factors that account for candidate selection

In this thesis, we will exploit the corpus-based generation paradigm for experimenting with these three dimensions of choice processes in an NLG system. Our approach carefully extends some well-studied frameworks with some additional types of choices. The extensions will be controlled and "careful" in the sense that they try to capture a phenomenon like the passive voice in a range of contexts, being aware of interactions with other choices such

as referring expressions and word order. The main question we will ask is which candidate from a particular candidate set representing a set of choices is preferred in a particular context. However, we model candidate selection by taking into account the following related questions:

- How does information specified in the input representation relate to context factors represented in the feature model?
- Which candidate sets can be obtained from a particular input representation?
- How do different architectures account for interactions between choices in a candidate set?

1.4 Outline

A central methodological aspect of this thesis is that we do not focus on a particular NLG framework, but on general issues related to interacting choices in corpus-based surface realization. Therefore, we investigate a range of different tasks and NLG frameworks that are not commonly combined. Some of our main insights will be gained through implementing, testing, and comparing certain methods *across* different frameworks. Therefore, we have adopted a structure that is deliberately not organised around specific NLG systems, but around high-level aspects of corpus-based NLG that are investigated in a number of settings.

A general overview of all the major NLG systems and tasks, that are adressed in this work, is given in Chapter 2. It starts with a literature review of research on corpus-based generation systems, statistical methods used in state-of-the-art surface realizers, and standard practice and common issues concerning evaluation. The generation frameworks investigated in the subsequent experimental Chapters are introduced in further detail, focusing on the relevant components for obtaining input representations, determining candidate sets and mechanisms for context-aware candidate selection.

In Chapter 3, we will delve deeper into notions of context and why it is desirable for a generator to be able to make choices depending on a particular discourse context. We provide some relevant linguistic background and review related work from NLG research. We also discuss possible synergies between these two fields.

The subsequent chapters which are devoted to empirical studies and results carried out in the respective generation set-ups are organized in terms of the different dimensions of interacting choice in generation, adopting a cross-framework perspective:

In Chapter 4, we start out with standard, separate set-ups for referring expression generation and surface realization. We focus on the feature models used for candidate selection and provide a detailed analysis of different ways of exploiting and representing context in these models. Precisely, we include factors from sentence-external context and analyze their interactions with sentence-internal factors that are specified through other, controlled choice types.

In Chapter 5, we extend the standard set-up of surface realization and REG. We look at ways of manipulating the input representations and the effects on candidate generation. We develop representations that account for implicit mentions of discourse referents in two different settings, creating data sets that will be used and evaluated in the subsequent Chapters. Section 5.1 and 5.3 summarize the extension of a grammar-based surface realization for voice alternations. Section 5.4 describes a combined setting for dependency-based syntactic realization and linearization, integrated with referring expression generation that includes implicit referents.

Chapter 6 discusses some experiments in the extended surface realization set-ups from Chapter 5. We analyze the impact of the generation source on candidate selection difficulty and output quality.

In Chapter 7, we make use of our extended generation inputs and manipulate architectural decisions in our generation set-ups. We focus again on candidate selection models that vary according to certain architectural implementations. We assess interactions between syntactic and referential choices by comparing different generation architectures in flexibly defined frameworks. These result on the combined dependency-based surface realization and REG have been published in Zarrieß and Kuhn (2013). We also report some analyses from the grammar-based generate-and-rank system for voice alternations and some interaction effects that we found in a human evaluation study, both published in Zarrieß et al. (2011).

Conclusions are given in Chapter 8.

Most of the experiments, models and approaches described in this thesis have been reported on in other publications:

- Experiments from Chapter 4.2 and 4.3 have been published in Zarrieß

et al. (2012).

- Chapter 5.1 and 5.3 relate to Zarrieß and Kuhn (2010); Zarrieß et al. (2011).
- The experiment from Chapter 6.2 has been published in Zarrieß et al. (2011).
- Chapter 5.4 and Chapter 7.2 extend on Zarrieß and Kuhn (2013).
- Chapter 7.3 and 7.4.2 extend on Zarrieß et al. (2011).

Chapter 2

Corpus-based Generation

In this thesis, we investigate corpus-based generation as a paradigm for computational modeling of choice processes in natural language. The Introduction has presented the major components of a corpus-based choice model: an abstract source that is derived from corpus sentence and yields a set of generation candidates, a context model that represents factors capturing preferences between particular realizations of a choice, and a selection mechanism that applies the context model to a set of generation candidates.

The fundamental idea of any corpus-based generation system is that the context model can be automatically learned on corpus data. Beyond that, existing corpus-based generators vary widely with respect to the actual implementation of selection mechanisms, candidate generation and input representation. Consequently, a range of techniques for exploiting corpus data, representing contextual factors and applying them to choice candidates have been developed in the field.

This Chapter introduces the generation frameworks that will be used in this thesis to investigate models of sentence-level choices in natural language discourse. It focuses on technical, architectural and methodological aspects of the respective generation set-ups. These aspects provide an important background for the experimental and empirical work presented in subsequent Chapters as they have immediate implications for the choice phenomena that we will be able to model, and the theoretical questions that the underlying models will be suited to answer.

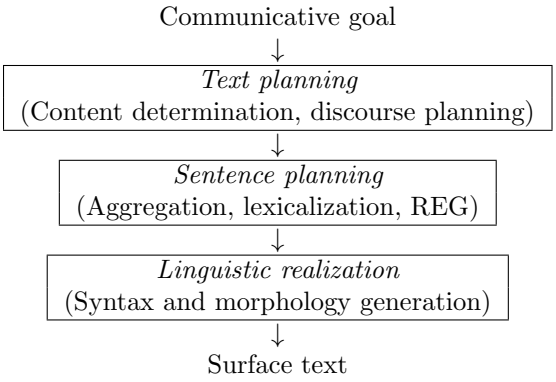


Figure 2.1: Standard NLG pipeline due to Reiter and Dale (1997)

2.1 General Overview

A common assumption in NLG research is that the complex task of translating an abstract input to a linguistic output can be decomposed into a range of components and modules that adress different stages of the translation process. A typical architecture that deals with the multitude of components is the pipeline where discourse-level tasks devoted to content and sentence planning are carried out prior to sentence-level tasks (Reiter, 1994). Figure 2.1 shows such a common NLG pipeline (Reiter and Dale, 1997; Bateman and Zock, 2003), where surface realization is the final step that is clearly separate from "higher level" discourse-oriented tasks.

Based on the pipeline assumption, many statistical generators have been developed for specific components of the generation process. As compared to rule-based methods, corpus-based statistical approaches provide some advantages for engineering NLG systems: they require less manual rule-crafting, tend to be more robust and less domain specific (Oh and Rudnicky, 2002; Belz, 2005). They also offer an interesting alternative to template-based systems that tend to be less flexible and have issues with maintainability (Reiter, 1995; Ratnaparkhi, 2000). The following list gives an overview of existing corpus-based systems according to the sub-tasks they address:

- content planning (Duboue, 2002; Barzilay and Lapata, 2005; Kelly et al., 2009)
- sentence planning and aggregation (Walker et al., 2001; Stent et al., 2004; Barzilay and Lapata, 2006)

- lexical choice (Bangalore and Rambow, 2000a)
- referring expression generation (Siddharthan and Copestake, 2004; Belz and Varges, 2007; Greenbacker and McCoy, 2009)
- surface realization (Langkilde and Knight, 1998; Ratnaparkhi, 2000; Corston-Oliver et al., 2002; Ringger et al., 2004; White, 2004; Velldal and Oepen, 2006; Filippova and Strube, 2007a; Wan et al., 2009; Bohnet et al., 2010)

As can be seen from the above list, statistical methods have been most extensively investigated for surface realization. The main reason for this is that standard resources and tools that have been developed for other NLP domains, such as treebanks, parsers and language models, can be straightforwardly exploited for deriving large amounts of generation inputs (Callaway, 2003). For instance, Langkilde-Geary (2002) automatically constructs inputs for the Nitrogen generator, a successor of Langkilde and Knight (1998), by transforming syntactic representations annotated on the Penn Treebank. Zhong and Stent (2005) use a pipeline of general-purpose syntactic and semantic analysis tools to automatically acquire surface realizers from a set of corpora.

Statistical methods also have some conceptually attractive aspects for surface realization: the task typically involves dealing with linguistic phenomena that are subject to soft constraints. Most importantly, surface realizers have to deal with the prediction of word order, which is known to be sensitive to a range of contextual factors. These factors can be naturally modeled in statistical classification and ranking systems, which will be discussed in detail in the following Chapter 3.

The above mentioned surface realizers exhibit considerable differences with respect to the input representation used as a source for generation, and consequently, with respect to the candidates that they deal with. Thus, the HALogen system by Langkilde and Knight (1998) generates from semantic representations where not all words and syntactic structures are specified, so that their models involve lexical choice, syntactic realization and word ordering. Making use of heavy over-generation in the step that produces candidates, their resulting candidate sets include sentences that are ungrammatical. The systems by Filippova and Strube (2007a) and Bohnet et al. (2010) use a basic type of syntactic structure where all lexical words are

specified. The generation of an ungrammatical sentence is possible but unlikely due the the relatively specific input information. In contrast, Wan et al. (2009) generates surface sentences from bags of words so that his model needs to capture grammatical well-formedness constraints. Finally, in the generate-and-rank realizers presented by Velldal and Oepen (2006) and Cahill et al. (2007a), large-coverage grammars determine the candidate sets of a generation input so that ungrammatical candidates are almost completely excluded. The resulting candidates contain word order paraphrases, but also variation with respect to the morpho-syntactic realization of function words.

It has been noted that the diversity of existing surface realization approaches, which still make system-specific assumptions about the input representation and often require some preprocessing for deriving suitable inputs from general-purpose syntactic representations, calls into question the original motivation of corpus-based techniques as providing more robust and flexible methods (DeVault et al., 2008). Callaway (2003) shows that this preprocessing also requires a significant engineering effort, in order to make the grammar-based generator achieve a satisfactory coverage on corpus-based inputs. A similar finding is reported by Belz et al. (2011). In Chapters 5 and 6, we will come back to the issue of defining corpus-based generation inputs, investigating and discussing it in great detail.

Research on referring expression generation (REG) has been dominated for a long time on rule-based approaches in small domains (Siddharthan and Copestake, 2004; Krahmer and Van Deemter, 2012). The task of generating referring expressions in context (GREC), which is the framework we also adopt for modeling referential choice, been first proprosed by Belz and Varges (2007). Subsequent work on this problem has been mainly carried out in a series of shared tasks (Belz et al., 2008, 2009; Belz and Kow, 2010). The idea of GREC is to treat mentions of referents in a text as slots where the REG system has to insert contextually appropriate surface forms, using the original surface forms from the corpus text as input candidates. This task mainly targets contextual choice in terms of distinguishing e.g. contexts for pronominal vs. definite references to an entity in a text, in contrast to selecting a set of attributes that identifies a particular referent in a communicative scene, which is the classical REG paradigm (Dale, 1992).

Regardless of the specific module they are designed for, statistical generators can be separated into two basic categories with respect to how they actually exploit corpus-based knowledge. The first type, which was introduced in Chapter 1 as a generate-and-rank architecture, can be considered as a hybrid

generator as it employs some rule-based mechanisms that defines the search space or candidate set for a particular generation input. Statistical learning techniques are used to obtain a weighting of the predefined candidates. The second type of statistical generators completely discards manually engineered rules and extracts the candidate set or search space from the corpus data itself. In the following Section 2.1.1, we will give a short overview about the different statistical methods exploited for corpus-based NLG, focussing mainly on surface realization.

In this thesis, we will use generation frameworks from both categories. As an instance of a hybrid generate-and-rank architecture, we will employ the LFG-based system from Cahill et al. (2007a), which is presented in detail in the following Section 2.2. Second, we use a dependency-based surface realizer, described in Bohnet et al. (2012). It employs an algorithm that traverses an undordered dependency-tree and conditions its ordering decisions on a classifier. The system is presented in Section 2.3. Finally, in Section 2.4, we will discuss the corpus-based setting for referring expression generation.

2.1.1 Statistical Models for Realization Problems

Language models The most well-known and influential paradigm in corpus-based NLG is the two-stage generate-and-rank approach, first implemented by Knight and Hatzivassiloglou (1995) and Langkilde and Knight (1998), that apply language models to score candidate surface sentences generated by a rule-based system. It has been adapted to a variety of syntax-based settings, such as in Bangalore and Rambow (2000b) outputs generated with an XTAG grammar are ranked with a trigram language model. Oh and Rudnicky (2002) employ n-gram models in a generative mode, for producing sentences in a spoken dialogue system. They work in the air travel domain where utterances are distinguished into a set of classes and a language model is trained for each class. Zhong and Stent (2005) opt for a general framework for inducing surface realizers on several data sets, incorporating a standard language model ranking of output candidates.

Instead of delaying the output scoring until all possible candidates have been generated, White (2004) uses n-gram probabilities for pruning in chart-based CCG realization. White et al. (2007) implements a factored language model that interpolates word-based language model with a PoS-based and a supertag-based language model. The scoring can be done in a two-stage mode, or in anytime mode, which is useful for application in dialogue systems.

Since language models can be generally be trained on large corpora, and mostly do not presuppose any linguistic annotations, the development costs for generate-and-rank set-ups are typically cheap. Thus, this set-up has also inspired methods from other generation domains. For sentence planning, Stent et al. (2004) implement a two-stage approach that first generates instantiations of sentence plans by randomly combining a set of predefined, linguistic realizations operations, a surface realizer ranks the alternatives. Similarly, Barzilay and Lapata (2008) first generate various randomizations of the order of sentences in a corpus text that constitute the input to a ranking component which predicts the most appropriate sentence order.

Stochastic generation grammars From the perspective of system performance, generate-and-rank architectures do not always offer the most attractive solution, especially when the inputs are very abstract and trigger a lot of candidates that have to be exhaustively generated (Belz, 2005). This has been addressed in a number of works where the translation between generation input and output sentence is directly modeled as a statistical mapping procedure. These approaches require some treebank or data annotated with generation input-output pairs . Ratnaparkhi (2000) trains maximum-entropy classifiers on a corpus of user queries in the air travel domain, annotated with attributes corresponding to a set of generation templates. The models learn to choose and order words for a template-based generation input, his features include n-gram probabilities and dependency information. Belz (2005, 2008) use PCFGs in a data-to-text weather forecast generation task. A manually defined CFG defines the generation space, the rules are weighted probabilistically. She compares a greedy decoding method that selects the most likely generation rule at each choice point and a viterbi decoder that finds the most likely overall derivation sequence of generation rules.

In a similar vein, Marciniak and Strube (2005) generates route-descriptions based on a variant of the TAG formalism, They split the generation task into a set of linguistic realization problems (similar to grammar rules) and train a classifier for each subproblem. In contrast to the standard NLG pipeline architecture sketched above, they propose an ILP formulation of the generation problem for optimizing the combined decisions of the classifiers. More recent approaches have tried to model such statistical mapping procedures without defining an underlying set of grammatical operations (Wong and Mooney, 2007; DeVault et al., 2008; Mairesse et al., 2010; Angeli et al., 2010).

Log-linear Ranking An approach that combines language model-based ranking with the possibility to encode rich feature models trained on annotated data has been first proposed by Velldal and Oepen (2005), for surface realization with large-coverage HPSG grammars. In this architecture, a reversible HPSG grammar is not only used for generating candidates, but for annotating the original corpus sentence with the syntactic analysis and a set of alternative realizations produced for the analysis by the same grammar. A log-linear model learns to rank these paraphrases that correspond to the underlying grammatical representation. While Velldal and Oepen (2005) opts for a combination of morpho-syntactic features and n-gram probabilities, Nakanishi et al. (2005) finds that n-gram probabilities decrease the performance of the ranker. Cahill et al. (2007a) adapt Velldal and Oepen (2005)'s architecture for surface realization with the borad-coverage German LFG, the system is described in Section 2.2.

While the set-up implemented by Velldal and Oepen (2005) and Cahill et al. (2007a) exploits the reversibility of the grammar-based component for data acquisition, de Kok et al. (2011) introduces the framework of Reversible Stochastic Attribute-Value Grammars, where a single maximum-entropy model is used for parse-disambiguation and fluency ranking.

Tree-based Linearization Another way to incorporate fine-grained syntactic features into models for broad-coverage surface realization, is to implement an incremental tree transformation process where a probabilistic model weights ordering decisions at each step. Ringger et al. (2004) incorporates linguistically-informed tree transformation models into the Amalgam surface realizer. Similar to maximum entropy models for ranking, Filippova and Strube (2007a) and Filippova and Strube (2009) use maximum entropy classifiers for ordering constituents in a dependency tree input, separating classification for sentence-initial constituents from the rest of the sentence. They also find a benefit of combining n-gram probabilities and linguistic features. Wan et al. (2009) uses a spanning tree algorithm for inducing most probable dependency structure on a bag of words. Finally, Guo et al. (2011) presents an algorithm for a bottom-up traversal of a dependency or F-structure where dependency n-gram models score possible permutations of realizations at each step. Their approach also models insertion of function words. This procedure is very similar to the dependency-based linearizer by Bohnet et al. (2012), described in Section 2.3.

2.1.2 Evaluation

Even before the rise of statistical methods in NLG research, the evaluation of NLG systems has been a notorious problem (Mellish and Dale, 1998). Langkilde-Geary (2002) first suggested a corpus-based evaluation for surface realization that automatically compared the sentence generated by the system against the original corpus sentence, i.e. the sentence used to derive an input for the generator. Inspired from automatic evaluations of machine translation systems, she used Exact Match, the proportion of entirely identical sentence, and the BLEU score (Papineni et al., 2001) which measures n-gram overlap between the system output and the gold reference in the corpus. This evaluation method is now widely established in corpus-based generation, while at the same time, causing more debates about appropriate ways of assessing the output quality of an NLG system.

The major limitation of automatic evaluation measures for NLG stems from the combination of two facts: On the one hand, these measures have to be based on some reference output, which is, in corpus-based NLG, usually the original corpus sentence or text. On the other hand, it is clear that native speakers often do not expect a single way of realizing a certain content, i.e. they accept a certain amount of variation (Reiter and Sripada, 2002). As a consequence, it is generally agreed that an automatic measure which only counts generated sentences as valid which are identical to the corpus sentence are too strict. The ideal automatic evaluation measure would not penalize acceptable deviations from the original sentence, but only give negative scores to variation that is ungrammatical, or unacceptable for other reasons.

BLEU is basically an average over all the n-grams in the output that match the reference without placing restrictions on the order of n-grams, thus allowing for some deviations from the gold sentence. If BLEU is calculated on several reference sentences, it can also allow for variation in word choice which is especially important for machine translation (Papineni et al., 2001). Although it has been shown that BLEU correlates with human judgements in machine translation and generation applications (Papineni et al., 2001; Reiter and Belz, 2009) and it is frequently used in the communities, problems and shortcomings of the measure are equally well-known. Callison-Burch et al. (2006) argues that BLEU actually allows too much variation giving example calculations for sentences where BLEU would assign equal scores to thousands of permutations of a sentence (i.e. and it is unlikely that humans would also equally like all these permutations). For this reason, a

close relative of BLEU, NIST (Doddington, 2002), has been proposed. In contrast to BLEU where all n-grams are weighted equally, NIST gives more weight to less frequent n-grams of a text. Another frequently discussed shortcoming of the BLEU measure is the fact that mismatches on the n-gram level do not necessarily reflect syntactic violations or ungrammaticalities. Therefore, Owczarzak et al. (2007) have developed a syntactic, dependency-based measure which is, on the one hand, less surface-oriented as n-gram overlap measures, but tries to capture syntactic similarity. Some more well-known automatic measures used as alternatives to BLEU are METEOR (Banerjee and Lavie, 2005), and TER (Snover et al., 2006).

However, despite intensive research on automatic evaluation measures for machine translation and generation, it is not clear whether any of the more sophisticated measures are actually more appropriate to use than BLEU. For instance, Stent et al. (2005) report an evaluation study on a set of 118 English paraphrase sentences that correlates a range of simple and sophisticated measures with two types of human judgments: a) adequacy ("how much of the meaning of the reference is expressed in the output sentence?") and b) fluency ("how do you judge the fluency of the output sentence?"). They find that all measures have a positive, but not very strong, correlation with human adequacy judgements, but no correlation with fluency. Reiter and Belz (2009) correlate human and automatic measures for weather forecast generation and come to the opposite conclusion: They find that none of the automatic measures correlates with human ratings of the content quality, which is similar to Stent et al. (2005)'s adequacy criterion. If systems are controlled such that they generate the same content, the NIST measure achieves the best correlations on the level of fluency. This result for fluency judgements is supported by Cahill (2009) who finds correlations between human judgements and automatic measures collected for surface realizations for German LFG F-structures and a large number of native speakers. She compares two experimental settings: an evaluation where humans where asked to rank several outputs from different systems relative to each other, and an evaluation where speakers where asked to judge the naturalness of a single sentence on a scale from 1 to 5. In the first ranking experiment, the correlations with automatic measures are generally higher than in the second experiment.

Thus, as it turns out, it is also not trivial to evaluate evaluation measures as the resulting correlations with human judgements depend on the method for collecting these judgements. Here again, the crucial observation

is that the degree of acceptable variation is hard to capture. Cahill and Forst (2009) report that, in their human judgement experiments for German surface realization, human judges selected the same string as the original corpus sentence in 70% of the cases. In 5 out of their 41 items, most judges select a string other than the corpus sentence. Similarly, for referring expression realization, Belz and Varges (2007) conducted an experiment where humans where asked to fill in RE slots given a list of NP candidates. They collected 3 human RE choices for 734 RE slots and obtained an absolute agreement of 50.1%, i.e. the number of cases where all 3 humans selected exactly the same RE. For pronominalization, the agreement was 64.9%.

In this thesis, we will mostly use a combination of automatic measures, and interpret results in terms of the respective strengths of the single measures. This has become standard practice in recent shared tasks, see e.g. Belz et al. (2011). If possible, automatic measures should be also coupled with human judgements. In Chapter 7.4, we report on a pilot human evaluation of our extended surface realization set-up, and discuss some intricacies for collecting human judgements for generation tasks going beyond word order variation.

2.2 Realization Ranking in a Reversible LFG-based Architecture

In the following, we present the LFG-based, surface realization ranking system developed by Cahill et al. (2007a). It is the basis for the realization ranking experiment in Chapter 4.2, and the core part of the extended surface realization architecture presented in Chapter 5.1, 6.2, and 7.3.

The system has two components: a) a large-scale hand-crafted LFG for German (Rohrer and Forst, 2006), used to parse and regenerate a corpus sentence, b) a stochastic ranker that selects the most appropriate regenerated sentence in context according to an underlying, linguistically motivated feature model. The system first exhaustively generates the set of sentences that correspond to the grammatically well-formed realizations of a syntactic representation and then models the choice problem in a separate statistical component, being a typical instance of the generate-and-rank framework shown in Figure 1.1.

The grammar serves a core device in the set-up as it specifies the mapping

between an abstract input and its surface candidates, and also defines the abstract input itself. Therefore, Section 2.2.1 will give a short introduction to LFG and some general theoretical motivations that underly the definition of syntactic representations in LFG grammars. Section 2.2.2 describes the candidate generation component, and Section 2.2.3 the ranking component.

2.2.1 Deep and Surface Syntax in LFG

LFG is a constraint-based theory of grammar (Bresnan, 2001). It posits two levels of representation, constituent-structure and functional-structure. C-structure is represented by context-free phrase-structure trees, and captures surface grammatical configurations. F-structures are attribute-value matrices that represent the basic predicate-argument and adjunct structures of a sentence in terms of grammatical relations.

For our surface realization experiments, F-structures will be used as an abstract input representation for the generator. Figure 2.2 illustrates a German sentence and its corresponding, simplified F-structure analysis produced by the German LFG. In the analysis direction, an LFG assigns a C-structure/F-structure pair to a sentence. On the level of C-structure, the surface constituents of a sentence are fully specified. The C-structure for the sentence in Example (1) is given in Figure 2.3.

The relation between C-structure and F-Structure in an LFG analysis is defined via the functional projection ϕ. This function maps every node in the C-structure tree to a node in the F-structure. This mapping is *many-to-one*, such that several C-structure nodes can get assigned to the same F-structure. In Figure 2.2, every node of the F-structure is annotated with a set of numerical identifiers, corresponding to the C-structure nodes in Figure 2.3. For instance, the auxiliary *werden* with the id 4529 and the participle of the main verb *gemacht* with the id 2727 in the C-structure are mapped to the same, namely the top-level node in the F-structure. Intuitively speaking, only lexical verbs such as verbs, nouns, adjectives, etc. will receive a PRED-value in the F-structure. Function words typically yield certain atomic features (e.g. TENSE for auxiliaries).

The general motivation for this relational design of the LFG formalism is that differences and similarities between configurational and non-configurational languages can be captured elegantly. Historically, many linguistic formalisms and theories have been defined for highly hierarchical languages such as English. In these languages, sentences are typically orga-

(1) Diese Gruppe wird für einen Großteil der Gewalttaten
This group is for a major part of the violent acts
verantwortlich gemacht.
responsible made.
'This group is made responsible for a major part of the violent acts.'

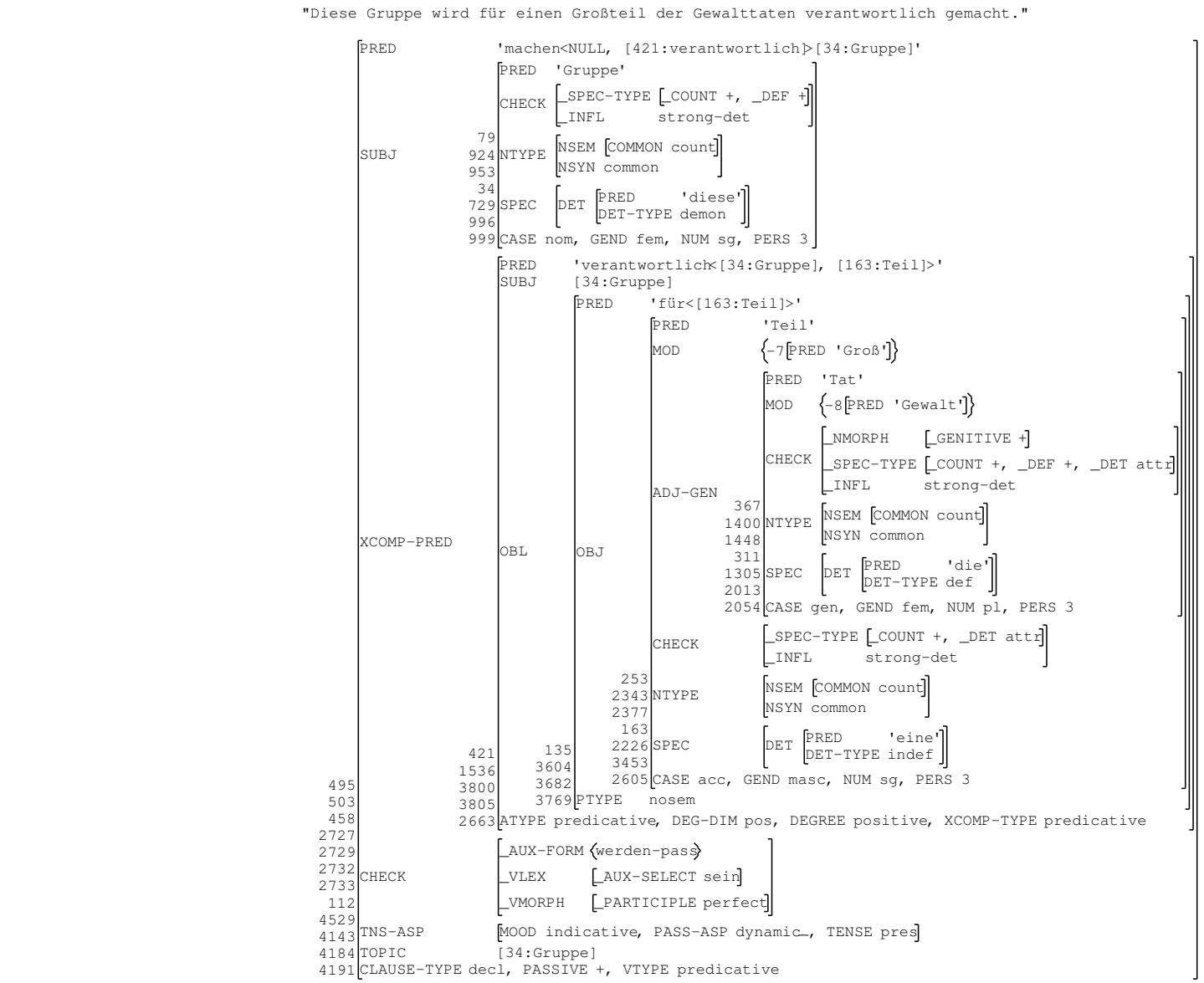


Figure 2.2: LFG-based generation input: an F-structure representation

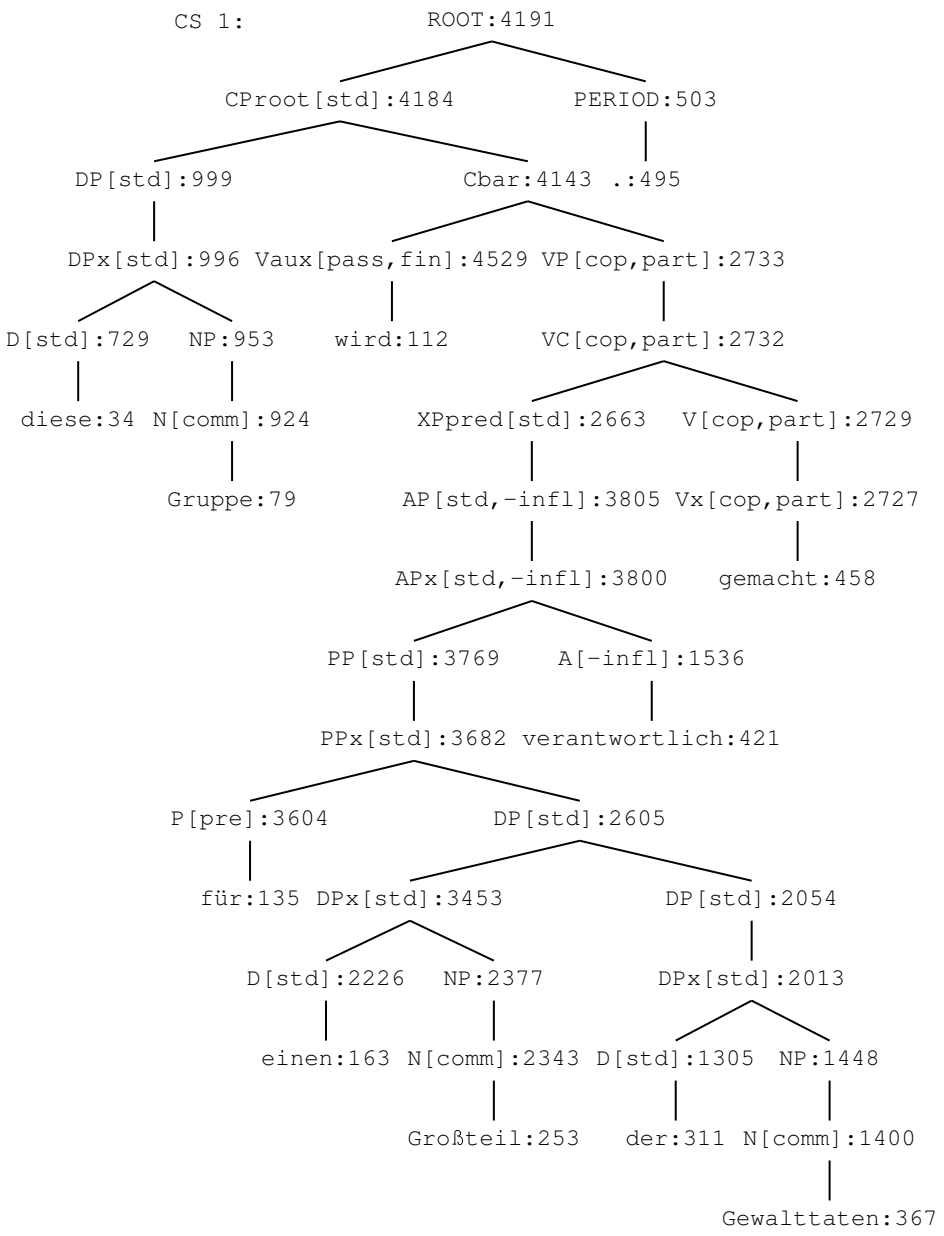


Figure 2.3: Surface syntax in LFG-based generation: a C-structure representation

nized in phrases, units of contiguous words, that correspond to some conceptual unit (for instance a verb phrase, a subject, etc.). By constrast, non-configurational languages do not organize conceptual units in terms of surface syntactic phrases but mark syntactic relations on the lexical words (via morphology). Despite these differences in the surface syntax, languages still seem to have similar syntactic organization principles on a deeper level that concerns the constraints on conceptual units, i.e. grammatical relations, that can be expressed. This deeper level is described by LFG F-structures.

This general theoretical goal has been put to a practical test in the context of the ParGram Project (Butt et al., 2002). An important aim of the project is to develop methodologies for large-scale development of grammars, such that the LFG formalism, its universality and parallelism can be empirically tested on a range of languages. For this prospect, the parallel definition of F-structure representations across languages plays a major role. Butt et al. (2002) discuss various methods and tools for maintaining parallelism of F-structure representations developed for a range of languages.

2.2.2 Grammar-based Candidate Generation

An important practical basis of the ParGram project is the XLE platform for grammar development which includes a very efficient LFG parser and a generator. Within the spectrum of approaches to natural language parsing, XLE can be considered a hybrid system that combines a hand-crafted grammar with a number of automatic ambiguity management techniques: (i) C-structure pruning where, based on information from statistically obtained parses, some trees are ruled out before F-structure unification (Cahill et al., 2007b), (ii) an Optimality Theory-style constraint mechanism for filtering and ranking competing analyses (Frank et al., 2001), and (iii) a stochastic disambiguation component which is based on a log-linear probability model (Riezler et al., 2002) and works on the packed representations. Moreover, the platform comes for a so-called *transfer* module where term-rewrite rules that map F-structures to some external representation can be specified.

The LFG parsing mechanism that is implemented in the XLE platform is defined in a reversible way. Therefore, XLE can be used f or generating C-structures and surface sentences from an F-structure input. The generator maps an F-structure to the set of all C-structures licensed by the grammar. Thus, if the LFG grammar assigns identical F-structures to sentences that differ in their word order, the generator will produce these different word

orders, i.e. their corresponding C-structures, in its output.

The XLE generator was originally designed for grammar development purposes. The originally intended usage of the system is a scenario where a grammar writer can manually check whether the grammar overgenerates, i.e. whether it licenses ungrammatical surface strings.

Cahill et al. (2007a) first pursued the idea to use the LFG generator for regenerating from F-structures obtained as analyses for corpus sentences with the German LFG grammar. As we discussed above, the F-structure is a representation that abstracts from certain surface syntactic aspects of the surface realization. For instance, it does not specify the order of constituents or the realization of an auxiliary. Figure 2.4 shows the word order variants for the Sentence (2), that the grammar produces for the underlying F-structure for the sentence.

(2) Der größte Teil der Unternehmen arbeite allerdings
The biggest part the.GEN companies work however
weiterhin mit Verlust.
still with loss.

As the German LFG grammar is a precise and broad-coverage model of German syntax, the generator will mostly produce grammatical sentences for an abstract sentence. But the candidate set in Figure 2.4 also shows that the generation output might comprise candidates that a native speaker of German would never produce intuitively, but that are actually licensed by the grammar. Sentence (2) contains two adverbs ("allerdings" and "weiterhin") and the grammar basically produces all variants to position these adverbs in the sentence, where some of these would be clearly rejected by a native speaker of German (e.g. the variants with "allerdings" at the end). As it is generally hard to categorize adverbials and predict their positions in German syntax, the grammar is not very restrictive here. This example suggests that there is a certain grey area where it is not clear whether a certain word order should be ruled out by the hard constraints encoded in the grammar, or the soft constraints implemented in a ranker.

Compatibility Checking When the XLE parser yields an ambiguous F-structure in the analysis step of the regeneration pipeline, a particular structure has to be selected for the generation step. Otherwise, the generator would also generate from an F-structure chart and produce all sentences that

```
Mit Verlust arbeite weiterhin der größte Teil der Unternehmen allerdings.
Mit Verlust arbeite allerdings weiterhin der größte Teil der Unternehmen.
Mit Verlust arbeite weiterhin allerdings der größte Teil der Unternehmen.
Allerdings arbeite der größte Teil der Unternehmen mit Verlust weiterhin.
Allerdings arbeite mit Verlust der größte Teil der Unternehmen weiterhin.
Allerdings arbeite der größte Teil der Unternehmen weiterhin mit Verlust.
Allerdings arbeite weiterhin der größte Teil der Unternehmen mit Verlust.
Allerdings arbeite mit Verlust weiterhin der größte Teil der Unternehmen.
Allerdings arbeite weiterhin mit Verlust der größte Teil der Unternehmen.
Der größte Teil der Unternehmen arbeite mit Verlust allerdings weiterhin.
Weiterhin arbeite der größte Teil der Unternehmen mit Verlust allerdings.
Der größte Teil der Unternehmen arbeite allerdings mit Verlust weiterhin.
Der größte Teil der Unternehmen arbeite allerdings weiterhin mit Verlust.
Der größte Teil der Unternehmen arbeite weiterhin allerdings mit Verlust.
Der größte Teil der Unternehmen arbeite mit Verlust weiterhin allerdings.
Der größte Teil der Unternehmen arbeite weiterhin mit Verlust allerdings.
Weiterhin arbeite mit Verlust der größte Teil der Unternehmen allerdings.
Weiterhin arbeite der größte Teil der Unternehmen allerdings mit Verlust.
Weiterhin arbeite allerdings der größte Teil der Unternehmen mit Verlust.
Weiterhin arbeite mit Verlust allerdings der größte Teil der Unternehmen.
Weiterhin arbeite allerdings mit Verlust der größte Teil der Unternehmen.
Mit Verlust arbeite der größte Teil der Unternehmen allerdings weiterhin.
Mit Verlust arbeite allerdings der größte Teil der Unternehmen weiterhin.
Mit Verlust arbeite der größte Teil der Unternehmen weiterhin allerdings.
```

Figure 2.4: Word order alternations produced by the German LFG for the sentence "Der größte Teil der Unternehmen arbeite allerdings weiterhin mit Verlust."

correspond to the different analyses. In principle, any procedure could be used to select an unambiguous F-structure input for the generator. However, it has to be kept in mind that certain incorrect analyses of a sentences yield paraphrases that are not meaning-equivalent. (e.g. incorrect PP-attachments typically result in unnatural and non-equivalent surface realizations). For this reason, Cahill et al. (2007a) use sentences from an annotated treebank for their experiments. Based on the gold syntax annotation, they choose the parsed F-structures that are compatible with the manual annotation. This compatibility check eliminates noise which would be introduced by generating from incorrect parses. Practically, this compatibility check can be non-trivial if the annotation format of the treebank differs from the F-structure in the way certain phenomena are represented, see Forst (2007).

"Blumen gekauft hat er."

```
      [PRED     'kaufen<[118:pro], [1:Blume]>'                                                    ]
      [       118[PRED  'pro'                                                                     ]
      [SUBJ   950[NTYPE [NSYN pronoun]                                                            ]
      [       953[CASE nom, GEND masc, NUM sg, PERS 3, PRON-FORM sie, PRON-TYPE pers              ]
      [      1026[                                                                                ]
      [         1[PRED  'Blume'                                                                   ]
      [       371[                                                                                ]
      [OBJ    374[NTYPE [NSEM [COMMON count]]                                                     ]
      [       673[      [NSYN common       ]                                                     ]
  156 [       676[CASE acc, GEND fem, NUM pl, PERS 3                                              ]
  164 [                                                                                           ]
   72 [          [_AUX-FORM ⟨haben⟩                                                              ]
  750 [                                                                                           ]
   34 [CHECK     [_VCONSTR  [_VP [_TOPIC head]]                                                  ]
  632 [                                                                                           ]
  635 [          [_VLEX     [_AUX-SELECT haben]                                                  ]
 1406 [          [_VMORPH   [_PARTICIPLE perfect]                                                ]
 1523 [                                                                                           ]
 1097 [TNS-ASP  [MOOD indicative, PERF +_, TENSE pres                                            ]
 1104 [CLAUSE-TYPE decl, PASSIVE -, VTYPE main                                                   ]
```

"Er hat Blumen gekauft."

```
      [PRED     'kaufen<[4:pro], [109:Blume]>'                                                    ]
      [         4[PRED  'pro'                                                                     ]
      [SUBJ   462[NTYPE [NSYN pronoun]                                                            ]
      [       465[CASE nom, GEND masc, NUM sg, PERS 3, PRON-FORM sie, PRON-TYPE pers              ]
      [       581[                                                                                ]
      [       109[PRED  'Blume'                                                                   ]
      [       828[                                                                                ]
  183 [OBJ    869[NTYPE [NSEM [COMMON count]]                                                     ]
  191 [      1110[      [NSYN common       ]                                                     ]
  147 [      1113[CASE acc, GEND fem, NUM pl, PERS 3                                             ]
 1063 [                                                                                           ]
 1066 [          [_AUX-FORM ⟨haben⟩                                                              ]
 1514 [                                                                                           ]
 1562 [CHECK     [_VLEX     [_AUX-SELECT haben]                                                  ]
   69 [          [_VMORPH   [_PARTICIPLE perfect]                                                ]
  663 [                                                                                           ]
 1149 [TNS-ASP  [MOOD indicative, PERF +_, TENSE pres                                            ]
 1181 [TOPIC    [4:pro]                                                                          ]
 1188 [CLAUSE-TYPE decl, PASSIVE -, VTYPE main                                                   ]
```

Figure 2.5: LFG F-structure analyses for two word order variants for partial VP fronting

Output Tweaking Our general discussion of the LFG formalism in the previous Section 2.2.1 has basically suggested that F-structures constitute a representation that does not encode information about surface syntax, such as sequential order. In large-scale LFG implementations, however, this assumption turns out to be an idealization. The practical grammar implementations in the ParGram project provide various examples where grammar writers make use of surface-oriented features on the level of F-structures in order to express certain syntactic constraints. For generation this means that, in certain cases, pretty detailed knowledge of a grammar and its F-structure representation might be required to obtain or reproduce a particular generation output.

Figure 2.5 presents the two F-structures obtained by parsing the Sentences in (3) and (4) with the German LFG. Sentence (3) shows an example of VP fronting which is a word order variant for Sentence (4).

(3) Blumen gekauft hat er.
Flowers bought has he.
'He has bought flowers.'

(4) Er hat Blumen gekauft.
He has flowers bought.
'He has bought flowers.'

If the F-structures in Figure 2.5 are used for regeneration without further transformations, VP fronting cannot be generated as a paraphrase for canonical word order and vice versa. The reason is that the first F-structure in Figure 2.5 specifies the fronted VP as _VP _TOPIC under the CHECK _VCONSTR feature. The second F-structure produced for the canonical word order lacks the CHECK _VCONSTR feature and specifies the subject of the sentence as a TOPIC. For such cases, the configuration of the XLE generator allows F-structure features to be listed as "removable" and "addable". These features are then freely removed and added by the generator such that all possible instantiations are produced. In order to generate VP fronting along with other word orders, the following features would need be specified as such: _VCONSTR _VP _TOPIC and TOPIC.

The interaction between grammar-internal specifications and generator output can be even more complex. In the case of VP fronting, the grammar writers of the German LFG have introduced an OT constraint that prohibits the generation of fronted (partial) VPs (in the grammar versions used by

Cahill et al. (2007a); Cahill and Riester (2009)). This constraint has been introduced for efficiency reasons. If the generator sees an input like the first F-structure in Figure 2.5, it only produces word orders as output that do not include VP fronting. As a result, the original corpus sentence is not among the generated surface realizations. Other OT constraints with similar effects on the generation output have been included for e.g. extraposed relative clauses, parentheticals, or punctuation.

2.2.3 Statistical Ranking

The training of a surface realization ranking component in the reversible LFG architecture is parallel to disambiguation in parsing: In generation, an input F-structure is associated with several C-structures (mapping to surface sentences) and the ranker has to discriminate between the appropriate C-structure. In parsing, an input sentences is associated with several C-structures and F-structures and the ranker has to discriminate between the correct analyses.

Cahill et al. (2007a)'s surface realization ranking component adopts the set-up of the parse reranking component implemented by Forst (2007). It is based on the `cometc` software provided with the XLE framework which can be used to train a log-linear model that discriminates between several C-structure/F-structure pairs. `cometc` takes packed LFG representations, i.e. charts of C-structures and F-structures, as input such that training and testing can be done efficiently.

XLE's ranking component comes with a number of hardwired feature templates, originally designed for disambiguation with the English ParGram grammar (Riezler et al., 2002; Riezler and Vasserman, 2004). Forst (2007) extended these templates to German, adding some more flexible transfer-based feature templates. Essentially, Cahill et al. (2007a) uses these templates, but due to the application of automatic feature selection the remaining features in the model are not necessarily parallel to the parse disambiguation model. Cahill and Riester (2009) add more features that are based on an informed, theoretically inspired model of information status.

Using XLE's hardwired ranking component for realization ranking has, however, one particular technical drawback: If external features such as sentence length and language model scores are integrated into the model, all the C-structures have to be unpacked in order to be able to assign different scores to each surface sentence. This unpacking step leads to a considerable

slow-down of the entire development cycle as representations can be large for long sentences. For this reason, we decided to reimplement the surface realization ranking component based on an external ranking module. This module does not discriminate between C-structures, but directly between surface sentences.

The more flexible implementation of the ranking component also has the advantage that we can directly use it for the extended version of the surface realization architecture presented in Chapter 5.1, 6.2, and 7.3. Technically, the only difference in the extended ranking is that the sentences are not generated from a single F-structure, but from several F-structure candidates. Our reimplementation of the realization ranking component is an SVM ranking model implemented with SVMrank, a Support Vector Machine-based learning tool Joachims (2006). The input to the module is a set of items where each item corresponds to the set of candidate sentences generated by XLE. Each sentence or surface realization candidate is annotated with a rank and a set of features extracted from the F-structure, its surface string and external resources (e.g. a language model).

Labeling For the training of our ranking model, we have to tell the learner how closely each surface realization candidate resembles the original corpus sentence. If the sentence matches the original corpus string, its rank will be highest, the assumption being that the original sentence corresponds to the optimal realization in context. During testing, the ranker predicts a rank for each candidate. The output of generation, the top-ranked sentence, is evaluated against the original corpus sentence.

We distinguish the ranks: "1" identical to the corpus string, "2" identical to the corpus string ignoring punctuation, "3" small edit distance ($<$ 4) to the corpus string ignoring punctuation, "4" different from the corpus sentence. The intermediate ranks "2" and "3" are useful since the grammar does not always regenerate the exact corpus string, see Cahill et al. (2007a) for explanation.

Features Since we want to avoid the unpacking of C-structures associated with the realization candidates, we cannot use the existing hardwired feature templates and we basically have to ignore the information coming from the underlying C-structures. Thus, our feature extraction is a little bit more involved than in the work by Cahill et al. (2007a). The main idea is to relate

the nodes or PREDs in the F-structures to the tokens in the surface sentence. If this mapping is established, we can relate the tokens in the sentence to the annotations in the F-structure, i.e. we can extract their grammatical function, etc.

The feature model is built as follows: for every lemma in the F-structure, we extract a set of morphological properties, the voice of the verbal head, its syntactic and semantic role, and a set of informations status features following Cahill and Riester (2009). The morphological properties are basically taken from Cahill et al. (2007a) and include definiteness, person, pronoun type, noun type, etc. These properties are combined in two ways:

- Precedence features: relative order of properties in the surface string, e.g. "subject < direct object" "theme < agent in passive", "1st person < 3rd person", "adverbial adjunct < subject" ;

- Non-precedence features: combinations of voice and role properties with morphological properties, e.g. "subject is singular", "agent is 3rd person in active voice".

Note that the non-precedence and voice-related features are only relevant for the extended surface realization architecture and the Experiment 8 in Chapter 7.3. In addition, each candidate is annotated with features extracted from the underlying F-structure and meaning representation, the surface string of the sentence and a language model score.[1] An example for a training item is given in Figure 2.6.

In terms of the underlying feature model, our ranking component is not an exact reimplementation of Cahill et al. (2007a), but the feature set is reduced to precedence relations between properties in the F-structure. Cahill et al. (2007a)'s model also includes a number of C-structure features, e.g. a feature the counts the frequency of certain constituent label in the underlying tree. We obtain similar performance with our ranking component, so we decided to ignore C-structure features.

[1]The language model is trained on the German data release for the 2009 ACL Workshop on Machine Translation shared task, 11,991,277 total sentences.

R	*Sentence and Features*
1	% **Diese Gruppe** wird für einen Großteil der Gewalttaten verantwortlich gemacht. % *This group is for a major part of the violent acts responsible made.* subject-<-pp-object, demonstrative-<-indefinite, lm:-7.89
3	% Für einen Großteil der Gewalttaten wird **diese Gruppe** verantwortlich gemacht. % *For a major part of the violent acts is this group responsible made.* pp-object-<-subject, indefinite-<-demonstrative, lm:-10.33
3	% Verantwortlich gemacht wird **diese Gruppe** für einen Großteil der Gewalttaten. % *Responsible made is this group for a major part of the violent acts.* subject-<-pp-object, demonstrative-<-indefinite, lm:-9.41

Figure 2.6: Training example for surface realization ranking: candidate sentences annotated with LFG-based precedence features extracted from F-structures and language model scores

2.3 Statistical Dependency-based Linearization

This Section presents the statistical dependency linearizer from Bohnet et al. (2012), which is an advanced version of the linearization component taken from the semantic realization pipeline in Bohnet et al. (2010). The component is called linearizer as it takes an unordered dependency tree as input where all surface tokens are specified and it computes the linear order of these tokens. We used it in multi-level pipeline set-up in Chapter 7.2.

A major advantage of the linearizer is that it can be trained on any type of dependency annotation, without imposing specific requirements on its input. It does not use a grammar-based generator for producing candidates. Instead, candidate selection and generation are closely interleaved in a statistical decoding procedure that iteratively maps an unordered tree to an optimally ordered tree. The candidate space of possible linear orders is defined, on the one hand, by the scheme of dependency annotation and, on the other hand, by the algorithm used for decoding. Section 2.3.1 introduces dependency syntax and some relevant aspects for generation. Section 2.3.2 provides more details about the linearization procedure.

2.3.1 Dependency Syntax

In the past decade, dependency syntax has become a popular framework for broad-coverage, probabilistic approaches to parsing. Nowadays, a number

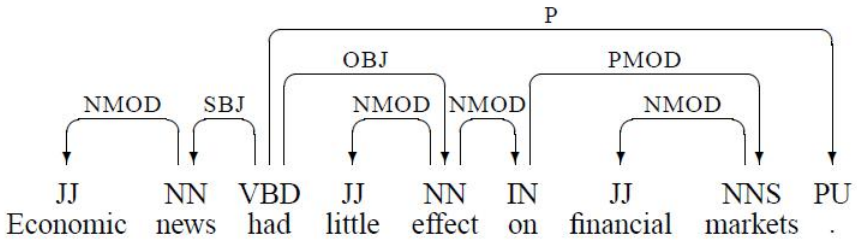


Figure 2.7: Dependency syntax: example annotation from the Penn Treebank

of off-the-shelf dependency parsers and annotated treebanks in a number of languages are available (Merlo et al., 2010).

In dependency syntax, the analysis of a sentence is represented by binary relations between lexical words, so-called dependencies. Figure 2.7 shows a dependency tree from the English PennTreebank, adopted from Nivre (2005). Each dependency encodes a labeled relation such as SBJ (subject), its syntactic head (*had*) and the syntactic dependent (*news*). Thus, in contrast to standard constituency trees, dependency syntax encodes the predicate argument structure in a transparent way such that is useful for a range of NLP applications and also seems better suited for analyzing free word order languages (Merlo et al., 2010).

Thus, the motivation that underlies the design of F-structures and dependencies, as they are used for instance in the English PennTreebank, is fairly similar. However, the dependency representation used in corpus-based broad-coverage parsing is typically more shallow. This is shown in Figure 2.8 where both analyses for a German corpus sentence are given. The simplified, shallow dependency tree in Figure 2.8 represents the relations between words independently of the surface order in the original sentence (in contrast to the representation in Figure 2.7). This is the way dependency trees will be used as input for the linearizer.

The major differences between the representations are that a) the shallow dependency analysis contains auxiliary words such as copula verbs or determiners, and b) the F-structure represents deep argument relations that can be recovered from the syntax, such as the pronoun *er* (*her*) which figures as the subject of the passivized main verb and the object of the embedded verb, c) the F-structure represents more information about verb morphology, i.e.

it recognizes the main verb as a passivized predicate.

Another difference between dependency and LFG syntax is that the dependency community subsumes a fairly heterogeneous range of grammars, annotation styles and formalisms (Nivre, 2005). Most of the dependency treebanks used in the NLP community are actually not genuine dependency annotations that have been manually crafted, but automatic conversions from some other format of syntactic representation.

Bohnet et al. (2012) train their system on Seeker and Kuhn (2012)'s dependency conversion for the German TiGer treebank (Brants et al., 2002). The treebank has been originally annotated with phrase-structures where functional labels are attached to each edge so that the labels can be used for the dependency annotation.

When converting phrase structures to dependencies, the main task of the procedure is to detect the head of the phrase where all its dependents can be attached. As discussed in Seeker and Kuhn (2012), there are some syntactic phenomena that seem to fall out of this simple syntactic scheme and receive different treatments in different conversions and treebanks. For instance, Figure 2.9 shows the possible options for annotating coordinated phrases with a dependency structure. Note that these annotation options have direct implications for the linearization task: The structure at the bottom of Figure 2.9, adopted by Seeker and Kuhn (2012), encodes the linear order of the conjuncts in the labeling scheme, i.e. the first conjunct of a coordination is always the head where the conjunction is attached to. A statistical linearizer trained on these dependencies will not have to treat variation in the linear order of conjuncts as opposed to a system trained on structures shown at the top-left of Figure 2.9.

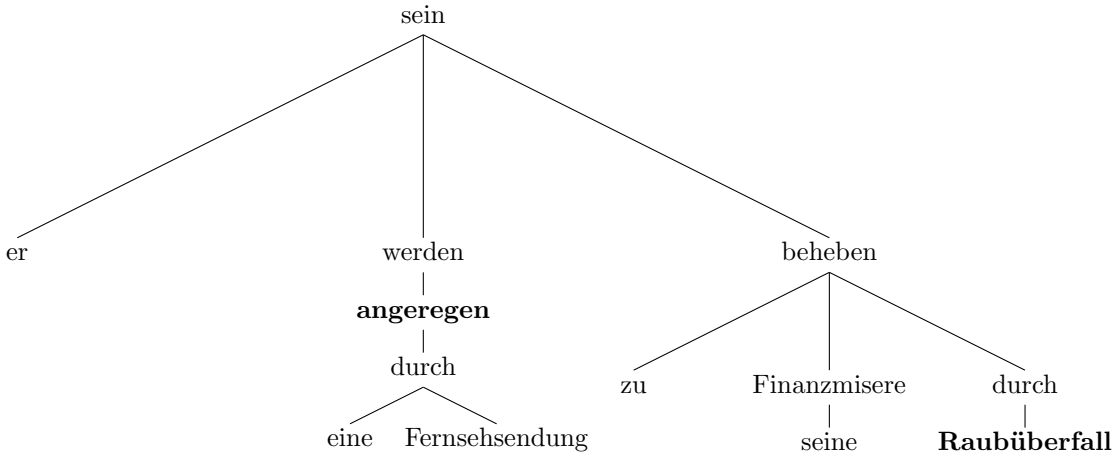
2.3.2 The Linearization Procedure

Bohnet et al. (2012)'s linearization system takes as input an unordered dependency tree. It implements a top-down algorithm that starts traversing the tree from the root node and builds so-called word order domains. A word order domain contains a head word with all its direct syntactic dependents.

The word order domains for the tree in Figure 2.7 would be as follows: {*had,news,effect,.*}, {*news,economic*}, {*effect,little,on*}, {*on,market*}, {*market,financial*}. The algorithm would start of the word order domain {*had,news,effect,.*}, the position of the subtrees, e.g. {*news,economic*} is then recursively computed from the position of the heads.

(5) Durch eine Fernsehsendung sei er angeregt worden, seine
By a TV show was he incited be, his
Finanzmisere durch einen Raubüberfall zu beheben.
financial misery by a robbery to solve.
He was incited by a TV show to solve his financial problems with a robbery.

Shallow dependency analysis:



F-structure analysis:

$$\begin{bmatrix} \text{PRED} & \text{'anregen} < (\uparrow \text{NULL})(\uparrow \text{SUBJ})(\uparrow \text{VCOMP}) > \text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'er'} \end{bmatrix} \\ \text{VCOMP} & \begin{bmatrix} \text{PRED} & \text{'beheben} < (\uparrow \text{SUBJ})(\uparrow \text{OBJ}) > \text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'er'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Finanzmisere'} \end{bmatrix} \\ \text{ADJUNCT} & \begin{bmatrix} \text{PRED} & \text{'durch} < (\uparrow \text{OBJ}) > \text{'} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Raubüberfall'} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{ADJUNCT} & \begin{bmatrix} \text{PRED} & \text{'durch} < (\uparrow \text{OBJ}) > \text{'} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Fernsehsendung'} \end{bmatrix} \end{bmatrix} \\ \text{PASS} & + \end{bmatrix}$$

Figure 2.8: Shallow dependency and F-structure analysis for a sentence from the robbery data set

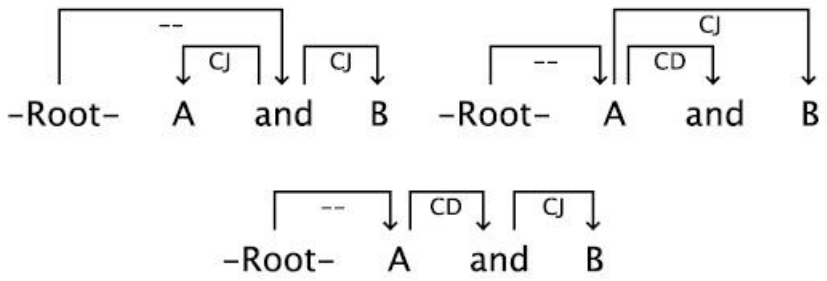


Figure 2.9: Options for the dependency-based analysis of coordination

In contrast to the generate-and-rank system from Section 2.2, the system takes fairly local ordering decisions. Thus, for linearizing the the tree from Figure 2.7, it first orders the domain {*had,news,effect,.*}, which basically means determining the order of predicate arguments and subsequently realizes the phrase-internal orders for PPs and NPs.

An exhaustive list of the candidates that could be produced by this linearization algorithm would correspond to all possible combinations of the different permutations of the embedded word order domains. However, the system keeps a list of word orders that are optimal given a probabilistic scoring function. At each step in the linearization algorithm, an SVM classifier scores alternative orderings of each word order domain. This classifier is previously trained on the ordered dependency trees in the original corpus. The feature model uses a list of templates that combine the lexical and syntactic properties (part-of-speech, lemma, label) of the dependents that have to be ordered. For being able to use global features that apply to subtrees larger than a particular domain, the system uses a beam search, meaning that it keeps a stack of alternative linearizations at each step and condition its decisions on global and contextual features.

In this way, the statistical linearizer treats the problem of candidate generation and selection in an interleaved procedure. The final output linearization is the sentence that is within the beam and has the highest score which is computed as a sum of the local scores for the word order domains and some global features.

Non-Projectivity Note that the recursive linearization algorithm for word order domains imposes some constraints on the orders that can be reproduced

from the tree. For instance, the algorithm cannot reproduce the original order for the sentence in Figure 2.8 where the adjunct *durch eine Fernsehsendung*, the dependent of the main verb *anregen* is realized in the sentence-initial position. The only way to realize the adjunct before the finite verb, would be to realize the entire verb phrase in sentence-initial position:

(6) Durch eine Fernsehsendung angeregt worden sei er, seine
By a TV show incited was he, his financial misery
Finanzmisere durch einen Raubüberfall zu beheben.
by a robbery to solve.
He was incited by a TV show to solve his financial problems with a robbery.

This problem is related to so-called non-projective or crossing edges in ordered dependency trees. Thus, the edge that attaches the adjunct headed by *durch* to its verbal head *angeregt* crosses the edge that attaches *angeregt* to its auxiliary head *sei*. Note that this problem would not occur in deep F-structure-style dependencies were auxiliaries are not recorded as nodes in the tree.

In its formulation described above, the statistical linearization procedure produces only linear orders that result in projective trees. To solve this issue, Bohnet et al. (2012) implement a preprocessing step that lift dependency edges in the tree.

To provide some intuition for the output quality of the statistical linearizer, Example (7-a) shows a sentence produced by the core projective version of the linearizer, (7-b) shows the same sentence produced with an preproccesing for non-projectivity:

(7) a. das betriebsergebnis war zuletzt vor bewertung aufgrund) niedriger zinsen und hoher investitionen in die modernisierung des filialnetzes zwischen hachenburg) rheinland-pfalz (und hochheim (hessen um 14 prozent auf 243 millionen gesunken .

b. aufgrund) niedriger zinsen und hoher investitionen in die modernisierung des filialnetzes zwischen hachenburg) rheinland-pfalz (und hochheim (hessen war zuletzt das betriebsergebnis vor bewertung um 14 prozent auf 243 millionen gesunken .
'The operational result recently dropped by 14% to 243 millions due to low interest rates and big investments in the modernization of the branch network between Hachenburg (Rheinland-

Pfalz) and Hochheim (Hessen).'

Disregarding the problem with predicting punctuation, the output quality in Sentence (7-b) is fairly impressive given that we are dealing with a long and complex sentence and given that the system does not use a grammar. Ignoring the misplaced brackets, the system actually reproduces the word order of the original corpus sentence. However, the output quality also has to be seen in the light of the specific dependency annotation scheme. For instance, the phrase-internal word order of the constituent headed by *aufgrund* which is placed in the *Vorfeld* in Sentence (7-b) is basically predetermined by the fact that the dependency labels encode the order of the conjuncts.

2.4 Corpus-based Referring Expression Generation

Whereas the LFG-based surface realization and dependency-based linearization systems discussed above deal with the mapping between a linguistic sentence and a syntactic representation in the reverse direction of of LFG-based and dependency-based parsing, the task-definition of corpus-based REG, its underlying representations and assumptions cannot be directly mapped to an understanding task. Of course, coreference resolution and named entity recognition systems from the NLU domain also broadly address the problem of finding and interpreting mentions of referents in a text. But, to the best of our knowledge, the detection and realizations of referents in an input text have so far been treated from relatively different angles, using different methods and algorithms in NLU and NLG.

The thesis by Dale (1992) (or Dale (1989)) initiated extensive research in the generation community that treated the problem of realizing referring expressions as a free-standing problem. and established an important paradigm for REG research: the task is restricted to generating definite noun phrases that identify concrete objects to a hearer. The input is defined as a set of objects described by a set of attributes in a knowledge base. The optimal REG algorithm is supposed to generate expressions that identify objects from that knowledge base such that the hearer will not confuse them with the other distractors in the set, and such that he does not make false implicatures, i.e. the expression should be minimal or effective in some sense. In this paradigm, the main problem for generating referring expressions is to select

attributes that identify a referent in a meaningful way.

Recent work on this problem discovered that human strategies for identifying objects in a scene do not always comply with computational definitions of pragmatic effectiveness (Viethen and Dale, 2006, 2010), e.g. humans produce redundant attributes that would not be necessary for distinguishing a referent in a strict sense. Therefore, Garoufi and Koller (2013) propose to address the task in an interactive dialogue system where effectiveness can be more objectively assessed in terms of the usefulness of the generation output. Other extensions of the basic distractor paradigm considered further types of references, such as relational descriptions or reference to sets, see Krahmer and Van Deemter (2012) for an overview.

A first step towards corpus-based, contextual REG is done by Siddharthan and Copestake (2004) who try to regenerate first mentions of referents in the Penn WSJ Treebank. They automatically extract these mentions and a set of possible distractors from the surrounding sentences in the discourse. They argue that the classical approach to REG in line with Dale (1992) makes the following quite restrictive and artificial assumptions (which should be overcome by a corpus-based, less knowledge-intensive approach): a semantic representation exists, an attribute classification scheme exists, the linguistic realizations are unambiguous, and attributes cannot be reference modifying. From an NLU perspective, all these assumptions are not realistic such that most work on REG has been carried out in small domains.

While Siddharthan and Copestake (2004) try to transplant the classical attribute-oriented REG approach into a corpus-based framework, Belz and Varges (2007) take a different perspective on corpus-based REG, mainly targeting the prediction of subsequent reference. The paradigm of subsequent reference does not restrict the REG task to identifying a particular object once in a certain scene, but especially includes mentions of referents that are already known to the speaker from the previous discourse. They draw on some early empirical studies that looked at the prediction of pronominalization in corpus text (McCoy and Strube, 1999; Henschel et al., 2000). As the resolution of pronouns is a key issue for automatic coreference resolution, this perspective on corpus-based REG seems to be more directly related to interpretation approaches where chains of expressions referring to a particular entity need to be automatically recognized in free corpus text.

2.4.1 Main Subject Reference Generation

Due to the fact that resources with entity or referent-based annotations and tools that predict these annotations automatically are much less mature than syntactic annotations and automatic parsers, Belz and Varges (2007) create their own data set where the corpus-based REG paradigm can be addressed in a meaningful way. Belz and Varges (2007) create a corpus of 1000 Wikipedia articles about cities, countries, rivers and people and annotated them for the references to the main subject. They phrase the REG task as follows: all REs from the original text are extracted and provided as an input candidate list. The input to the REG system is a text with reference slots that have to be filled with instances from the candidate list. An example input is shown in Figure 2.10.

The systems that were submitted to the series of GREC shared tasks (Belz et al., 2008, 2009; Belz and Kow, 2010) mostly implemented the task as a classification problem. The classifier or a set of classifiers is trained to label a slot in a corpus sentence with a corresponding type of referent. The contextually most appropriate type has the highest score given the feature model of the classifier. The main differences between the systems relate to the training method for the classifier, the underlying feature models and how the task with split up among several classifiers.

As such, the problem of determining contextually appropriate REs is similar to ranking surface realizations corresponding to an underlying syntactic representation, as described in Section 2.2.3. In both cases, an exhaustive list of candidates is given in the input and the contextually most appropriate candidate is the represented by the highest global score according to a feature model. In contrast to grammar-based surface realization, however, the space of candidates in the GREC task is defined through the list of expressions found in a corpus. Thus, if an author decided to use 14 different definite descriptions to refer to an entity in a text, as opposed to a less creative author who only used 3 different expressions, the size of the candidate list varies accordingly. Belz and Varges (2007) partly account for this variation by adding some default expressions to each candidate list, making sure that the basic types of REs can always be produced by the system (i.e. pronouns, default name, category nouns).

While this approach to REG is less domain-specific and makes less assumptions about underlying semantic representations for referents and their attributes, it still relies on some far-reaching, simplifying assumptions in

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TEXT SYSTEM "reg08-grec.dtd">
<TEXT ID="36">
<TITLE>Jean Baudrillard</TITLE>
<PARAGRAPH>
 <REF ID="36.1" SEMCAT="person" SYNCAT="np-subj">
  <REFEX REG08-TYPE="name" EMPHATIC="no" HEAD="nominal" CASE="plain">Jean Baudrillard</REFEX>
  <ALT-REFEX>
   <REFEX REG08-TYPE="name" EMPHATIC="no" HEAD="nominal" CASE="plain">Jean Baudrillard</REFEX>
   <REFEX REG08-TYPE="name" EMPHATIC="yes" HEAD="nominal" CASE="plain">Jean Baudrillard himself</REFEX>
   <REFEX REG08-TYPE="empty">_</REFEX>
   <REFEX REG08-TYPE="pronoun" EMPHATIC="no" HEAD="pronoun" CASE="nominative">he</REFEX>
   <REFEX REG08-TYPE="pronoun" EMPHATIC="yes" HEAD="pronoun" CASE="nominative">he himself</REFEX>
   <REFEX REG08-TYPE="pronoun" EMPHATIC="no" HEAD="rel-pron" CASE="nominative">who</REFEX>
   <REFEX REG08-TYPE="pronoun" EMPHATIC="yes" HEAD="rel-pron" CASE="nominative">who himself</REFEX>
  </ALT-REFEX>
 </REF>
 (born June 20, 1929) is a cultural theorist, philosopher, political commentator,
 sociologist, and photographer.
 <REF ID="36.2" SEMCAT="person" SYNCAT="subj-det">
  <REFEX REG08-TYPE="pronoun" EMPHATIC="no" HEAD="pronoun" CASE="genitive">His</REFEX>
  <ALT-REFEX>
   <REFEX REG08-TYPE="name" EMPHATIC="no" HEAD="nominal" CASE="genitive">Jean Baudrillard's</REFEX>
   <REFEX REG08-TYPE="pronoun" EMPHATIC="no" HEAD="pronoun" CASE="genitive">his</REFEX>
   <REFEX REG08-TYPE="pronoun" EMPHATIC="no" HEAD="rel-pron" CASE="genitive">whose</REFEX>
  </ALT-REFEX>
 </REF>
 work is frequently associated with postmodernism and post-structuralism.
</PARAGRAPH>
</TEXT>
```

Figure 2.10: Example annotation for corpus-based REG due to Belz et al. (2008)

terms of the underlying input: First of all, choice in referring expressions is restricted to one particular referent in the text. This assumption sets the task apart from coreference resolution where the main challenge is to resolve expressions that could to refer to more than one entity. Second, the approach does not use any kind of abstract representation: candidate REs and sentences with RE slots are completely realized in terms of linguistic structures. This assumption sets the task apart from realistic text generation where such a shallow input cannot be expected. Possible applications for this task are more related to text-to-text generation and summarization where referring expressions can be treated in a post-processing step that edits the surface text (Siddharthan et al., 2011).

2.4.2 Our Approach

In this thesis, we adopt an approach to corpus-based REG that is strongly inspired from Belz and Varges (2007), but aims at overcoming some of its limitations from the perspective of general text generation. We created our own data set consisting of German newspaper articles that we annotated with particular types of referents. The specific design and annotation decisions for this data set are motivated in Chapter 5.1 and explained in detail in Chapter 5.4. We will give a brief summary here, as it is also relevant for an experimented described in Chapter 4.4.

The data set for our generation experiments consists of 200 newspaper articles about robbery events. The articles were extracted from a large German newspaper corpus. The texts in our robbery data set describe an event involving two main referents, a victim and a perpetrator (and sometimes an additional source, see below). Thus, we go beyond the MSR paradigm that only treats input texts where a single referent is important.

The robbery texts are manually annotated for different types of mentions of the two main referents. Moreover, the annotations include a shallow syntactic dependency layer of the entire sentences. The dependency structures are produced automatically using Bohnet (2010)'s state-of-the art probabilistic dependency parser. The shallow dependencies are mapped to a deep dependency layer by means of hand-written rules. Finally, the syntactic and referential annotations are integrated such that the most abstract representation for a text in our data set is a sequence of trees where referential expressions correspond to slots for a list of candidate phrases.

Figure 2.11 shows a simplified example of the combined deep syntactic

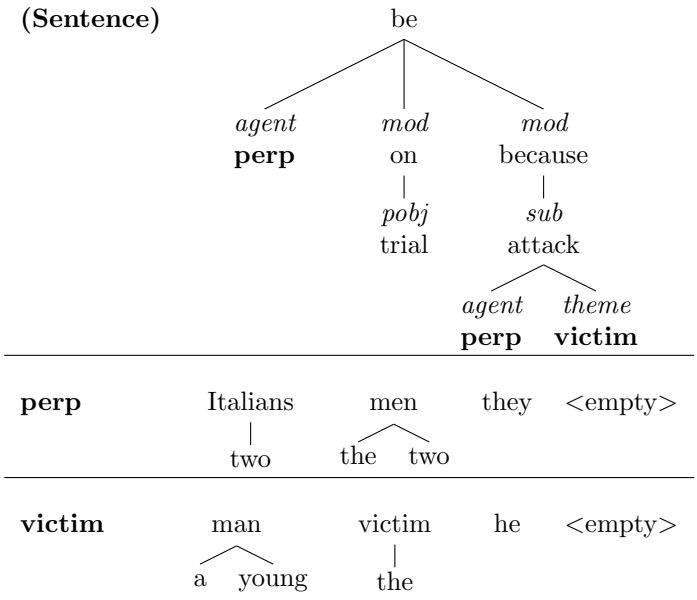


Figure 2.11: Example annotation in the robbery data set: deep dependency tree with RE candidates

dependencies and RE representation that we produced in or annotations: the tree defines dependency relations between abstract lemmas or nodes. Thus, if an NP in the sentence refers to the victim or perpetrator of the referent, we replace the subtree corresponding that NP by an abstract lemma that specifies the role of the referent.

Our approach to corpus-based REG basically integrates the task with the problem of surface realization discussed in the previous Section. Instead of inserting surface REs into slots of a predefined surface text, we assume a more abstract syntactic input representation. Moreover, our referent annotations include a a range of implicit referents that have not received much attention in previous work. Chapter 5 explains why the phenomenon of implicit referents is particularly relevant when addressing surface realization and REG in a combined setting.

2.5 Summary

In this Chapter, we have discussed three different generation frameworks, ranging from a classical grammar-based generate-and-rank system for surface realization, a purely statistical dependency-based system for linearization to a corpus-based approach for referring expression generation. These three approaches all deal with choice in language generation by exploiting corpus data that is annotated with some type of abstract input representation and that yields several linguistic candidate realizations in the generation output. As the discussion has shown, this general corpus-based setting can be instantiated in different ways in the respective systems leading to quite diverse assumptions about the underlying candidate space of a particular choice phenomenon and computational methods that are used to distinguish contextual appropriateness of a generation candidate.

Grammar-based generators that are implemented in a reversible parsing-generation architecture, as the LFG-based architecture in Section 2.2, are conceptually interesting for being applied to surface realization since they provide a relatively transparent way of data acquisition for generation and an explicit representation of generation candidates. Here, the idea is to regenerate from syntactic analyses of corpus sentences producing all grammatical realizations of a given abstract input. Thus, the candidate space that corresponds to a choice phenomenon, i.e. word order variation, is computed by means of a rule-based, objective (i.e. independent of a particular corpus) model of syntax. Candidate generation and ranking are treated in separate architectural stages, which nicely corresponds to a division of labour between hard syntactic constraints accounting for grammaticality and soft naturalness constraints accounting for contextual choice.

While the generate-and-rank architectures are theoretically elegant, we have also seen that a grammar-based system is, in practice, subject to a number of technical parameters that have an effect on the exact set of paraphrases produced by the generator. Without some detailed knowledge of grammar-internal specifications and configurations it might turn out difficult to reproduce candidate sets for certain sentences. Moreover, a clear practical limitation of grammar-based generators is that they require a very specific input representation which has to correspond to the analysis that the grammar would produce for the desired output sentence in the parsing direction. Consequently, the type of paraphrases that a generator can produce in its output is defined and fixed by the syntactic representation format used

by the underlying grammar. This issue is central for our work in Chapter 5.

The interactions between internal configurations of a system and assumptions of the given input representation are even harder to tease apart in the statistical linearization system presented in Section 2.3. This linearizer iteratively constructs an ordered version of a dependency tree from an unordered input, interleaving the determination of the candidate space and the probabilistic scoring. In principle, the system models the same choice phenomenon as the LFG-based surface realizer from Section 2.2, but it can be hardly compared in terms of its ability to predict word order variation. Due to the fact, that the dependency annotation used by the linearizer is more shallow, it encodes more information about the original surface realization. On the other hand, it does not have access to a model of syntactic constraints so that it can also produce ungrammatical output.

Finally, the corpus-based approach to REG discussed in Section 2.4 circumvents the problem of obtaining abstract inputs and corresponding generation candidates as much as possible. Here, the candidate space is defined by a list of expressions used for referring to a particular referent in a text. Thus, the candidate space is encoded in a very transparent way, and candidate selection can be phrased as a ranking between surface phrases. However, from the perspective of REG approaches that target the generation of identifying expression in line with Dale (1992), the input pre-specifies a number of interesting choices, i.e. the attributes used to describe a certain referent.

Essentially, the two paradigms prevalent in REG research illustrate the dilemma we have pointed out in the Introduction of this thesis: On the one hand, knowledge-intensive approaches targeting the generation of an appropriate content of a referring expression work on abstract, realistic inputs but are very domain-specific. On the other hand, surface-oriented, corpus-based approaches are more suited for capturing contextual factors on relatively broad candidate sets, but have to make simplifying assumptions in terms of the underlying input.

Chapter 3

Choice and Context

A major challenge for automatic text generation is the extremely variable nature of language production. The variability of linguistic expression allows speakers to realize a certain idea in many different ways. Thus, the process of generating natural language necessarily involves *choice*: the decision to realize some content by a particular combination of lexical words, arranged in a particular syntactic structure. From the perspective of language use, choice phenomena fullfil a function: they serve to adapt linguistic utterances to their context. If a generation system does not account for contextual dependencies of linguistic choices, it is likely to produce output that sounds unnatural under certain circumstances.

While the Introduction of this thesis gave some rather intuitive account of the importance of context in NLG, this Chapter discusses more systematically why and how context has been modeled in theoretical and computational approaches. As a range of state-of-the art generators that deal with choice and candidate selection have been shown to benefit from linguistically informed context modeling, we look at the problem from both perspectives.

Chapter 1 has discussed the fact *choice* in language generation is a complex phenomenon. Consequently, NLG research has been interested in controlled settings where particular choices, such as syntactic or referential choices can be studied in isolation. The same thing is actually true for investigations into *context*: it involves so many different aspects of a communicative situation that researchers are typically working in paradigms where particular contextual notions are described in a controlled setting.

Section 3.1 starts this Chapter with on overview of different linguistic perspectives on context modeling. The following Sections concentrate on

choice phenomena that are relevant for this thesis and discuss them from a theoretical and a generation perspective: word order (Section 3.2), syntactic verb alternations (Section 3.3) and referring expressions (Section 3.4).

3.1 Theoretical and Corpus-based Perspectives

The space of linguistic means of expression available to a speaker for realizing a certain content is usually considerable and basically cuts across all levels of linguistic processing, such as the structuring and ordering of sentences, the choice of lexical items and syntactic structure, the ordering and intonation of words. When humans produce language and speak, they deal with the large space of choices in a natural and often completely unconscious way. For theoretical descriptions of natural language discourse as well as NLG systems that produce text, these choice processes pose a major challenge.

The choice phenomena addressed in this thesis mostly do not affect the grammaticality of the resulting sentence. Thus, we are interested in so-called *soft constraints* on certain syntactic constructions or referring expressions. When the choice between two meaning-equivalent constructions is subject to soft constraints, it cannot be modeled as a categorical problem in the grammar. Instead, there will be differences in appropriateness or naturalness between to alternative realizations attributed to the context. The theoretical notions related to context and communicative appropriateness are still widely debated and are often hard to formalize. This situation is reflected in existing generation systems: while state-of-the-art generators are able to produce grammatically well-formed sentences, the implementation of naturalness constraints remains a challenge.

Why is it so hard to devise formal and computational systems that deal with contextual constraints on linguistic choices? The major reason seems to be the fact that virtually every aspect of a communicative context can have impact on the way people speak. First, a range of "extra-linguistic" factors such as age, social status, emotial relation of the interlocutors, properties of the communication channel, time and place of the communication etc. can be subsumed under the umbrella of contextual notions. But, second, "context" is also frequently used to refer to the more narrow "discourse context" which means that a particular sentence is typically surrounded by other sentences

in a text, utterance, or dialogue. Thus, a sentence will typically relate to other sentences uttered in the immediate communication context.

Hence, it is essential to set up a controlled framework for being able to study and model the contextual effects on language use. Generally, most linguistic studies pursuing such questions target choice phenomena such as the ordering of words within a sentence or the realization of a referring expression. Context is restricted to local relationships between sentences and the realization of entities or referents across a text. Global discourse structure as well as the range of extra-linguistics factors are not taken into account.

A typical instance of this paradigm is Centering Theory (Grosz et al., 1995) which aims at describing local coherence in a discourse. It makes predictions about which texts are more coherent than others according to their distbribution of referents. For instance, a fundamental intuition that Centering Theory tries to formalize is that coherent texts are caracterized by continuous topics, i.e. sentences that talk about the same salient discourse referent, instead of abrupt switches between topics. This general idea is illustrated by the following example texts from Kibble and Power (2004):

(1) a. Elixir is a white cream.
It is used in the treatment of cold sores.
It contains aliprosan.
Aliprosan relieves viral skin disorders.

b. Elixir contains aliprosan.
Viral skin disorders are relieved by aliprosan.
Elixir is used in the treatment of cold sores.
It is a white cream.

Text (1-a) is more coherent than Text (1-b) since it has a central topic or a salient referent, Elixir, which is introduced in the first sentence and realized as a pronominal subject in the subsequent sentences. In Centering, the most salient entity in a sentence is captured in terms of the central notion of the backward-looking center (C_b) which is basically a referential expression in an utterance that corefers with an entity in the previous utterance. A claim that Centering in its original formulation makes about realization is the following rule: if an entity in a sentence is pronominalized, it should be the backward-looking center. The following example by (Grosz et al., 1995) shows that violations of that rule lead to incoherent text:

(2) a. He has been acting quite odd. [C_b = John = referent("he")]

b. He called up Mike yesterday. [C_b = John = referent("he")]
c. John wanted to meet him urgently. [C_b = John; referent("him") = Mike]

While Centering looks at transitions between sentences in a text, other theories focus mainly on the sentence level. An important discovery, which has been treated in a huge body of linguistic literature, is that utterances are not only structured on the level of morphology, syntax and semantics, but also on a level that linguists call "information-structure" or "information-packaging". The study of information structure has originated from a variety of traditions and linguistic schools, such as (Halliday, 1967; Kuno, 1972; Chafe, 1976; Sgall et al., 1986), and is, even today, characterized by a heterogeneous and debated terminology: see Krifka (2008) or Gundel and Fretheim (2004) for an introduction into the basic notions of information structure and some related confusions.

The phenomenon that the diverse range of information structure theories try to describe can be phrased as follows: In order to be able to communicate successfully, speakers make assumptions about knowledge or information that the hearer has such that they can structure their message in a way so that the hearer can understand what the utterance is *about*, i.e. its topic. Related or alternative terms for topic would be theme, background, aboutness, or givenness. Moreover, the speaker should provide some *new* information about this topic, the element which is often called "focus", which is related to rheme, comment, newness. These two categories, the topic and the focus, are useful for describing how speakers partition the information they convey in a sentence. This brings us back to our example from Chapter 1:

(3) a. Polymers are digested in the lysosomes of eukaryotic cells.
b. The function of a lysosome is intercellular digestion of polymers in a eukaryotic cell.

In Sentence (3-a), *polymers* is the topic and expresses information about this entity in a way that it is clear that the speaker assumes the interlocutor to be familiar with it. This information flow can be directly related to the syntactic choices made by the speaker: The passivized verb in Sentence (3-a) allows the sentence-initial subject position for *polymers*, which is a typical realization of topics in English. In Sentence (3-b), the sentence-initial topic-position for *lysosomes* is achieved through the copular construction.

Similar to the local coherence paradigm, the study of information struc-

ture excludes extra-linguistic factors and focuses on concrete linguistic realizations on the sentence-level. However, the underlying theoretical notion of context is not strictly the "discourse context" defined by the surrounding sentences. Thus, it is usually stressed that a constituent can be topic or focus regardless of its discourse status and prior mentions in the text (Vallduví, 1993; Krifka, 2008). Here, we cite an example from Reinhart (1981):

(4) a. Who did Felix praise?
b. Felix praised HIMSELF.

In Sentence (4-b), *Felix* and *himself* are referring expressions for an entity known to the hearer and speaker through the previous discourse. However, the focus of the sentence, i.e. the new content, is the information that Felix praised himself. According to Gundel and Fretheim (2004), a distinction has to be made between referential givenness, meaning givenness in the discourse or the speakers mind, and relational givenness, which captures a partition of sentences into two complementary parts.

The exact definition of these information-structural notions has turned out to be a major challenge and a wide spectrum of competing formalizations has been proposed in the literature (e.g. see Lambrecht (1994); Vallduví and Engdahl (1996); Schwarzschild (1999)). For instance, a notorious difficulty for definitions of givenness is whether the information actually has to be present or mentioned in the preceding discourse (as in Schwarzschild (1999)) or whether it also relates to general, implicit knowledge that the hearer can be assumed to have (as in Lambrecht (1994)). Similarly, Krifka (2008) points out that focus should not be confused with *new* information, comment or emphasis. In his view, the most succesful approach defines it on the level of semantic interpretation: according to Rooth (1992), focus indicates the presence of alternatives that relevant for the interpretation of linguistic expressions. Others have tried to capture the subtleties of information-structural notions by defining them in terms of several dimensions, e.g. Jacobs (2001). Yet a different set of approaches addresses the question how "topic" and "focus" are marked as concrete syntactic structures in the grammatical system, such as Ward (1988), or intonational patterns in the phonological system, e.g. Büring (1997).

From an empirical perspective, a central question is how abstract notions of information packaging are instantiated in actual linguistic realizations (Féry and Krifka, 2008; Féry, 2008). Two main ways of expression

have been investigated in many languages: First, given information seems to be prosodically non-prominent and new information prosodically prominent. Second, it has often been claimed that the canonical position of given information is the beginning of a sentence such that given precedes new information, e.g. Ward and Birner (2004). However, these very general patterns should be treated with care. Féry and Krifka (2008) show that there can be drastic, cross-lingual differences in the way languages realize information structure in terms of, e.g. positions in the sentence. For instance, in the following French example, the topic of the sentence (*the apple*) is placed at the right periphery.

(5) Pierre l' a mangée, la pomme.
Peter it-ACC has eaten, the apple.
'Peter has eaten the apple.'

Thus, the Example (3), where topics always seem to occupy the sentence-initial position, does not generalize to other contexts and languages. Féry (2008), or Gundel and Fretheim (2004), make the case that a certain information-structural category should never be directly correlated to or defined on the basis of an invariant grammatical property.

Generally, the controlled set-ups of local coherence or information structural modeling have a natural counterpart in computational text generation systems: many of these systems use a so-called *surface realization* module that takes some type of meaning representation or syntactic structure as input and map it to an output sentence. The local linguistic choices that such a surface realizer has to model correspond more or less exactly to the range of phenomena that have been studied in theories of local coherence: word ordering, referring expressions and restricted types of syntactic variation such as voice alternations. As predicted by local coherence, these numerous decisions that a generation system has to take at the sentence level will have an effect on the fluency of the generation output on the text level.

In parallel to these paradigms that aim at building theories of discourse and context, linguists have always been interested in investigating particular choice phenomena, such as word order variation, and tried to make sound empirical predictions for these. One of the first important discoveries is usually attributed to Behaghel (1909) who found that shorter constituents tend to precede longer constituents in German. This pattern has also been attested in other languages, such as English (Arnold et al., 2000):

(6) a. The waiter brought the wine we had ordered to the table.
b. The waiter brought to the table the wine we had ordered.

In Example (6), it is possible to realize the direct object of the verb in a sentence-final position due to its length. However, the "canonical" order in Sentence (6-a) is also grammatical. Thus, what we observe here is a typical instance of a soft constraint: The length of a constituent in a particular context can cause deviation from the usual word order, but it is not obligatory.

It is highly language-dependent whether a syntactic choice is subject to a certain soft constraint. A famous example, due to Bresnan et al. (2001), is the active-passive alternation in Lummi and English. In Lummi, the passivization of a verb is obligatory when its subject argument is in the third person and its object argument is in the second or first person. In English, this categorical effect of person on passivization does not exist. However, there is a statistical tendency in English corpus data that mirrors this effect: The frequency of passives relative to actives is elevated when there is this hierarchical difference in person between the subject and object argument.

Next to person and constituent length, the list of known morpho-syntactic properties that impose soft constraints on syntactic choices includes animacy and grammatical relation (Aissen, 1999), definiteness (Uszkoreit, 1987), pronominality (Aissen, 1999), number (Bresnan et al., 2007), and possibly more.

What is the relation between these contextual surface cues on the one hand and information structural categories like topic and focus on the other hand? Arnold et al. (2000) firsted showed in a corpus-based analysis that both the structural complexity (heaviness) and the discourse status of a constituent have an effect on the ordering in English. A crucial methodological issue for such investigations is that surface cues like heaviness correlate with information status: given constituents tend to be realized by shorter material (e.g. pronouns), whereas new information is often realized by longer constituents. However, Arnold et al. (2000)'s study finds a statistically independent effect of these two factors. From this perspective, the information structure of a sentence, the givenness of its constituents, is one among other soft constraints or contextual factors that are relevant giving a functional explanation of phenomena like variation in constituent ordering.

These works, which focus on functional accounts and corpus-based models of particular choice phenomena, are closely related to NLG research and can sometimes be directly implemented in surface realization modules, as we will

see in the following Sections on word order and verb alternations.

3.2 Word Order

3.2.1 Morpho-syntactic Cues and Information Structure

A typical linguistic phenomenon of context-dependent variation and choice that has been investigated in a lot of languages is word order. The reason for that interest in word order is that, usually, two ordering variants of a sentence that contain exactly the same words clearly convey the same truth-conditional content, such that differences in meaning cannot account for the variation.

In this thesis, we will look at a less-configurational language, namely German. German is called less-configurational since, still, many parts of the sentence are essentially fixed in terms of their position or order, for instance the position of verbs and phrase-internal order. The major phrasal categories, verb adjuncts and arguments, can often be freely moved between the sentence-initial position (the *Vorfeld*) and a position between the finite and the non-finite verb (the *Mittelfeld*). This is illustrated in Example (7). [1]

(7) a. Er hat Tiger auf der Straße gesehen.
He has tigers on the street seen.
b. Auf der Straße hat er Tiger gesehen.
On the street has he tigers seen.

[1]Note, however, that for instance the position of so-called focus-sensitive particles can have an effect on the truth-conditional effect on the interpretation of the sentence (a similar effect could be achieved by certain prosodic markings):

(i) a. Er hat nur Tiger auf der Straße gesehen.
He has only tigers on the street seen.
'He has seen only tigers on the street.'
b. Er hat Tiger nur auf der Straße gesehen.
He has tigers only on the street seen.
'He has seen tigers only on the street.

In this thesis, we will not consider the interpretation of such particles, but we are interested in the relative orderings of argument and adjunct phrases.

c. Tiger hat er auf der Straße gesehen.
Tigers has he on the street seen.
'He has seen tigers on the street.'

The influence of a range of contextual factors on German word order has been early discovered in the literature. Lenerz (1977) and Höhle (1982) both use these factors to explain deviations from a word order that they assume to be canonical or unmarked. For instance, Lenerz (1977) claims that there is a standard or canonical order for indirect and direct objects in the German middlefield, illustrated in (8).

(8) a. Er gab dem Schüler das Buch.
He gave the student the book.
b. *marked:* Er gab das Buch dem Schüler.
marked: He gave the book the student.
'He gave the book to the student.'

This canonical word order can be overridden by a range of factors related to definiteness, syntactic complexity, or agentivity. Thus, if the indirect object is indefinite or syntactically complex, it is more likely to follow the direct object:

(9) Er gab das Buch dem sehr fleißigen, immer pünktlichen Schüler.
He gave the book the very diligent, always accurate student.

The seminal work by Uszkoreit (1987) was one of the first who proposed a formalized account of German word order implementing the interaction between syntactic and pragmatic constraints. His formalization does not rely on a canonical word order, but it computes the order for a given set of constituents based on weighted, partially ordered linear precedence rules. He suggests the following basic principles that influence the order in the Vorfeld and Mittelfeld:

- constituents in the nominative case precede those in other cases, and dative constituents often precede those in the accusative case
- focused constituents precede non-focused constituents
- personal pronouns precede other NPs
- light constituents precede heavy ones

Note that these principles define partial orderings, i.e. there are defined as relative precedences between certain binary properties, not as absolute rules that order particular constituents.

A notorious problem for models of German word order are empirically valid characterizations of the prefield position. The standard paradigm which assumes that given precedes new information would predict that the prefield position is filled by the topic (or given information) of a sentence. This generalization clearly does not hold, as illustrated by the following utterance situation, taken from Frey (2004) (capitalization marks a prosodic stress):

(10) a. Wer geht in die Oper?
Who goes to the opera?
b. PAUL geht in die Oper.
Paul goes to the opera.

In Sentence (10-b), the phrase *Oper* is the topic, and *Paul* is the focus or new information. The sentence-initial position of the focused constituent sounds perfectly natural. Consequently, Frey (2004) has proposed that the designated topic-position in German is in the middle-field. Generally, the order within the German middle-field seems to be easier to predict than the filling of the prefield position. For instance, in the presence of pronominalized elements it is not possible to realize a non-pronominal constituent to the left of the middlefield, however non-pronominal elements can clearly figure in the prefield when pronouns are present in the middlefield:

(11) a. Gestern hat er ihn in der Oper gesehen.
b. ??Gestern hat er in der Oper ihn gesehen.
c. ??Gestern hat in der Oper er ihn gesehen.
d. In der Oper hat er ihn gestern gesehen.

Büring (2001) observes similar patterns for the precedence of definite and indefinite NPs in the middlefield:

(12) Wem hast du ein Buch gegeben?
a. Ich habe dem Schüler ein Buch gegeben.
b. *Ich habe ein Buch dem Schüler gegeben.

In Example (12), it is not possible to realize the topic at the left position of the middlefield, preceding the focus, because it is indefinite. The same

contrast does not hold if the topical element is definite:

(13) Wem hast du das Buch gegeben?
  a. Ich habe dem Schüler das Buch gegeben.
  b. Ich habe das Buch dem Schüler gegeben.

The patterns are even less clear, when the prefield is taken into consideration, as indefinite or non-pronominal elements can naturally fill the position when definite and pronominal elements are located in the middlefield. Based on an empirical studies, Speyer (2005) and Dipper and Zinsmeister (2009) find that the preferred filler of the German Vorfeld are brand-new constituents or scene-setting elements.

In terms of corpus studies, Kempen and Harbusch (2004) and Weber and Müller (2004) also investigate a wide range of contextual factors and information structural properties as soft constraints on German word order, such as the syntactic function, length, definiteness, animacy, topicality of the involved constituents. In general, its seems that the effect of these different factors is subject to complex interactions. For instance, Weber and Müller (2004) find that, on the one hand, "SVO" orders in a German corpus mostly confirm the expected patterns, i.e. definite subjects tend to precede indefinite objects, pronominal subjects precede full NP objects, etc. On the other hand, the observed "OVS" orders did not confirm any of these general patterns.

Similar complexities have been observed in other languages with a prefield position, such as Dutch. Bouma (2008) investigates the influence of three factors, grammatical function, definiteness and grammatical complexity, on the choice of a prefield occupant. Based on logistic regression modeling, he shows that all factors have an influence, but the correlations found are more or less direct and complex depending on the factor. For instance, whereas grammatical functions can be clearly arranged on a scale capturing their likelihood of appearing in the prefield (subject $\prec$ indirect object $\prec$ direct object), the tendencies for definiteness are more differentiated, especially for different types of pronouns.

Next to these morpho-syntactic properties of verb arguments, Wasow (1997) points out that factors related to idiomaticity and semantic connectedness between the verb and its argument also play a role. He finds that idioms like *take into account* occur more frequently with shifted orders than non-collocational, semantically transparent verb preposition pairs. These lexical factors, however, have found less attention in theoretical studies of

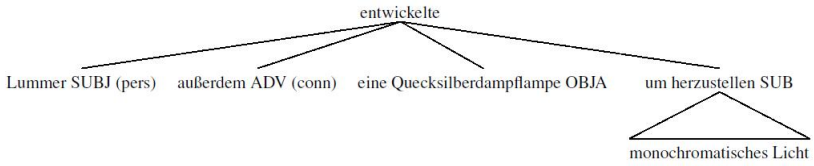


Figure 3.1: Input representation for generating constituent orders in German from Filippova and Strube (2007a)

syntactic variation.

3.2.2 Features for Surface Realization

The prediction of word order is a standard problem that has to be adressed in NLG systems. While theoretical studies of word order variation often look at the position of specific constituents, e.g. subjects and objects in the Vorfeld, or binary alternations such as the Heavy NP shift, NLG systems typically deal with all possible orderings of constituents in a given input representation. For instance, Filippova and Strube (2007a) assume an input like the tree in Figure 3.1, where constituent internal word order of NPs and PPs as well as the position of verbs is specified and the algorithm orders phrases in the German prefield and middlefield. Some approaches go further and take a completely unstructured bag of words as input (Langkilde and Knight, 1998; Wan et al., 2009). Most frequently, surface realizers compute word orders from some syntactic structure such as TAG trees (Bangalore and Rambow, 2000b), LFG F-structures Cahill et al. (2007a), HPSG or CCG representations (White, 2004; Velldal and Oepen, 2006) or syntactic dependency trees (Bohnet et al., 2010).

While lexical factors accounting for word order variation have been less central in theoretical research, they are often used a basic model for surface realization. Thus, Langkilde and Knight (1998) apply n-gram statistics from a language-model to compute the realization candidate that is most likely for a given generation input. In a language model, context is represented by the local relationships of a word to its preceding and following words. A big practical advantage of this approach is that it does not rely on data specifically produced for a particular generation set-up as the language model

can be trained on any given corpus. A similar approach is pursued by Bangalore and Rambow (2000b). White (2004) integrates n-gram scores in the OpenCCG realizer as means of pruning and ensuring efficiency for generation with a large Combinatory Categorial Grammar.

Subsequent work explored ways of exploiting linguistically annotated data for trainable generation models. Ratnaparkhi (2000) uses a corpus of user queries in the air travel domain which is annotated with templates to automatically learn the fillings of these templates. He implements a maximum entropy model that exploits a number of different feature functions including dependency information. In the Amalgam system (Corston-Oliver et al., 2002; Ringger et al., 2004), a sentence realizer traverses an input logical form graph transforming it into a surface tree by applying a number of decision tree classifiers for different linguistic decisions. The underlying features also go beyond n-gram scores and incorporate lexical and syntactic features of the context of a decision.

The importance of deep linguistic features for surface realization ranking has been systematically shown to improve n-gram based language models in the context of generation with broad-coverage reversible grammar implementations. Velldal and Oepen (2006) use a large, reversible HPSG grammar for English to first produce a symmetric treebank: Each sentence in the corpus is annotated with an HPSG analysis and all its possible paraphrases generated by the grammar. For every paraphrase, the realization ranker can extract a number of deep syntactic features from the corresponding attribute-value matrix. The implementation of their feature functions is based on templates that combine certain syntactic configurations (labels of syntactic functions, modifiers, features for number, person) with n-gram scores. Cahill et al. (2007a) adopt this architecture for surface realization with the broad-coverage German LFG grammar.

For the automatic prediction of German constituent order, Filippova and Strube (2007a)'s study provides corpus-based evidence that, besides linguistically informed modeling, it is important to deal with the German prefield position in a special way designing a classifier that separates the prefield from other prediction problems. A list of their features is given below:

- lemma and PoS tag of the word and its head
- syntactic function, semantic class
- constituent length in words

- number and types of modifiers
- features related to determiners
- features related to pronouns

Many of these properties have been related to the realization of word order in the theoretical literature. For instance, Arnold et al. (2000) shows that short constituents tend to precede long or heavy constituents, Büring (2001) shows that constituents with a definite article precede indefinite ones, etc. However, in the corresponding computational approaches, these features are not necessarily chosen and analyzed systematically with respect to the underlying theoretical concepts of context. Rather, an effective modeling strategy seems to be to take most of the properties that are available in the input and let the classifier or ranker decide about the usefulness.

Cahill and Riester (2009) provide a more systematic account of sentence-internal approximations of context. They use a small corpus of German radio news that is annotated with Information Status categories to extract frequent asymmetries in the order of IS categories. They incorporate these asymmetries in the LFG-based surface realizer described in Chapter 2.2 and improve over the naive sentence-internal model with linguistic features. As an example, they find that constituents annotated with "D-GIVEN-REL" precede constituents annotated with "NEW" in 89% of the occurrence of such a pair in the corpus. Since the automatic prediction of IS labels has still moderate accuracy, they approximate the IS labels with morpho-syntactic properties extracted from the LFG f-structures. For instance, they find that the IS label "NEW" often correlates with the syntactic feature "SIMPLE_INDEF". Thus, finally, the asymmetries between IS labels are mapped to asymmetries between morpho-syntactic features in their generation ranking model. They show that this informed way of incorporating these morph-syntactic features is more effective than simply taking all possible feature combinations. This results leads to the conclusion that sentence-internal, morpho-syntactic properties can provide a reasonable approximation of complex, discourse-related notions such as Information Status. Their extremely fine-grained syntactic annotation scheme is presented in Figure 3.2.

de Kok (2010) applies automatic feature selection methods to a sentence realizer for Dutch, showing that only a small number of features are actually informative given a large set of instantiations for linguistically informed feature templates.

Precedence features:

PERS PRON precedes INDEF ATTR
PERS PRON precedes SIMPLE INDEF
DA PRON precedes INDEF ATTR
DA PRON precedes SIMPLE INDEF
DEMON PRON precedes INDEF ATTR
DEMON PRON precedes SIMPLE INDEF
GENERIC PRON precedes INDEF ATTR
GENERIC PRON precedes SIMPLE INDEF

Syntactic types used for precedence features:

	Definites
SIMPLE DEF	simple definite descriptions
POSS DEF	simple definite descriptions with a possessive determiner
DEF ATTR ADJ	definite descriptions with adjectival modifier
DEF GENARG	definite descriptions with a genitive argument
DEF PPADJ	definite descriptions with a PP adjunct
DEF RELARG	definite descriptions including a relative clause
DEF APP	definite descriptions including e.g. an apposition
	Names
PROPER	combinations of position/title and proper name (without article)
BARE PROPER	bare proper names
	Demonstrative descriptions
SIMPLE DEMON	simple demonstrative descriptions
MOD DEMON	adjectivally modified demonstrative descriptions
	Pronouns
PERS PRON	personal pronouns
EXPL PRON	expletive pronoun
REFL PRON	reflexive pronoun
DEMON PRON	demonstrative pronouns (not: determiners)
GENERIC PRON	generic pronoun (man – one)
DA PRON	"da"-pronouns (darauf, darüber, dazu, . . .)
LOC ADV	location-referring pronouns
TEMP ADV,YEAR	Dates and times
	Indefinites
SIMPLE INDEF	simple indefinites
NEG INDEF	negative indefinites
INDEF ATTR	indefinites with adjectival modifiers
INDEF CONTRAST	indefinites with contrastive modifiers (einige – some, andere – other, weitere – further, . . .)
INDEF PPADJ	indefinites with PP adjuncts
INDEF REL	indefinites with relative clause adjunct
INDEF GEN	indefinites with genitive adjuncts
INDEF NUM	measure/number phrases
INDEF QUANT	quantified indefinites

Figure 3.2: Feature model from Cahill and Riester (2009) for German surface realization ranking

3.3 Verb Alternations

3.3.1 Meaning vs. Function

When it comes to analyzing the discoursive or pragmatic function of verb alternations such as the active-passive voice or nominalization, the effect of information structure and its interaction with other semantic and grammatical factors is even harder to tease apart than for the domain of word order variation. A fundamental difference between word order and verb paraphrases is that the second alternation type involves a change in the morphology of the predicate and the grammatical functions of its corresponding semantic roles. In order to form a passive in languages like German and English, the agent is deleted or demoted to an oblique argument and the patient is promoted to a subject. A very basic intuition is that speakers use this syntactic operation in contexts similar to those where certain word order variations occur (Halliday, 1967). Thus, the passive would be an alternative device to topicalizing or foregrounding the patient of a verb (Givón, 1994).

However, the syntactic and morphological realization of verb alternations differs widely across languages. In the case of passives, certain languages do not have the structure, some languages restrict it to certain verb types (such as English and German to transitive verbs), other languages' grammar exhibits a wide range of different passive forms that can be applied to basically all types of verbs (Keenan and Dryer, 2007). These differences in productivity relate to general syntactic and morphological properties of the language. For instance, Keenan and Dryer (2007) give the example of Malagasy syntax that does not allow for the formation of object relative clauses. If a Malagasy speaker wants to realize a relative clause like the English *the clothes that John washed*, she has to passivize the verb as in *the clothes that were washed by John* such that the relative pronoun is the subject of the relative clause. Moreover, diachronically, the passive structure evolved from very different original structures across languages, e.g. from adjectival statives in English, from reflexives in Spanish or from left dislocation (Givón, 1994).

Aissen (1999, 2003) has formalized the predictions for the usage of argument alternations on the basis of markedness hierarchies in an OT framework. These hierarchies associate argument functions and certain properties for person, animacy, or definiteness with universal rankings; e.g., subjects tend to be higher on the person scale than objects, 1st person outranks 3rd person. These hierarchies can be combined via alignments and constraints on these

alignments. She discusses a number of languages where certain alignments of these rankings determine the grammaticality of a sentence. For instance, in the Lummi language, the combination of a 3rd person subject and a 1st person object is ungrammatical due to language-specific alignment constraints. In English, the usage of passive voice is explained on the basis of a thematic prominence scale and a function scale, i.e. more prominent elements need to be promoted to the subject position.

However, as we have observed in our above discussion of information structural notions, it is typically very difficult to assign precise and operationalizable definitions to properties like "prominence" or "topicality". Similar to the different approaches to the definition of topic in the previous Section, Tomlin (1983) relates the prominence of themes in passives to their individual attention state, whereas Thompson (1987) defines the increased topicality of a theme in a passive with previous mentions in the discourse. Bresnan et al. (2001) find that there is a statistical tendency in English to passivize a verb if the patient is higher on the person scale than the agent, a result which is closely related to the markedness hierarchies described by Aissen (1999). Thus, as for the case of word order, a number of different factors seem to contribute.

A maybe even harder problem for approaches that define pragmatic constraints on passives in English might be the fact that the construction clearly allows for the omission of the agent verb argument. Thus, it has been argued that the passive should not be analyzed as a promotional grammatical device that accounts for marked properties of the theme, but rather as a demotional device used in contexts when the agent is not important or "defocused" and has similarities with other intransitive constructions such as reflexives (Shibatani, 1985). As an elegant consequence, this approach can explain that certain languages can derive passives from intransitives, meaning that the event did not involve an agent.

3.3.2 Statistical Accounts

A well-known corpus-based account that models contextual usages of a verb alternation is the seminal work by Bresnan et al. (2007) which focuses on the English dative alternation. This phenomenon actually circumvents the problem of argument omission or demotion as it varies the realization of a recipient as an indirect or prepositional object:

(14) And I said, I want a backpack.
I told him, if you want to give me a present for Christmas ...
a. ... give me a backpack.
b. ... give a backpack to me.

Bresnan et al. (2007) correlate the use of the alternation to a number of context factors in a statistical multiple regression model. Most of the factors relate to surface cues of the recipient and the the object: pronominality, definiteness, constituent length, animacy, concreteness, number and person. Additionally, the model contains an information-structural factor, i.e. the "discourse accessibility" of the verb arguments. Based on these - manually coded - factors, the model achieves a high accuracy on the data and all the coefficients are significant, meaning that they explain some variaton in the data independently of other factors. Moreover, Bresnan et al. (2007) shows that the coefficients computed for the modeling factors mirror the markedness hierarchies established in cross-lingual research (Aissen, 1999).

Bresnan et al. (2007) see these results as evidence for the fact that speakers use the dative alternation to achieve a particular precedence of verb arguments: first-person, pronominal, discourse-given recipients (*me* in (14)) tend to precede nominal, discourse-new themes (*backpack*).

Interestingly, Bresnan and Ford (2010) report on an updated version of the dative alternation model where some of the surface cues have been recoded and corrected. In this version, the "discourse accessibility" predictor drops out because it is not significant anymore. This demonstrates that contextual factors and morpho-syntactic properties are highly correlated, and have to be coded with care in order to achieve sound empirical results.

A similar study is done by Levy and Jaeger (2007) who propose a multifactorial account of *that*-clause reduction. It is one of the rare examples where a phenomenon that involves the omission of linguistic material is addressed. They show that the insertion or omission of *that* is, among other factors, correlated to the information flow in a sentence such that more linguistic material is used in statistically less predictable contexts. They develop the notion of *information density*, using the term "information" in an information-theoretic sense. Their theory predicts that speakers should adapt the structure or modulation of parts of their utterance in accordance with its predictability, in order to avoid peaks and troughs in the amount of information that is transmitted per linguistic unit. They show that this effects holds for *that*-clause reduction in English: the more predictable an

embedded clause, the more likely speakers will omit the function word *that*. Similar to Bresnan et al. (2007), they propose a multiple regression model for their data, acknowledging the independent effect of a range of soft constraints that influence the choice.

Rajkumar and White (2011) integrate features inspired from Levy and Jaeger (2007) into a corpus-based, large-coverage surface realizer. They show that a feature that computes a contextual information density score improves the prediction of *that*-omission in a corpus-based generation experiment. Otherwise, we are not aware of approaches that adress verb alternations at scale comparable to word order prediction. A recent exception is Bohnet et al. (2010) who try to exploit representations stemming from semantic role annotations for surface realization beyond standard syntactic phenomena. However, they discover a number of problems with the annotations. It is not clear to what extent this generation setting actually captures alternations beyond word order.

3.4 Referents

3.4.1 Givenness and Forms of Referring Expressions

As compared to syntactic paraphrases such as word order and verb alternations where different levels of linguistic structure and pragmatic context effects are closely intertwined, the form of referring expressions used for (several) mentions of a referent in a discourse has been clearly and almost exclusively related to the contextual, discoursive or informational status of that referent. In her foundational work, Prince (1981) develops of taxonomy of notions related to newness and givenness of information in a discourse. Some subtle distinctions captured by her hierarchy are illustrated in the Examples (15-a-b). According to Prince (1981), *I* would be a given, situationally evoked discourse referent. The phrase *a dress* in (15-a) refers to a new, namely brand new referent, which is established or introduced in the discourse. In contrast, *Chomsky* in Example (15-b) is a new, unused referent which is known to the hearer but not previously mentioned.

(15) a. I bought a dress.
b. Chomsky is famous.

There has been a substantial amount of work trying to relate Prince (1981)'s Givenness Hierarchy , or some variant thereof, to surface forms that identify referents in a text (Ariel, 2001; Gundel et al., 1993; Givón, 1983). An example pattern that such theories try to predict is illustrated in (16) and (17) (taken from (Gundel et al., 1993)):

(16) My neighbor's bull mastiff bit a girl on a bike.
   a. It's the same dog that bit Mary Ben last summer.
   b. That's the same dog that bit Mary Ben last summer.

(17) Sears delivered new siding to my neighbors with the bull mastiff.
   a. *It's the same dog that bit Mary Ben last summer.
   b. That's the same dog that bit Mary Ben last summer.

According to Gundel et al. (1993), the *bull mastiff* has a different cognitive status in (16) and (17). Whereas in (16), the phrase is introduced in a subject position and has the status of o focused referent. Therefore, both pronominal forms *it* and *that* can be used. In (17) where *bull mastiff* is not focused, it simply restricts the reference of some head referent, the unstressed pronoun *it* does not seem appropriate.

The problems that such a theory faces are very similar to those we have discussed for information-structural accounts of word order: What are the factors or components that correspond to a particular notion of givenness? How can these notions be operationalized to analyze empirical examples? How do we characterize the relationship between a particular theoretical definition of givenness and its possible articulations in the form of referring expressions? For instance, from the examples seen in (16) and (17), a possible conclusion would be that the grammatical function (i.e. subject position) determines the focused status of a referent. This status, in turn, would predict the felicity of certain expressions. However, Gundel et al. (1993) states that the status of a referent in a discourse depends on a range of pragmatic factors (which are not exhaustively described). For instance, the authors state that the grammatical function is clearly not the only determining factors. As an example, see (18) where the phrase *a large wind energy project* has the same syntactic function as the *bull mastiff* in (17), but a pronominal reference with *it* is possible.

(18) However, the government of Barbados is looking for a project manager for a large wind energy project.

a. I'm going to see the man in charge of it next week.

Recently, Prince (1981)'s Givenness Hierarchy has been used as a theoretical basis for developing annotation schemes that mark *information status* of referents in actually occuring corpus texts. Thus, Riester (2008) suggests an empirically applicable inventory of information status categories considering theoretical motivations from DRT (Kamp and Reyle, 1993), annotated on a corpus of German radio news Riester et al. (2010).

Concerning the relationship between a referent's status and the form of referring expression, a wide-spread assumption is that prominent, focused or accessible referents tend to be referred to by unstressed or less specific forms, whereas specific forms are used for new, non-prominent, less accessible referents. This hypothesis basically predicts that unstressed pronouns are used for given referents that were recently mentioned in the preceding discourse. However, it is clear that there are many exceptions and counterexamples to this. For instance, Vonk et al. (1992) shows that "overspecific" referring expressions have an important function for discourse coherence, i.e. they are able to mark boundaries in a text. As an example, Vonk et al. (1992) give the following text which always refers to its main protagonist with a personal pronoun, but does not read very natural:

1. Sally Jones got up early this morning.
2. She wanted to clean the house.
3. Her parents were coming to visit her.
4. She was looking forward to seeing them.
5. She weighs 80 kilograms.
6. She had to lose weight on her doctor's advice.
7. So she planned to cook a nice but sober meal.

One way to make the above text more coherent, would be to insert an overspecific reference to *Sally* in Sentence (5) so as to indicate a new theme of the text starting in that sentence. This clearly shows that referring expressions do not only serve the function of identifying the entities mentioned in a text, but also contribute to the general naturalness and perceived coherence of a discourse.

Important formalizations for modeling relationships between discourse coherence, referring expressions and transitions between discourse segments have been developed in the framework of Centering Theory (Grosz et al.,

1995). This framework has been theoretically (and also computationally) implemented for different languages with a number of different underlying definitions and claims associated with these (Poesio et al., 2004). The basic ingredients of Centering are the so-called discourse *centers*, the referents of a text, which are are ordered by a ranking function. The set of all referents contained in an utterance and ordered by that function is the set of forward-looking centers. A backward-looking center is a referent that is contained in the current utterance and the set of forward-looking centers of the previous utterance. If two utterances in a row have the same backward-looking center, the transition between them is characterized as a continuation, other transitions are center retaining and center shifting. Within this framework, it is possible to state constraints on relations between the realization and ranking of centers, good transitions between discourse segments and the way centers are chosen and arranged in a discourse. For instance, the famous *Rule 1* defined by Grosz et al. (1995) states that pronominalization only occurs for backward-looking centers, or other centers if the backward-looking center is pronominalized as well. Note, however, that this rules yields only partial predictions for a small number of occurrences of referents in a text.

3.4.2 Context in REG

An interesting challenge for REG research is to scale theoretical predictions to a much wider set of choices, i.e. types of referring expressions, and to a wider set of contexts. For instance, the actual core of Centering Theory only makes one claim about pronominalization in Rule 1 which applies to two subsequent utterances that share more than one referent and the referent which is not the backward center is pronominalized.

However, due to the fact that a considerable body of REG research has been done in the 'distractor paradigm' established by Dale (1989) and Dale and Reiter (1995), the role of wider discourse context in REG has not been a prominent research topic in the field.

McCoy and Strube (1999) were one of the first to adress corpus-based prediction of pronominalization. They point out limitations of theories based on salience for predicting occurrence of pronouns and definite descriptions in corpus text, showing that especially pronouns occur much less often in corpus text than would be predicted by theories. One of their central examples is the text in (19) where every utterance expresses information about the same person entity.

(19) When **Kenneth L. Curtis** was wheeled into court nine years ago, mute, dull-eyed and crippled, it seemed clear to nearly everyone involved that it would be pointless to put **him** on trial for the murder of **his** former girlfriend, Donna Kalson, and the wounding of her companion.
It had been a year since **Mr. Curtis** had slammed **his** pickup truck into them, breaking their legs. **He** then shot them both and, finally, fired a bullet into **his** own brain. **Mr. Curtis** fingered in a coma for months, then awoke to a world of paralysis, pain and mental confusion from which psychiatric experts said **he** would never emerge. One expert calculated **his** I.Q. at 62.

Although *Mr. Curtis* could be regarded as the backward center in all the sentences, also definite noun phrases are used as referring expressions by the author. Moreover, McCoy and Strube (1999) point out that there are many references to the entity which do not occur in an argument position, e.g. possessive pronouns, a phenomenon which is not treated in Centering theory at all.

In line with psycholinguistic research (Vonk et al., 1992), they argue that referring expressions also serve the discourse function to indicate boundaries in the narrative structure of the text. As an approximation of discourse boundaries, they use shifts in time scale which are often signaled by overt linguistic means such as temporal phrases. They design an algorithm for predicting the occurrence of definite descriptions versus pronouns that takes into account the following criteria:

- sentence boundaries
- distance to last mention
- discourse boundaries (so-called thread changes)
- ambiguities with preceding antecedents

The main idea of the ambiguity criterion is that pronouns are avoided if there are several possible antecedent that the pronoun could be referring to. Henschel et al. (2000) take up the task of predicting occurrence of pronouns and add two further criteria to their algorithm: subject-hood of a referent in the previous utterance and parallelism of syntactic configurations between

two subsequent utterances. The parallelism criterion captures cases in their data set where a discourse-new non-subject referent is realized as a pronoun in the subsequent utterance if it has the same function there. For each utterance, their algorithm computes a local focus, which is the set of referents that could be realized by a pronoun. Thus, their approach to sentence-external context again relies on some basic ideas from Centering Theory.

In the more recent approach to the REG task that we will also pursue, the choice of subsequent referring expressions is not reduced to a binary decision between pronoun and definite description. Depending on the original corpus text, several candidates of slightly different descriptions and pronoun types have to be assigned to the correct slot in the text. Greenbacker and McCoy (2009) address this task with a rich feature model that is inspired from linguistic and psycholinguistic research. The basic factors that their feature model takes into account are very similar to McCoy and Strube (1999). They implement a decision tree classifier with various feature subsets and find that a classifier that uses a rather limited subset of the sentence-external features has the highest performance. They list the following features for their most effective classifier:

- syntactic category of the referent instance
- semantic category of the referent
- was entity subject of last sentence?
- sentences since last reference
- instance follows "and","","but","or","then","?"
- instance between "","&", "and"

Note that 4 out of these 6 features do not relate to the sentence-external context but constitute either properties of the referent or properties of the immediate local context.

3.5 Summary

In this Chapter, we have seen numerous ways of describing and formalizing the status of contextual notions for capturing linguistic choice phenomena.

Broadly speaking, there are two main perspectives on the problem: On the one hand, researchers are interested in contextual structure as a level of linguistic description, similar to syntactic or semantic analysis. This approach entails that theoretical categories such as topic, focus or centers, have to be established and correlated with linguistic structures on the level of syntax or prosody. On the other hand, researchers have been interested in capturing and predicting the occurence of particular alternatives of a choice phenomenon in realistic language data.

While the status of the higher-level notions such as the information-structural focus is still widely debated in the literature, theoretical and applied approaches focusing on predicting particular choice phenomena have adopted strikingly similar methods and reached comparable conclusions.

Thus, using the framework of regression modeling, it has been shown for several choice phenomena that multiple context factors have an independent effect on a particular choice. Bresnan et al. (2007) show that the weights (or correlation coefficients) computed by the regression model for factors such as pronominalization, animacy, or syntactic complexity of the verb arguments confirm markedness hierarchies developed in typological research (Aissen, 1999). For instance, it seems to be a crosslinguistically stable finding that pronominal arguments are less marked than nominal arguments: whereas some languages encode these hierarchies directly in the morphosyntactic system, others display the same pattern in corpus-based frequencies. These hierarchies also seem to relate to word order. Using a regression model approach, Bouma (2008) shows that these tendencies hold for prefield occupancy in Dutch.

In parallel to these theoretical advances, some recently proposed generation approaches, in particular those dealing with word order variants, have made some progress towards reaching a good output quality that reflects human preferences to some extent. An important methodological insight of these approaches is that huge amounts of corpus data can be successfully exploited for the automatic acquisition of models of linguistic choice. Even very basic n-gram statistics have been shown to be useful for selecting between alternative sentences since they inherently capture preferences for certain surface patterns (Langkilde and Knight, 1998). Other, more sophisticated approaches have shown that certain information-structural notions can be approximated by carefully designed linguistically informed classifier or ranking models (Cahill and Riester, 2009). In these models, the multitude of competing factors that have an effect on the appropriateness of a certain

choice are represented as sets of properties which are weighted by the learning algorithm. These weights computed on a training corpus serve to make predictions for unseen generation inputs.

Due to these similarities, it is straightforward to implement certain theoretical predictions in computational generation systems. Some examples for NLG studies aiming at testing theoretical findings can be found in the literature: Rajkumar and White (2011) integrate features inspired from Levy and Jaeger (2007) into a surface realizer for CCG. Filippova and Strube (2007a) implement features taken from Uszkoreit (1987) and other works on German word order, testing them in their corpus-based model of German constituent order. Generally, in the surface realization domain, state-of-the-art systems exploit various kinds of "linguistically motivated" features (Ringger et al., 2004; Filippova and Strube, 2009; Cahill and Riester, 2009) which are typically extracted as general templates (Velldal and Oepen, 2006; de Kok, 2010; Bohnet et al., 2010). In corpus-based REG, systems adopt a similar approach, although here, the findings are a little bit less promising: the analysis of empirically well performing features in statistical models often reveals that notions taken e.g. from Centering theory and other theoretical predictions are vague in terms of the implementation and sometimes less effective than could be expected (Henschel et al., 2000; Greenbacker and McCoy, 2009).

Finally, an important question remains open when looking at the recent success of multi-factorial accounts for the prediction of choice: What is the influence of discourse context and information-structural factors which are hard to annotate and recognize next to the range of morpho-syntactic cues which can be easily assessed in syntactic annotations of corpus data? While previous theoretical accounts seem to provide evidence that information-structural properties co-exist with other surface-oriented factors, as argued by e.g. (Arnold et al., 2000), recent models achieve high prediction accuracy without any discourse-oriented factors (Bresnan and Ford, 2010; Bouma, 2008). The same holds for the linguistically-informed feature models used in corpus-based surface realization and referring expression generation. These models exclusively rely on sentence-level, surface-oriented morpho-syntactic cues. Even Cahill and Riester (2009)'s model, which is based on data annotated with information status categories, uses only morpho-syntactic approximations of these factors.

This question is the underlying motivation for the corpus-based generation experiments in the following Chapter 4 which deals with sentence-internal and sentence-external factors for context modeling.

Chapter 4

Context Modeling in the Wild

The aim of this thesis is to predict and generate linguistic alternations whose usage varies with the context and is subject to soft constraints not captured in the core grammatical model of a language. Intuitively, it could be expected that the modeling of soft constraints for predicting choice in a given sentence is a matter of describing and formalizing sentence-external, text-external or global context factors. Theroetically, however, the previous Chapter 3 has shown that it is generally difficult to formalize context factors for making predictions of language use and to operationalize the notions developed in theories of information-structure or even Centering.

In Chapter 3, we have also looked at a range of linguistic studies that model particular choice phenomena like the dative alternation or particular morpho-syntactic cues like "heaviness of an NP". In these studies, it turns out that the contextual variation can be modelled relatively reliably by taking into account a multitude of lexical, morpho-syntactic, and other factors (e.g. definiteness, animacy, person, constituent length) that mostly stem from the immediate sentence-internal material. These factors have also been successfully implemented in generation tasks that use very detailed input representations where actual lexical and referential choices are fully fixed for modeling syntactic choice and can be exploited in the contextual model, e.g. in Cahill and Riester (2009).

Thus, the relation between sentence-internal factors and morpho-syntactic surface cues on the one hand, and theoretical notions of context, information structure and discourse status on the other hand turns out to be intricate. If the sentence-internal generation inputs provide very specific information and pre-determine a range of choices, many relevant discourse factors are

reflected indirectly in properties of the sentence-internal material. Most notably, knowing the shape of referring expressions narrows down many aspects of givenness and salience of its referent, and consequently, its syntactic position in the sentence. Thus, pronominal realizations indicate givenness, and in German there are even two variants of the personal pronoun (*er* and *der*) for distinguishing salience. Similarly, knowing the grammatical function of a constituent and its position in a complex sentence can narrow down its possible morpho-syntactic surface forms. For instance, in the German *Mittelfeld*, pronouns cannot follow definite or indefinite NPs.

There have beend countless corpus-based approaches to modeling soft contextual constraints for generating word order variation and choice of referring expressions in German and other languages (e.g. (Ringger et al., 2004; Velldal and Oepen, 2006; Belz and Varges, 2007; Filippova and Strube, 2007a; Cahill et al., 2007a; Dipper and Zinsmeister, 2009)). However, these approaches have almost exclusively looked at very controlled generation tasks where the generation input predefines other interacting and competing choices in the sentence. Thus, surface realisation tasks use inputs where the morpho-syntactic realisation of the constituents and referents of a sentence is fully specified. The GREC (*Generating referring expressions in context*) task has been designed such that the REG systems predict the morpho-syntactic realisation of referents in a fully-specified, linearised surface text that simply contains slots where referents have to be inserted. As a result, generation systems in both tasks can exploit a rich inventory of sentence-internal surface cues that are systematically related to the sentence-external context or information structure.

These observations give rise to two questions that will be investigated in this Chapter:

- In the light of the difficulty in obtaining reliable discourse information on the one hand and the effectiveness of exploiting the reflexes of discourse in the sentence-internal material on the other – can we nevertheless expect to gain something from adding sentence-external feature information?

- In the light the fact that surface realisation and GREC/REG systems are typically trained on highly specified generation inputs – to what extent do the resulting models for word order variation and referring expressions exploit the interaction between syntactic and referential

choice? Can we expect that these models will generalize to "deeper" and more applied generation settings?

Experiments and Frameworks In order to investigate the above questions, this Chapter presents and discusses three experiments in standard corpus-based generation settings, i.e. surface realisation and GREC. In both settings, we will start from state-of-the-art, rich, sentence-internal models and extend them to incorporate sentence-external context factors. We provide a detailed evaluation and analysis of the effect of sentence-external contextual factors in these two complementary frameworks. However, our basic approach to operationalizing context will be the same in the different systems: As explained in Section 4.1, we exploit representations of entity transitions in a text for extracting information about the discourse status of a referent or constituent.

Experiment 1 in Section 4.2 describes a constituent ordering experiment where we analyze the interaction between accurate sentence-external factors and a set of rich, high-quality types of sentence-internal factors. Experiment 2 in Section 4.3 presents a surface realization ranking experiment where sentence-external context is approximated through relatively shallow representations of entity transitions or chains. Experiment 3 in Section 4.4 compares two settings for referring expression generation (REG) with respect to the effects of available syntactic information.

4.1 Entity Transitions as Sentence-external Context

While there would be many ways to construe or represent discourse context (e.g. in terms of the global discourse or information structure), we concentrate on capturing local coherence through the distribution of discourse referents. The instances of discourse referents in a text basically correspond to the constituents that our surface realization model has to put in the right order, or that the REG model has to assign to appropriate referring expressions. As the order or realization of referents is arguably influenced by the information structure of a sentence given the previous text, our main assumption is that information about the prior mentioning of a referent can be operationalized for these generation tasks.

In previous work, the entity-based approach to context has been shown to be successful for the tasks of summarization or content ordering. A widely used representation for intersentential relations between discourse referents is the entity grid, introduced in Barzilay and Lapata (2008). The entity grid model represents the distribution of referents in a discourse with the help of a matrix where the columns correspond to the referents and the rows to the sentences of the text. The cells of the matrix record information about the occurrence of a particular referent in a sentence, and possibly further linguistic information such as syntactic function, position or lexical realization. Figure 4.1 illustrates an entity grid for an example text taken from the robbery corpus (see Chapter 2.4.2 and 5.4).

Barzilay and Lapata (2008) (and many others) use the matrix to compute a vector with entity transition features that represents the coherence of a document. They show that a sentence ordering model can learn to distinguish between vectors extracted from documents with randomized sentence order and vectors extracted from sentence orderings in an original document. An example entity transition feature would be the probability that an entity occurs as an object in sentence n and as a subject in sentence $n + 1$. These probabilities are calculated by aggregating over all transitions of a certain length in the document.

While the entity-grid is a commonly used representation for sentence ordering, applications to sentence-level generation tasks such as surface realization are less common and less successful. Cheung and Penn (2010) report on a constituent ordering experiment for German where they incorporate features from the entity grid into the sentence-internal constituent ordering model of Filippova and Strube (2007b). In contrast to the document-level entity transition probabilities used for sentence-ordering, they now use plain entity-level transitions to predict the position of that entity in a sentence. An example for such a feature would be the information that the entity occurred in the Vorfeld two sentences ago, and had no occurrence in the preceding sentence. In their model, the sentence-external features only give a small, non-significant improvement over the sentence-internal model.

The discourse patterns that are captured by the entity grid representation can be seen as informal implementations of basic intuitions modeled by Centering Theory (Grosz et al., 1995). Its most important notions are related to the realization of discourse referents (i.e. described as "centers") and the way the centers are arranged in a sequence of utterances to make this sequence a coherent discourse. Another important concept is the "ranking" of discourse

	victim:0	victim:1	perp	police
1	S	-	-	-
2	-	-	O	S
3	O	-	S	-
4	-	O	S	S
5	-	S	-	-
6	-	O	S	-

(1) [Junge Familie]$_{victim:0}$ auf dem Heimweg ausgeraubt
[Young family]$_{victim:0}$ on the way home robbed

(2) Die Polizei sucht nach [zwei ungepflegt wirkenden jungen Männern im
The police looks for [two shabby-looking young men of about
Alter von etwa 25 Jahren]$_{perp}$.
25 years].

(3) [Sie]$_{perp:0}$ sollen am Montag gegen 20 Uhr [eine junge Familie mit
[They] are said to on Monday around 20 o'clock [a young family with
ihrem sieben Monate alten Baby]$_{victim:0}$ auf dem Heimweg von einem
their seven month old baby] on the way home from
Einkaufsbummel überfallen und ausgeraubt haben.
a shopping tour attacked and robbed have.

(4) Wie die Polizei berichtet, drohten [die zwei Männer]$_{perp}$ [dem Ehemann
As the police reports, threatened [the two men]$_{perp}$ [the husband
]$_{victim:1}$, [ihn]$_{victim:1}$ zusammenzuschlagen.
]$_{victim:1}$ [him]$_{victim:1}$ beat up.

(5) [Er]$_{victim:1}$ gab deshalb [seine]$_{victim:1}$ Brieftasche ohne Gegenwehr heraus.
[He] gave therefore [his] wallet without resistance out.

(6) Anschließend nahmen [ihm]$_{victim:1}$ [die Räuber]$_{perp}$ noch die Armbanduhr ab
Afterwards took [him] [the robbers] also the watch off
und flüchteten.
and fleed.

Figure 4.1: Entity transitions in a grid representation for an example text from the robbery corpus

referents which basically determines the prominence of a referent in a certain sentence and is driven by several factors (e.g. t

Karamanis et al. (2009) use Centering-based metrics to assess coherence in an information ordering system. The main difference to entity grid models is that they do not simply aggregate over all entity transitions in a document, but categorize certain transitions based on notions coming from Centering Theory. A major challenge for computational implementations of the theory is that many of its fundamental categories are not formally defined so that different implementations can lead to different predictions and empirical outcomes (Poesio et al., 2004).

A clean implementation of entity grid representations or a Centering model requires the presence of coreference information. In Section 4.3, we look at a generation task where coreference chains are not available. For this experiment, we will exploit the fact that, besides coreferential relations between occurrences of referents, constituents are related on the lexical level. Morris and Hirst (1991) have proposed that chains of (related) lexical items in a text are an important indicator of text structure. Clarke and Lapata (2010) have improved a sentence compression system by capturing prominence of phrases or referents in terms of lexical chain information inspired by Morris and Hirst (1991) and Grosz et al. (1995). In their system, discourse context is represented in terms of hard constraints modeling whether a certain constituent can be deleted or not.

In Figure 4.2, we illustrate some examples for lexical chains from the data set that we use for surface realization in Section 4.3. Examples (7) and (8) both exemplify the reiteration of a lexical item in two subsequent sentences that also reflects a coreference relation between the constituents. In Example (7), the second instance of the noun 'group' is modified by a demonstrative pronoun such that its "known" and prominent discourse status is overt in the morpho-syntactic realization. In Example (8), both instances of "Belgium" are realized as bare proper nouns without an overt morphosyntactic clue indicating their discourse status. Thus, in Example (8), it can be expected that the chain information is beneficial for the modeling of context.

In Example (9), we illustrate a further type of lexical reiteration where two identical head nouns are realized in subsequent sentences, even though they refer to two different discourse referents. While this type of lexical chain is described as "reiteration without identity of referents" by Morris and Hirst (1991), it would not be captured in Centering since this is not a case of strict coreference. A context model that is based on lexical chains would treat this

(7) a. Kurze Zeit später erklärte ein Anrufer bei Nachrichtenagenturen in Pakistan , **die Gruppe Gamaa** bekenne sich.
Shortly after, a caller declared at the news agencies in Pakistan, that **the group Gamaa** *avowes itself.*
b. **Diese Gruppe** wird für einen Großteil der Gewalttaten verantwortlich gemacht , die seit dreieinhalb Jahren in Ägypten verübt worden sind .
This group *is made responsible for most of the violent acts that have been committed in Egypt in the last three and a half years.*

(8) a. **Belgien** wünscht, dass sich WEU und NATO darüber einigen.
Belgium *wants that WEU and NATO agree on that.*
b. **Belgien** sieht in der NATO die beste militärische Struktur in Europa .
Belgium *sees the best military structure of Europe in the NATO.*

(9) a. **Frauen** vom Land kämpften aktiv darum , ein Staudammprojekt zu verhindern.
Women *from the countryside fighted actively to block the dam project.*
b. Auch in den Städten fänden sich immer mehr **Frauen** in Selbsthilfeorganisationen zusammen.
Also in the cities, more and more **women** *team up in self-help organizations.*

Figure 4.2: Examples for lexical chains as entity transitions

type of reiteration in the same way as the types illustrated in the previous examples.

4.2 Experiment 1: Constituent Ordering with Entity Transitions

We first look at a simplified generation setup where we concentrate on the ordering of constituents in the German *Vorfeld* and *Mittelfeld*. This strategy has also been adopted in previous investigations of German word order: Filippova and Strube (2007b) show that once the German *Vorfeld* is correctly chosen, the prediction accuracy for the *Mittelfeld* (the constituents following the finite verb) is in the 90s.

We extract sentence-external features from perfect coreference information available in the input and compare them to sentence-internal morpho-syntactic features. The aim is to establish an upper bound concerning the quality improvement for word order prediction by recurring to manual coreference annotation. The input to the generator is a set of constituents extracted from the syntactic representation of a corpus sentence. Essentially,

our experiments can be seen as a replication of Cheung and Penn (2010), who also incorporate features extracted from an entity grid into a constituent ordering model.

4.2.1 Data and Setup

We carry out the constituent ordering experiment on the Tüba-D/Z treebank (v5) of German newspaper articles (Telljohann et al., 2006). It comprises about 800k tokens in 45k sentences. We choose this corpus because it is not only annotated with syntactic analyses but also with coreference relations (Naumann, 2006). The syntactic annotation format is very convenient as it explicitly represents the *Vorfeld* and *Mittelfeld* as phrasal nodes in the tree.

The Tüba-D/Z coreference annotation distinguishes several relations between discourse referents, most importantly "coreferential relation" and "anaphoric relation" where the first denotes a relation between noun phrases that refer to the same entity, and the latter refers to a link between a pronoun and a contextual antecedent, see Naumann (2006) for further detail. We expected the coreferential relation to be particularly useful, as it cannot always be read off the morpho-syntactic realization of a noun phrase, whereas pronouns are almost always used in an anaphoric relation.

The constituent ordering model is implemented as a classifier that is given a set of constituents and predicts the constituent that is most likely to be realized in the *Vorfeld*.

The set of candidate constituents is determined from the tree of the original corpus sentence. We will assume that all constituents under a *Vorfeld* and *Mittelfeld* node can be freely reordered. Thus, we do not check whether the word order variants we look at are actually grammatical assuming that most of them are. In this sense, this experiment is close to fully statistical generation approaches. As a further simplification, we do not look at morphological generation variants of the constituents or their head verb.

The classifier is implemented with SVMrank (Joachims, 2006). In contrast to the following experiment in Section 4.3 where we learned to rank sentences, the classifier learns to rank constituents. The constituents have been extracted using the tool described in Bouma (2010). The final data set comprises 48.513 candidate sets of freely orderable constituents.

ConstituentLength + HeadPos	length of the constituent, the Part-of-speech of its head
BaseSyn	syntactic function (arguments and adjuncts) syntactic category (e.g. adverbs, proper nouns)
FullMorphSyn	modification (e.g. relative clauses, genitives in NPs) definiteness for nouns number and person for nominal elements constituent span and number of embedded nodes in the tree types of pronouns (e.g. demonstrative, reflexive)

Table 4.1: Experiment 1: sentence-internal feature classes for constituent ordering

4.2.2 Sentence-Internal Baseline Model

To compare the discourse context model against a sentence-based model, we implemented a number of sentence-internal features classes, summarized in Table 4.1. As a simple baseline, we assume that information about constituent length and part-of-speech is specified. "BaseSyn" adds some basic features including the labels of the constituent, "FullMorphSyn" records detailed morpho-syntactic properties.

4.2.3 Centering-inspired Features

For the sentence-external features, we extract coreference relations of several types (see Naumann (2006) for the anaphoric relations annotated in the Tüba-D/Z). For each type of coreference link, we extract the following properties: (i) function of the antecedent, (ii) position of the antecedent, (iii) distance between sentences, (iv) type of relation. We also distinguish coreference links annotated for the whole phrase ("head link") and links that are annotated for an element embedded by the constituent ("contained link"). The two types are illustrated in Examples (10) and (11). Note that both cases would not have been captured in the lexical chain model since there is no lexical overlap between the realizations of the discourse referents.

(10) a. Die Rechnung geht an **die AWO**.
The bill goes to **the AWO**.

b. [Hintergrund der gegenseitigen Vorwürfe in **der Arbeiterwohlfahrt**] sind offenbar scharfe Konkurrenzen zwischen Bremern und Bremerhavenern.
Apparently, [the background of the mutual accusations at **the labour welfare***] are rivalries between people from Bremen and Bremerhaven.*

(11) a. Dies ist die Behauptung, mit der **Bremens Häfensenator** die Skeptiker davon überzeugt hat, [...].
This is the claim, which **Bremen's harbour senator** *used to convince doubters, [...].*

b. Für diese Behauptung hat **Beckmeyer** bisher keinen Nachweis geliefert. *So far,* **Beckmeyer** *has not given a prove of this claim.*

These types of coreference features implicitly carry the information that would also be considered in a Centering formalization of discourse context. In addition to these, we designed features that explicitly describe centers as these might have a higher weight. In line with Clarke and Lapata (2010), we compute backward (CB) and forward centers (CF) in the following way:

1. Extract all entities from the current sentence and the previous sentence.

2. Rank the entities of the previous sentence according to their function (subject $<$ direct object $<$ indirect object ...).

3. Find the highest ranked entity in the previous sentence that has a link to an entity in the current sentence, this entity is the CB of the sentence.

In the same way, we mark entities as forward centers that are ranked highest in the current sentence and have a link to an entity in the following sentence.[1] In Table 4.2, we report the percentage of sentences that have backward and forward centers in the *Vorfeld* or *Mittelfeld*. While the percentage of sentences that realize a backward center is quite low, the overall proportion of sentences containing some type of coreference link is in a dimension

[1] In Centering, all entities in a given utterance can be seen as forward centers, however we thought that this implementation would be more useful.

	# VF	# MF
Backward Center	3.5%	5.1%
Forward Center	6.8%	6.8%
Coref Link	30.5%	23.4%

Table 4.2: Experiment 1: distribution of backward and forward centers and their positions in the Tüba-D/Z data

such that the learner could definitely pick up some predictive patterns. Going by the relative frequencies, coreferential constituents have a bias towards appearing in the *Vorfeld* rather than in the *Mittelfeld*.

4.2.4 Results

First, we build three coreference-based constituent classifiers on their entire training set and compare them to their sentence-internal baseline. The most simple baseline records the category of the constituent head and the number of words that the constituent spans. Additionally, we build a "BaseSyn" model which has the syntactic function features, and a "FullMorphSyn" model which comprises the entire set of sentence-internal features. To each of these baseline, we add the coreference features.

The results are reported in Table 4.3, showing an effect of the sentence-external features over the simple sentence-internal baselines. However, in the "FullMorphSyn" sentence-internal model, the effect is minimal. This suggests that the morpho-syntactic features are highly correlated with sentence-external factors. Moreover, for each baseline, we obtain higher improvements by adding further sentence-internal features than by adding sentence-external ones: the accuracy of the simple baseline (47.48%) improves by 7.34 points when function features (the accuracy of BaseSyn is 54.82%) are added and by only 3.48 points through adding coreference features. The performance of the basic syntactic model with coreference features is below the "FullMorphSyn" model which means that the morpho-syntactic features provide more informative contextual cues than the sentence-external features.

However, the statistics in Table 4.2 have shown that there is substantial set of sentences in our data that do not have a coreference link for one of their constituents. Therefore, we run a second experiment in order to see how coreference-based features behave on the subset of sentences where they

Model	VF
ConstituentLength + HeadPos	47.48%
ConstituentLength + HeadPos + Coref	51.30%
BaseSyn	54.82%
BaseSyn + Coref	56.21%
FullMorphSyn	57.24%
FullMorphSyn + Coref	57.40%

Table 4.3: Results for Experiment 1 reported for sentence-internal feature classes combined with coref features: accuracy for correct predictions of the Vorfeld, training and evaluation on entire treebank

can actually be detected. We build and evaluate the same set of classifiers on the subset of sentences that contain at least one coreference link for one of its constituents.

The results for the second evaluation are given in Table 4.4. This time, the coreference features improve over all sentence-internal baselines including the 'FullMorphSyn' model. This finding extends the results from Cheung and Penn (2010) who could not find a significant effect of sentence-external features.

However, it is interesting to note that in the second evaluation the morpho-syntactic features seem to behave differently than in the model trained on the full set of sentences. Thus, the performance of "FullMorphSyn" trained on the subset is 2 points below the performance reported in the first evaluation in Table 4.3. In contrast, "BaseSyn" achieves basically the same performance in both cases. This effect could be related to the fact that the models from Table 4.4 are trained on less data. It is possible that the more fine-grained set of morpho-syntactic features needs to be trained on a larger set of sentences, whereas the "BaseSyn" model is less sensitive.

Thus, although the "FullMorphSyn + Coref" model outperforms "FullMorphSyn" on the coreference subset, it improves only marginally over the top scores from the evaluation in Table 4.3. This could mean that the coreference features simply compensate for the lower performance of the morpho-syntactic features on the smaller data set, instead of really adding to the predictive power of the sentence-internal features. Another possible interpretation is that the sentences containing coreferent constituents are harder to linearize for some reason (maybe they are longer, or contain more compet-

Model	VF
ConstituentLength + HeadPos	46.61%
ConstituentLength + HeadPos + Coref	52.23%
BaseSyn	54.63%
BaseSyn + Coref	56.67%
FullMorphSyn	55.36%
FullMorphSyn + Coref	57.93%

Table 4.4: Results for Experiment 1 reported for sentence-internal feature classes combined with coref features: accuracy for correct predictions of the Vorfeld, training and evaluation on sentences that contain a coreference link

ing constituents, etc.). In any case, the models do not surpass a certain upper bound at around 58% prediction accuracy for the German *Vorfeld*. Interestingly, we find a similar compensation or upper bound effect for sentence-external factors in the REG experiment described in Section 4.4.

4.3 Experiment 2: Realization Ranking with Lexical Overlaps

In this Section, we present an experiment that investigates sentence-external context in a surface realization task where candidate selection is applied to entire sentences. The sentence-external context is represented in terms of lexical overlap features and compared to sentence-internal models which are based on morphosyntactic features. The experiment thus targets a generation scenario where no coreference information is available and aims at assessing whether relatively naive, but broad-coverage context information is useful.

4.3.1 Data and Set-up

The experiment is carried out in the LFG-based regeneration framework described in Chapter 2.2. We first use the XLE generator to produce surface realization candidates from LFG F-structures. The F-structures have been obtained from parsing the TIGER Treebank of German newspaper text (Brants et al., 2002) with the German broad-coverage LFG grammar. The actual surface realization, the choice of an appropriate output sentence from

# Sentences in context	% Sentences with overlap Training	Dev	Test
1	20.96	23.64	20.42
2	35.42	40.74	35.00
3	45.58	50.00	53.33
4	52.66	53.70	58.75
5	57.45	58.18	64.58
6	61.42	57.41	68.75
7	64.58	61.11	70.83
8	67.05	62.96	72.08
9	69.20	64.81	74.17
10	71.16	70.37	75.83

Table 4.5: Experiment 2: proportion of sentences that have at least one overlapping entity in the previous n sentences

the set of candidates, is implemented as an SVM-based ranking component.

The models are trained on 7,039 sentences from the TIGER corpus. The sentences in this set fulfill two conditions: a) the German LFG produces an analysis that is compatible with the their gold-standard annotation in the treebank, b) the XLE generator successfully generates such that the original corpus sentence is among the candidates, see Chapter 2.2 for more details. The set of 7,039 sentences comes from 1259 texts in the corpus. The TIGER treebank has no coreference annotation.

4.3.2 Sentence-Internal Baseline Models

The aim of this experiment is to understand the nature of sentence-internal features reflecting discourse context and compare them to sentence-external ones in a corpus-based generation setting that is more realistic as compared to the work in Section 4.2. Again, for our sentence-internal model, we build several baselines that capture the morpho-syntactic realization of the constituents in a given sentence with varying degrees of syntactic detail. The simplest model just includes the language model score and the sentence length for each sentence. The"BaseSyn" model includes features that capture the syntactic functions of the constituents in the sentence which can be extracted from the underlying input F-structure. The "FullMorphSyn" model includes

Language Model	language model score sentence length
BaseSyn	syntactic function (arguments and adjuncts) syntactic category (e.g. adverbs, proper nouns)
FullMorphSyn	modification (e.g. relative clauses, genitives in NPs) definiteness for nouns number and person for nominal elements constituent span and number of embedded nodes in the tree types of pronouns (e.g. demonstrative, reflexive)

Table 4.6: Experiment 2: sentence-internal feature classes for surface realization with F-structures

more features from the F-structure, such as definiteness or number for nouns. On overview of the sentence-internal features is given in Table 4.6.

In this experiment, we do not directly predict the position of a particular constituent in a sentence. Instead, we have to discriminate between sentences that display different word orders. Therefore, we combine the syntactic features for each constituent (Table 4.6) into complex features that describe the precedence relation between two constituents. For instance, in the BaseSyn model, we combine the syntactic function of two constituents into the complex features "subj << obj" for all candidates where the subject precedes the object, and "obj << subj" for all candidates that have the opposite order. The combinations are extracted for each pair of constituents. In the FullMorphSyn model, a constituent can be described by a complex feature such as "subj + sg + indef" for an NP that is indefinite, singular and the subject of the sentence. The resulting precedence features reflect complex patterns, such as "obj + pl + def << subj + sg + indef" for a candidate where a plural definite object precedes a singular indefinite subject.

4.3.3 Sentence-External Overlap Features

As coreference information is not available in the current setting, we represent sentence-external context information in terms of lexical overlaps, reiteration of lexical items in a text. For extracting the lexical overlaps, we check for any nouns in the n previous sentences whether they match a noun in the current

sentence being generated. We check proper and common nouns, considering full and partial overlaps as shown in Examples (7) and (8), where the (a) example is the previous sentence in the corpus. Table 4.5 shows how many sentences in our training, development and test sets have at least one textually overlapping phrase in the previous 1–10 sentences. The extraction of the sentence-external features will be parametrized by the size of the context window S_c.

Beyond the simple presence of reiterated items in sequences of sentences, we expected that it would be useful to look at the position and syntactic function of the previous mentions of a discourse referent. In Example (7), the reiterated item is first introduced in an embedded sentence and realized in the *Vorfeld* in the second utterance. In terms of centering, this transition would correspond to a topic shift. In Example (8), both instances are realized in the *Vorfeld*, such that the topic of the first sentence is carried over to the next. Thus, for each overlap, we record the following properties:

- function in the previous sentence
- position in the previous sentence (e.g. *Vorfeld*)
- distance between sentences
- total number of overlaps

These overlap features are then also combined in terms of precedence, e.g. "subject_overlap:3 << no_overlap", meaning that in the current sentence a noun that was previously mentioned in a subject 3 sentences ago precedes a noun that was not mentioned before.

4.3.4 Results

For comparing the string chosen by the models against the original corpus sentence, we use BLEU, NIST and exact match. Exact match is a strict measure that only credits the system if it chooses the exact same string as the original corpus string. BLEU and NIST are more relaxed measures that compare the strings on the n-gram level. Finally, we report accuracy scores for the *Vorfeld* position (VF) corresponding to the percentage of sentences generated with a correct *Vorfeld*.

S_c	BLEU	NIST	Exact	VF
0	0.766	11.885	50.19	64.0
1	0.765	11.756	49.78	64.0
2	0.765	11.886	50.01	64.1
3	0.765	11.885	50.08	63.8
4	0.761	11.723	49.43	63.2
5	0.765	11.884	49.71	64.2
6	0.768	11.892	50.42	64.6
7	0.765	11.885	50.01	64.5
8	0.764	11.884	49.78	64.3
9	0.765	11.888	49.82	63.6
10	0.764	11.889	49.7	63.5

Table 4.7: Results for Experiment 2 reported for different context windows (S_c): all sentence-internal features (FullMorphSyn) combined with sentence-external overlap features, tenfold-crossvali-dation

In Table 4.7, we report the performance of the full sentence-internal feature model, FullMorphSyn, combined with context windows from zero to ten. The scores have been obtained from tenfold-crossvalidation. For none of the context windows, the model outperforms the baseline with a zero context which has no sentence-external features.

In Table 4.8, we compare the performance of several sentence-internal baselines against their corresponding sentence-external models. We note that the function precedence features (i.e. the 'BaseSyn' model) are very powerful, leading to a major improvement compared to a language model. The sentence-external features lead to an improvement when combined with the language-model based ranking. However, this improvement is leveled out in the BaseSyn model.

On the one hand, the fact that the lexical chain features improve a language-model based ranking suggests these features are, to some extent, predictive for certain patterns of German word order. On the other hand, the fact that they don't improve over an informed sentence-internal baseline suggests that these patterns are equally well captured by morphosyntactic features. However, we cannot exclude the possibility that the chain features are too noisy as they conflate several types of lexical and coreferential relations. This will be addressed in the following experiment.

Model	BLEU	VF
Language Model	0.702	51.2
Language Model + Context $S_c = 5$	0.715	54.3
BaseSyn	0.757	62.0
BaseSyn + Context $S_c = 5$	0.760	63.0
FullMorphSyn	0.766	64.0
FullMorphSyn + Context $S_c = 5$	0.763	64.2

Table 4.8: Results for Experiment 2 reported for subclasses of sentence-internal combined with sentence-external features; 'Language Model': ranking based on language model scores, 'BaseSyn': precedence between constituent functions, 'FullMorphSyn': entire set of sentence-internal features.

4.4 Experiment 3: Context in Referring Expression Generation

In this Section, we now turn to experiments where we model choices that concern the morpho-syntactic realizations of referring expressions. We basically use the set-up established in the GREC task (see Chapter 2.4) where candidate referring expressions from a list have to be assigned to occurrence slots in a corpus text. Thus, the task involves the generation of first mentions as well as subsequent references to entities in a text.

The experiments about sentence-external features for word order prediction in Section 4.2 and 4.3 have shown that the effectiveness of these features depends to a large extent on the quality of the sentence-internal model, and additionally on the availability and coverage of high-quality sentence-external context information. For our experiment on REG, we will also look at context information at different levels of quality, comparing a scenario with almost perfect syntactic information about the discourse context to a scenario where this information is predicted by a generation component. Another factor that we investigate is the modeling of the context information in terms of hard or soft constraints. We compare a rule-based algorithmic baseline in the spirit of Henschel et al. (2000) against a soft ranking system in the spirit of McCoy and Strube (1999). Additionally, we experiment with hard constraints that extend the feature model of our REG module.

Intuitively, one might expect that sentence-external context can be operationalized more easily for the generation of referring expressions than for

word order prediction. For instance, indefinite referring expressions can only be used to introduce a discourse-new entity into the text whereas definite descriptions and pronouns are reserved for discourse-old referents. Whether a referent or its particular instance in the text is discourse-new or discourse-old can be read off the reference chain which is given in the generation input. However, somewhat surprisingly, the state-of-the-art in REG is comparable to word order prediction when it comes to representing and modeling sentence-external context. Similarly to word order prediction, sentence-internal features that simply exploit surface patterns of the immediate local context seem to be quite effective and are not always outperformed by more informed, theoretically motivated models (see e.g. Greenbacker and McCoy (2009)).

4.4.1 Data and Set-up

We use the data set of German robbery articles introduced in Chapter 2.4.2 (more details are given in Chapter 5.4). The articles are annotated for mentions of referents, represented as slots in an ordered, syntactic dependency tree. We leave out the annotation of implicit referents and the deep dependency layer such that the REG module has information to the surrounding surface syntactic context.

We compare the performance of the REG module in two scenarios: In the first scenario, the syntactic trees correspond to original annotations of the corpus text. This setting corresponds closely to the set-up of the GREC Shared Task (Belz and Kow, 2010) where the REG component has access to perfect context information. In the second scenario, the trees have been predicted and linearized by the generation components described in Chapter 7.2 such that the syntactic context and information about surface order is not perfect. For experiments that describe and evaluate these syntactic components, see Chapter 6.3 and 7.2.

The input to the REG module in this task is a sequence of sentences represented as trees with RE slots and a list of candidate REs for each referent involved in the robbery event (a victim, a perpetrator, possibly a source). For each referent, the candidates are specified in terms of dependency subtrees that have to be inserted in the appropriate slot in the sentence trees. The candidates contain all realizations of the referent in the original text, plus a pronominal realization and a default definite description for each referent.

We split the 200 articles in the data set into 10 splits of 20 articles. We

train the REG module on 9 splits and report scores on the development split.

4.4.2 Baseline Algorithm

For comparison with our trained REG module, we design a baseline algorithm that assigns REs to slots in the trees based on some simple heuristics. The intuition of the algorithm is that the the first mentions of a referent in a text tend be rather long descriptions of the person, whereas following mentions can be shorter or pronominal. We assign pronouns to all subsequent mentions of a referent, when there is no intervening mention of another referent. We generate the default nominal RE when all other nominal REs from the candidate list have been distributed over the text. The baseline for the REG component is defined as follows:

1. if sentence is a header:
   (a) generate the longest unused nominal RE without an article
2. else:
   (a) if the last generated RE mentions the same referent
      i. generate a pronoun
   (b) else:
      i. if an unused nominal RE is available:
         A. generate the longest unused nominal RE
      ii. else:
         A. generate the default nominal RE

4.4.3 Feature models

The REG module is implemented as an SVM-based ranker. During training, each slot of a referent is paired with all its candidate REs. The correct candidate for a certain slot is labeled with the best rank, all other candidate get the lower rank. During testing, the ranker predicts a rank for each candidate, and we select the candidate with the best rank as the predicted output.

The features for the ranker combine properties of the slot with properties of the RE candidate. We define the following feature classes:

- local slot: local features of the slot where the candidate has to be inserted, extracted from the tree of the sentence
- RE prop: properties of the RE candidate, extracted from the candidate's subtree
- current sent ref: features of other generated REs in the same sentence
- prev text ref: features of other generated REs in the previous text
- global ent: global features of the referent.

The features for each of the feature classes are illustrated in Table 4.9. The simplest feature model combines the properties of the slot with properties of the RE candidate. This model has only access to the local syntactic context and does not have any knowledge about the previously generated REs. The other feature classes extend this local, sentence-internal model with more information about the surrounding context.

4.4.4 Hard Context Constraints

In the standard training procedure of the REG module, the ranker is provided with the full candidate list for all instances of a referent in a text. The purpose of the hard constraints is that this set can be constrained especially for the later mentions of a referent when we know about the previous decisions of the ranker. This can ease the training and the prediction phase since the ranker has to discriminate between a smaller number of candidates. The underlying linguistic intuition is mostly a stylistic constraint: by excluding a nominal description of a referent that has been previously generated for the referent from the candidate list, we avoid that the exact same description is repeated often in the text and make sure that most of the nominal candidates will be assigned to a slot in the text.

1. exclude the pronominal RE
   (a) if the last generated RE is in the same sentence is pronominal and does not mention the same referent
   (b) if the sentence is a header
2. exclude a nominal RE
   (a) if it was previously generated for the same referent

local slot	sentence is a header
	node label
	role of the referent
	position in sentence; distance to mother node
	lemma, PoS, label, relative position of: the mother, grandmother, uncle and sibling nodes
RE prop	RE is a pronoun
	RE is a default description
	lemma, PoS of the RE head node
	lemma, PoS and labels for the RE head's children
	RE has an article
	RE length
current sent ref	n mentions of other referents in sentence
	n mentions of same referent in sentence
	preceding pronominal mentions of same referent in sentence
	preceding pronominal mentions of other referent in sentence
	preceding mention with same head lemma of same referent in sentence
	preceding mention with same head PoS of same referent in sentence
	preceding mention with same head lemma of other referent in sentence
	preceding mention with same head PoS of other referent in sentence
prev text ref	referent is mentioned in preceding text
	other referent is mentioned after last mention in preceding sentence
	other names are mentioned after last mention in preceding sentence
	same RE for referent used in previous sentences
	pronominal RE used for referent + label in previous sentences
	pronominal RE used for referent + label in preceding sentence
global ent	identity of the referent is known
	singular/plural referent
	age of the referent is known
	number of RE candidates for the referent

Table 4.9: Experiment 3: feature classes for REG in specified syntactic trees

	Exact Accuracy	Type Accuracy
Alg. baseline	40.95	54.28
Local slot + local ref	45.08	74.6
+ current sent ref	51.11	79.37
+ prev text ref	55.23	79.37
+ hard context constraints	57.14	79.37
+ global ent	57.46	78.73

Table 4.10: Results for Experiment 3: feature ablation for plain REG on gold syntactic input

4.4.5 Results

We assess the quality of the RE prediction with the following evaluation measures:

1. Exact Accuracy, proportion of REs selected by the system with a string identical to the RE string in the original corpus, as in Belz et al. (2010)

2. Type Accuracy, proportion of REs selected by the system with an RE type identical to the RE type in the original corpus, as in Belz et al. (2010)

In Table 4.10, we report the REG performance of the baseline and our ranking module in the perfect scenario. Table 4.11 shows the results in the predicted syntax scenario. Both the algorithmic baseline and the simplest feature model do not show a big drop in performance in the latter scenario. Generally, the baseline obtains very low scores when compared to the ranking model. Surprisingly, the sentence-internal ranking and the baseline still have comparable performance with respect to the Exact Accuracy, but there is huge drop for the baseline in the Type Accuracy measure. We see this as evidence for the fact that certain RE candidates can be assigned relatively easily to their slots (probably the nominal descriptions in the header and the first sentence), whereas the distinction between pronoun and definite description is not well captured by our rule-based heuristics.

When we compare the performance of the ranking model between the two scenarios, we find that, although the sentence-internal feature models obtain similar scores, the sentence-external feature classes behave differently. In the perfect scenario, the generation context from the current sentence leads to a

	Exact Accuracy	Type Accuracy
Alg. baseline	39.68	53.33
Local slot + local ref	44.76	73.65
+ current sent ref	45.71	75.55
+ prev text ref	50.16	75.24
+ hard context constraints	55.24	75.87
+ global ent	57.14	77.14

Table 4.11: Results for Experiment 3: feature ablation for plain REG on predicted syntactic input

substantial improvement in both measures. This improvement is smaller in the predicted syntax scenario. The sentence-external features from the previous text and the hard constraints further improve the Exact Accuracy in the perfect scenario whereas the optimal Type Accuracy is already reached by integrating the generation context from the current sentence. In the predicted scenario, the features from the previous text and the hard constraints lead to a big improvement for the Exact Accuracy, and the global referent features also improve the Type Accuracy. Thus, finally, the ranker achieves a comparable performance in both scenarios, however, the contribution of the different context feature classes are clearly different.

We interpret these differences as follows: in the perfect scenario, the model has access to perfect precedence and surface order information for the current sentence. This leads to the optimal prediction for pronouns vs. definite descriptions that the ranking model can reach. In the predicted syntax scenario, this surface information is of lower quality and the model makes more mistakes at this level. However, the sentence-external features can compensate for this misleading sentence-internal context and are more effective in this case. This effect is similar to the results on constituent ordering reported in Section 4.2, where the sentence-external features did not lead to an overall improvement, but could compensate for less accurate morpho-syntactic features.

4.5 Discussion

First of all, the experiments presented in this Chapter have replicated a number of effects found in previous work: Similar to Cheung and Penn (2010), we

find that a rich sentence-internal model for word order prediction that captures fine-grained morpho-syntactic properties of the involved constituents can be hardly improved by incorporating features that represent sentence-external relations of these constituents. Moreover, in line with Greenbacker and McCoy (2009), we have shown that an REG component applied on perfect syntactic trees achieves the optimal performance for predicting the distinction between pronouns and nominal descriptions when it is trained with a sentence-internal feature model.

This suggests that sentence-internal context cues and accurate morpho-syntactic properties provide a fairly good way of approximating local coherence in a discourse context. The fact that the sentence-external features we implemented improve over simple baselines, but get leveled out in rich sentence-internal feature models further corroborates that there are tight interactions and interdependencies between these factors, which explains the fairly high baseline performance of n-gram language models in the surface realization task, and the accuracy state-of-the-art linearizers and generators obtain by exploiting sentence-internal properties of the involved constituents.

But beyond confirming state-of-the-art results, the experiments also put the seemingly clear picture into a different perspective: Sentence-external features do play a role in generation scenarios where rich sentence-internal models cannot be effectively computed due to less training data (as in Section 4.2), or due to lower-quality input (as in Section 4.4). When looking at contextual factors for syntactic and referential choice side by side, it is striking that models of syntactic choice presuppose perfect knowledge about surface forms of referring expressions whereas referential choice models assume perfect syntactic knowledge to be available. In a system that generates from less artificial sources, these underlying assumptions mutually exclude each other.

Moreover, many corpus-based approaches discussed in Chapter 3 assumed perfect sentence-internal information to be available from manual annotation, as e.g. Bresnan and Ford (2010), or gold standard syntactic analyses, as e.g. Cahill and Riester (2009). However, in this thesis, we are ultimately interested in generating from abstract inputs where choice is less restricted. In this setting, it should be expected that context models have to deal with less perfect information sources as sentence-internal and sentence-external context changes according to the decisions of generation modules, and according to the actual input. From this perspective, a contextual factor is not an invariable property that has constant predictive ability, but it is likely to

change with a number of parameters defined by the particular state of the generation system. Finally, there is not even a clear boundary between a contextual factor that is useful for predicting a certain choice, or a choice that has to be modeled in interaction with other competing choices.

From a broader generation perspective, the results obtained from comparing contextual factors in perfect and less perfect scenarios can be interpreted as a finding about *choice*: Since referential and syntactic choice interact so closely, they have strong predictive power if they are treated as contextual cues available from some pre-specified input. Thus, the challenge that has to addressed in an obvious next step is to investigate context in generation scenarios that provide more abstract sources where several choices have to modelled simultaneously. This idea will be pursued in the remaining Chapters of this thesis.

Chapter 5

Reconstructing the Source of Syntactic Choice: Towards Broad-Coverage Alternation Generation

While the input to any NLU system that analyzes sentences will always be strings corresponding to some linguistic sentences, the input to an NLG system can be any type of abstract representation meant to be uttered in a linguistic sentence. A linguistic sentence will always be a sequence of lexical words structured in terms of syntactic phrases, an abstract representation of some content can vary substantially with respect to the amount of linguistic information it specifies: a weather forecast generator applied on numerical input is not provided with any syntactic or lexical information concerning the linguistic output, a generator used in a machine translation system is likely to be based on some syntactic structure over some set of lexical words. Hence, the amount of candidates, the number of choices and, the complexity of the generation problem depends crucially on its input. The surface realization and referring expression generation (REG) approaches discussed in the previous Chapter 4 were located at the extremely controlled and linguistically informed end of the scale: inputs for surface realization pre-specified referential choice and inputs for REG pre-specified syntactic choice such that the underlying candidate selection models had access to a highly informative set of sentence-internal contextual factors. This is illustrated again in Example (1) (which we have already seen in the Introduction):

(1) Input:
`schicken(subject:er, object:Buch, object-ind:sie)`

a. Er hat ihr das Buch geschickt.
He has her the book sent.
b. ?Er hat das Buch ihr geschickt.
c. Ihr hat er das Buch geschickt.
d. ??Ihr hat das Buch er geschickt.
e. Das Buch hat er ihr geschickt.
f.???Das Buch hat ihr er geschickt.

In this Chapter, we set out to develop corpus-based generation settings that involve a more abstract source such that a wider range of choice phenomena can be accounted for. Starting out from surface realization and REG in terms of separate choice processes, we now describe possible extensions that target passives and nominalizations as two important syntactic alternation types. For instance, we will replace the grammatical roles in the generation input in (1) by more abstract semantic roles, illustrated in Example (2) where the sentence can be realized in passive voice demoting the agent referent to a PP. Consequently, some of the constituent orders that are not natural in (1), are perfectly appropriate in (2). Thus, the fact that we remove information about grammatical roles from (1), leads to a wider range of underlying choice phenomena, i.e. word order and voice, and a more complex choice problem, since the set of well-formed and natural candidates is bigger.

(2) Input:
`schicken(agent:er, theme:Buch, recipient:sie)`

a. Von ihm wurde ihr das Buch geschickt.
By him was her the book sent.
b. ?Von ihm wurde das Buch ihr geschickt.
c. ?Ihr wurde von ihm das Buch geschickt.
d. Ihr wurde das Buch von ihm geschickt.
e.???Das Buch wurde von ihm ihr geschickt.
f. Das Buch wurde ihr vom ihm geschickt.

Figure 5.1 depicts the general idea of using the corpus-based setting for obtaining more candidates in generation. We extend the analysis process applied to the corpus sentence such that *Source A* is mapped to the more abstract representation *Source B* which triggers more candidates due to the

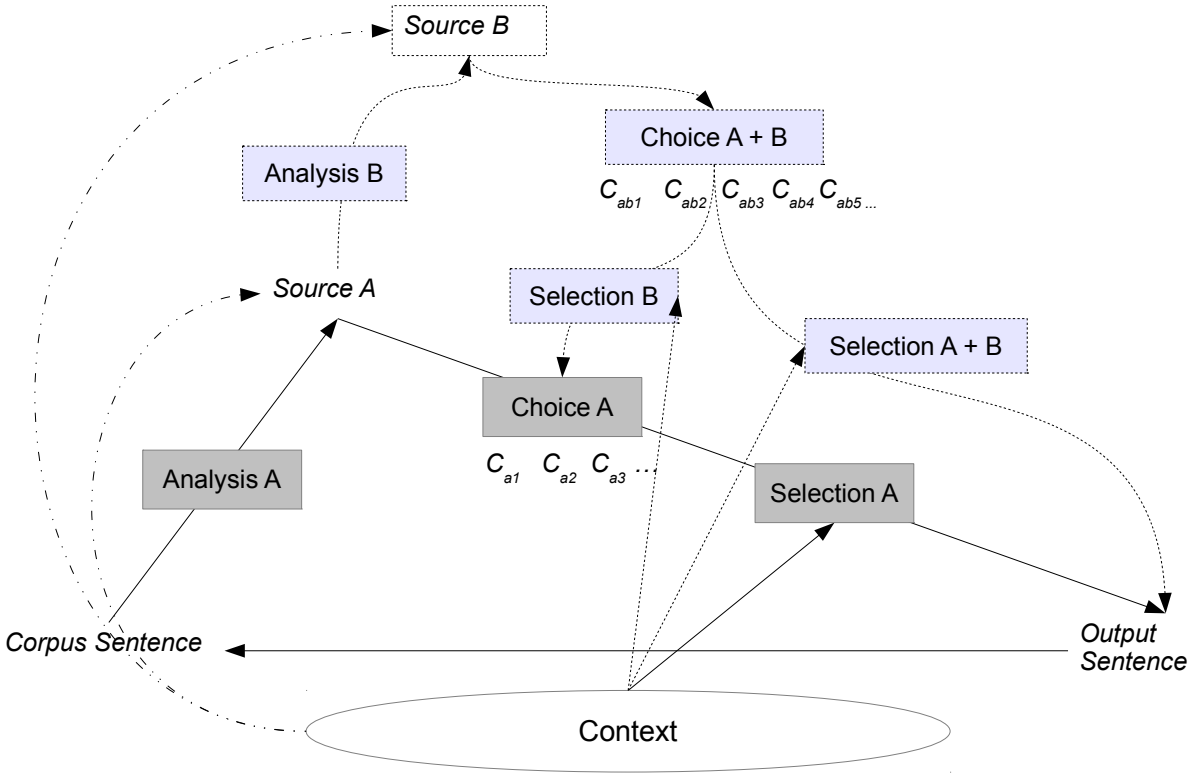


Figure 5.1: The general set-up for corpus-based generation with several levels of choice

combination of several choices. Generally, the analysis step has not received much attention in state-of-the-art surface realization since readily available syntactic representations can be exploited for modeling word order variation.

In this Chapter, we step back from contextual models of candidate selection for a moment and concentrate on deriving appropriate generation inputs that yield clean and well-defined candidate sets for a particular combination of choice phenomena. We observe that the analysis of syntactic alternations for generation is much more involved than word order, due to the underlying morpho-syntactic changes in the predicate argument structure. For instance, these syntactic alternations make the realization of certain semantic arguments optional, which turns out to be a critical issue when deriving corpus-based generation inputs.

Representations and Frameworks Our approach to deriving generation inputs for syntactic alternations is mainly targeted at linguistically appropriate candidate sets. However, issues related to the theoretical and linguistic appropriateness of representations used as generation inputs cannot be discussed independently from the technical properties of the underlying generation framework. As the overview of existing corpus-based generation frameworks in Chapter 2 has shown, there is an important difference between (i) grammar-based generate-and-rank architectures that rely on representation produced by a specific grammar and are often based on rich, linguistically informed syntactic input (e.g. LFG F-structures) and (ii) fully statistical approaches that can be trained on various representations and are often based on more shallow syntactic input (e.g. dependency structures).

The work presented in this Chapter shows that representations for syntactic alternations have important implications for corpus-based generation inputs in grammar-based and statistical frameworks, which have different methodological strengths and weaknesses for modeling this type of syntactic choice: Section 5.1 describes a first extension of the standard LFG-based generation architecture for surface realization, including a simple mapping step for deriving semantic representations from F-structures. We observe that we often do not obtain the desired generation output due to a range of contextual surface cues which block candidate generation. Section 5.2 discusses the phenomenon of implicit arguments and its implications for corpus-based generation. Then, we propose two different ways of dealing with blocking surface cues and implicit arguments. In Section 5.3, we present a carefully

designed set of heuristic rules which produce alternation candidates in the LFG-based surface realization setting. Section 5.4 describes a dependency-based generation approach that integrates syntactic and referential choice in a combined setting.

5.1 A Pilot Approach

In the corpus-based generation setting, input representations are derived from the analysis of sentences in a corpus. In order to provide abstract input that triggers choice in the generation stage, this derivation process needs to assign identical analyses to alternative surface realizations of some content. For generating word order variations, F-structures can be directly used as they are specified in existing broad-coverage LFG grammars, e.g. the German LFG from Rohrer and Forst (2006), as they assign grammatical roles to constituents, independently of their linear order.

In order to be able to generate syntactic alternations like passives and nominalizations, we need to extend the analysis process such that the representations do not encode the morpho-syntactic properties of the predicate in the original corpus sentence. Figure 5.2 first shows two F-structures for an active and a passive sentence. Depending on the voice of the verb, *Carthage* and *Romans* have different grammatical roles. The meaning representation at the bottom of Figure 5.2 abstracts from the syntactic argument realization, assigning the same underlying semantics to the alternation pair.

Semantic representations that remove certain information about the detailed syntactic realization of a sentence are also needed in other domains and NLU applications. For instance, Crouch and King (2006) and Bobrow et al. (2007) developped a question answering system based on semantic representations that are derived from F-strucures, making use of the deep, precise and broad-coverage processing techniques available in the XLE framework. In Zarrieß (2009), we have ported the F-structure rewrite rules from Crouch and King (2006) to the German LFG. Thus, it seems obvious to make use of existing resources for semantic analysis and integrate them in a surface realization architecture. In the following, we will describe a generation setting that extends the LFG-based surface realization framework to take semantic representations in the style of Crouch and King (2006) as input.

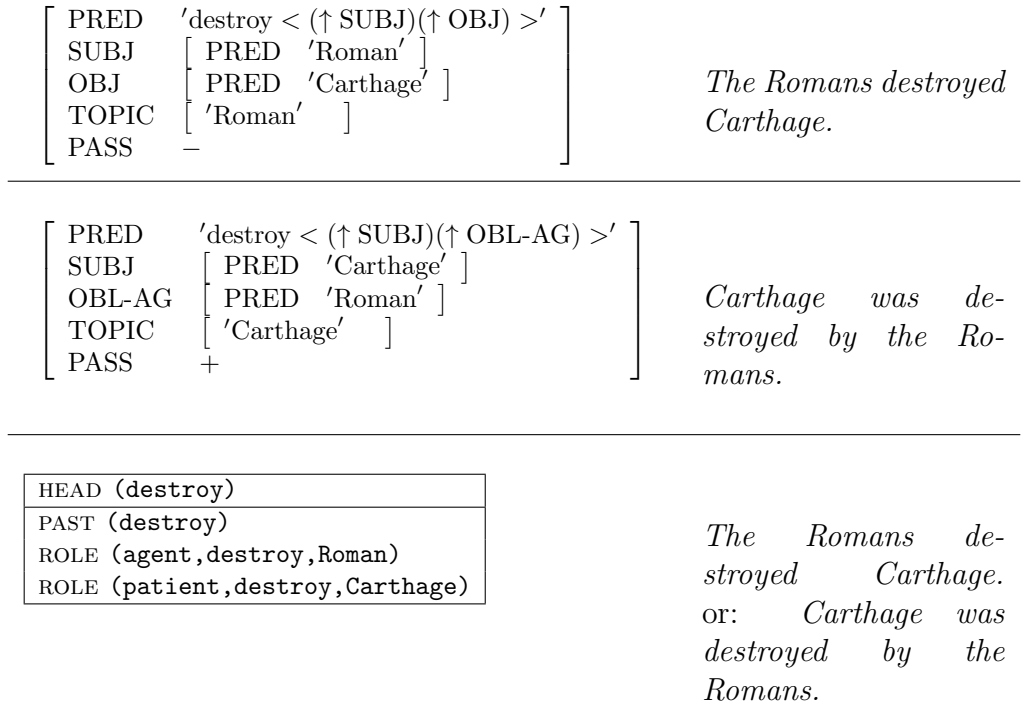


Figure 5.2: A meaning representation that normalizes syntactic alternations, derived from LFG F-structures

5.1.1 Extending Grammar-based Surface Realization

Starting out from the existing mechanisms to derive meaning representations from LFG F-structures, we implemented an extended surface realization architecture in the XLE-based framework described in Section 2.2. The general idea is to extend Cahill et al. (2007a)'s generation pipeline by an intermediate analysis and realization step.

The generation architecture is illustrated in Figure 5.3. First, an input corpus sentence is parsed and mapped to a semantic representation. In the reverse mapping, the generator produces an F-structure chart from a semantic input that, besides the original F-structure, realizes its meaning-equivalent syntactic paraphrases, e.g. voice alternations. This F-structure chart is then mapped to all its corresponding surface sentences by means of the standard XLE generator. Finally, a ranking model selects the most appropriate surface realization.

For mapping LFG F-structures to abstract knowledge representations, Bobrow et al. (2007) use Crouch and King (2006)'s set of semantic rewrite rules that derive flat semantic representations from F-structures as an intermediate analysis step. Instead of a theoretically precise implementation of a syntax-semantic interface that deals with e.g. scope ambiguities, the system targets a robust procedure that can be used in broad-coverage applications. The semantic rules are designed for a) normalization and canonicalization of F-structures, b) integration of external resources such as WordNet.

For our purposes, we focus on the normalization rules implemented by Crouch and King (2006). The main idea of their F-structure normalization is to remove syntax-internal detail from the representation such that syntactic paraphrases which have the same truth-conditional meaning receive an identical analysis. This is a list of core phenomena that the normalization rules address:

(3) a. Attributive vs. predicative modifiers
    (i) Peter reads a good book.
    (ii) Peter reads a book that is good.
  b. Clefts
    (i) It is a book that Peter reads.
    (ii) Peter reads a book.
  c. Genitives
    (i) the building's shadow

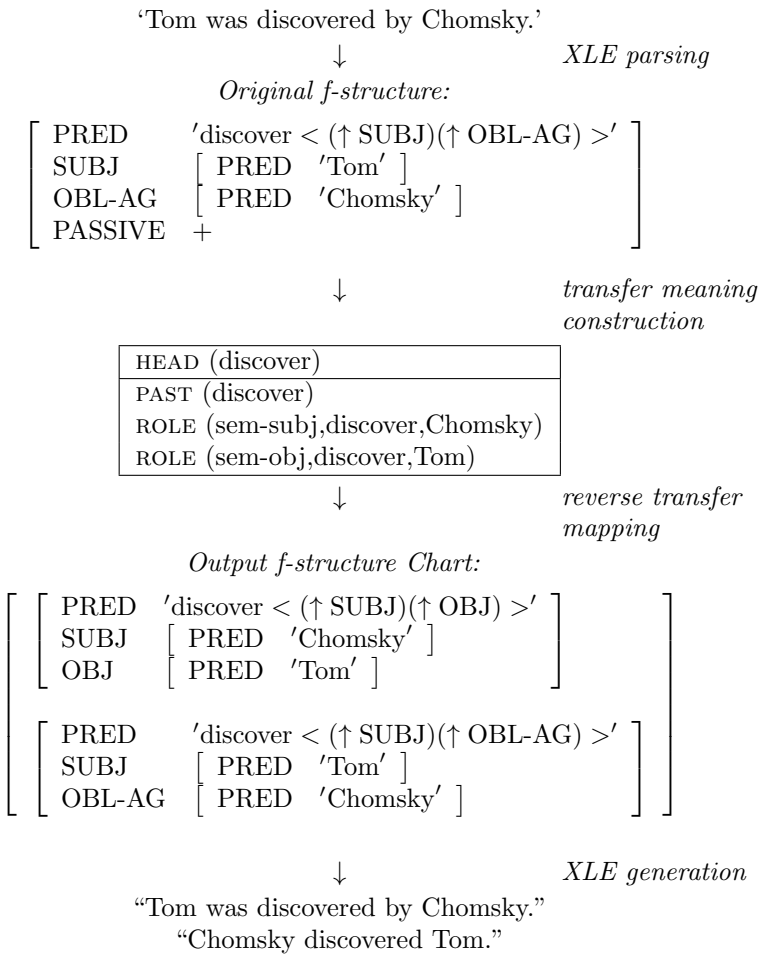


Figure 5.3: Architecture for extended LFG-based surface realization with F-structure generation via meaning representations

(ii) the shadow of the building

d. Nominalizations vs. verbal realizations
   (i) Peter regrets the destruction of the city.
   (ii) Peter regrets that the city was destroyed.

e. Passive vs. active verbal realizations
   (i) The Romans destroyed Carthage.
   (ii) Carthage was destroyed by the Romans.

From the entailment application perspective, the meaning representation in Figure 5.2 is useful as the semantic equivalence of two syntactically different sentences can be derived through simple matching of facts in the representations. From the surface realization, it is useful as abstract input that involves choice between syntactic paraphrases of some underlying content. Thus, it seems reasonable to exploit representations and resources used for semantic analysis for surface realization.

In the intermediate analysis/realization mappings of our extended surface realization pipeline, we have to reconstruct the alternation F-structure from the original analysis representation. For the mapping between active and passive F-structures, we have to rewrite the respective argument frames for transitive verbs. In addition to that, the mapping rules have to deal with a number of atomic and syntax-internal features because they would implicitly disambiguate an abstract semantic representation of an alternation. For instance, if the meaning representation would not underspecify the case of a noun phrase, the surface realizer would have implicit syntactic information about the original sentence realization.

Crouch and King (2006)'s derivation procedure uses the XLE-internal XFR term rewrite system, which has been used in other application contexts where F-structures need to be processed, i.e. F-structure based machine translation (Riezler and Maxwell, 2006), or sentence condensation (Crouch et al., 2004).

In contrast to the core grammar specification in XLE, the XFR component is not reversible. This means that we have develop F-structure rewrite rules for both directions, an analysis and a generation mapping between meaning representations and F-structures. In the mapping from meaning representation to F-structure, nothing guarantees that we actually generate an F-structure that is within the coverage of a given LFG grammar. We rely on the fact that the XLE generator will select from the chart those F-structures that comply with the grammar specification.

1. Analyzing argument roles:

```
+VTYPE(%V,%%),+PASSIVE(%V,+),OBL-AG(%V,%AG)
==>
AGENT(%V,%AG).

+VTYPE(%V,%%),+PASSIVE(%V,+),SUBJ(%V,%PA)
==>
PATIENT(%V,%PA).

+VTYPE(%V,%%),-PASSIVE(%V,+),SUBJ(%V,%AG)
==>
AGENT(%V,%AG).

+VTYPE(%V,%%),-PASSIVE(%V,+),OBJ(%V,%PA)
==>
PATIENT(%V,%PA).
```

2. Removing morphology:

```
+VTYPE(%N,%CHECK),PASSIVE(%V,%%) ==> 0.
CHECK(%N,%CHECK),_VLEX(%CHECK,%V),_AUX-SELECT(%V,%%) ==> 0.
+TNS-ASP(%N,%CHECK),PASS-ASP(%CHECK,%%) ==> 0.
( +AGENT(%N,%C) | +PATIENT(%N,%C) ), CASE(%C,%%) ==> 0.
```

3. Generating argument candidates:

```
AGENT(%V,%AG), PATIENT,(%V,%PA)
?=>
OBL-AG(%V,%AG), SUBJ(%V,%PA).

AGENT(%V,%AG),PATIENT(%V,%PA)
?=>
SUBJ(%V,%AG),OBJ(%V,%PA)
```

Figure 5.4: Example transfer rules used in the pilot approach for normalizing and generating voice alternations from F-structures

The XFR system represents an F-structure internally as a set of two-place terms. The term's name represents the F-structure attribute; the first argument is the F-structure under which the attribute is embedded (where F-structures are referenced by variables var(0), var(1), ..., which have a fixed reference for the full analysis); the second argument is the attribute value, either an atomic value (e.g., CASE(var(1),acc)), or an embedded F-structure node OBJ(var(0),var(1)). The rule syntax for terms to be rewritten vs. conditions is as follows: A prefixed + on left hand rule side turns a term into a (positive) condition, which is not consumed during rule application. Identifiers starting with a % are variables.

Figure 5.4 shows a first transfer grammar we implemented for analyzing and generating voice alternations from LFG F-structures. First, grammatical roles are mapped to general semantic roles that abstract from the morpho-syntactic realization of the predicate. In a second steop, we remove a number of syntax-internal, atomic features that encode e.g. the case of the verb arguments or some tense and aspect features of the predicate. In the final mapping, the abstract semantic roles are optionally mapped to possible argument roles such that F-structure candidates realizing different argument frames for active and passive are produced.

Generally, we do not need to generate full-fledged F-structures from the meaning representations because the XLE generator can handle underspecified input to a certain extent (see Crouch et al. (2004) and Section 2.2.2). Thus, if it does not find an atomic feature that it needs for generation, it tries to generate for all possible instantiations of that feature. By allowing the generator to add atomic features (such as CASE), it can essentially follow the exact grammatical and lexical restrictions on this feature so that we avoid a redundant (and presumably error-prone) duplication of this knowledge in the backward rewrite rules.

5.1.2 Surface Cues Block Candidate Generation

The major goal of the surface realization architecture described above is to extend the candidate sets as compared to candidate generation from F-structures. Due to the completely transparent candidate representation in our generate-and-rank set-up, we can systematically assess whether we capture syntactic choice by an intermediate mapping between semantic representations and F-structures.

When we debugged the extended generator on a testsuite of corpus ex-

amples, we observed a number of cases where the semantic representations did not provide the appropriate level of abstraction. Next to the syntactic features like grammatical roles, verb morphology or noun case, a range of other morpho-syntactic cues can interfere with generation. Example (4-a) illustrates the case where the choice of referring expression restricts the realization of the verb to the active voice.

(4) a. Man hat den Kanzler gesehen.
One has the chancellor seen.
"People have seen the chancellor."

b. *Der Kanzler wurde von man im Park gesehen.
The chancellor was by one in the park seen.

c. *Bad F-structure*:

$$\begin{bmatrix} \text{PRED} & \text{'sehen} < (\uparrow \text{SUBJ})(\uparrow \text{OBL-AG}) > \text{'} \\ \text{SUBJ} & [\ \text{PRED} \quad \text{'Kanzler'}\ ] \\ \text{OBL-AG} & [\ \text{PRED} \quad \text{'man'}\ ] \\ \text{PASS} & + \end{bmatrix}$$

The passive sentence in (4-b) is not ungrammatical because the arbitrary reference pronoun *man* cannot be realized in dative case. Since our relatively shallow procedure for mapping between F-structures and semantic representations is not aware of such morpho-syntactic constraints, it simply maps the subject of the active sentence to the oblique agent in the passive F-structure. The XLE-based generator filters the F-structure candidate as it cannot produce a syntactically well-formed candidate.

While it is a nice feature of a grammar-based generation architecture that ungrammatical sentences get filtered, it is not desirable for a semantic account of verb alternations to be constrained by lexical idiosyncrasies. Thus, there would be natural passive paraphrases with an equivalent meaning, such as a passives with an implicit agent in (5-a) or a different existential pronoun as in (5-b).

(5) a. Der Kanzler wurde im Park gesehen.
The chancellor was in the park seen.

b. Der Kanzler wurde von jemandem im Park gesehen.
The chancellor was by somebody in the park seen.
"The chancellor was seen by somebody in the parc.

Similar interactions exist with other types of pronouns. In Example (6), the neutral demonstrative pronoun *das* (*this*) cannot be used in the *von*

phrase.

(6) a. Das hat den Kanzler verärgert.
This has the chancellor annoyed.
"This annoyed the chancellor."
b. *Der Kanzler wurde von das verärgert.
The chancellor was by this annoyed.
c. Der Kanzler wurde dadurch verärgert.
The chancellor was thereby annoyed.

Furthermore, the realization of the verbal voice interacts with the structural syntactic context of the verb. Figure 5.5 illustrates an interaction of verbal voice with a raising construction. The common F-structure analysis for raising verb is shown at the top, where the raising verb shares the subject F-structure with the embedded verb. The naive transfer grammar does not handle the syntactic context of the embedded verb and produces the F-structure in the middle without changing the subject of the matrix verb. The correct F-structure that can be handled by the XLE generator is given at the bottom.

A similar case occurs when argument phrases are part of a coordination as in Sentence (10-a), where the noun phrase *Marie* is the subject of a coordinated verb phrase where one conjunct is active and the other is passive. The meaning representation in Sentence (10-b) underspecifies the syntactic function for *Marie* in both conjuncts, however it keeps the information about the lexical identity in terms of the unique index for the referent. If the generator "knows" that the two subjects have to be realized by the same noun phrase, the paraphrase in Sentence (10-c), where the second conjunct is realized in active voice, is very unnatural as *Marie* has two different syntactic functions in the coordination.

(10) a. Marie hat Äpfel geklaut und wurde von der Polizei erwischt.
Mary has apples stolen and was by the police caught.
"Mary has stolen apples and was caught by the police."
b.

HEAD (klauen)
ROLE (sem-subj,klauen,Marie:1)
ROLE (sem-obj,klauen,Apfel:2)
ROLE (sem-subj,erwischen,Polizei:3)
ROLE (sem-obj,erwischen,Marie:1)

1. *Original F-structure*:

   (7) Peter soll das Buch lesen.

$$\begin{bmatrix} \text{PRED} & \text{'sollen} < (\uparrow \text{SUBJ})(\uparrow \text{XCOMP}) > \text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Peter:1'} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'lesen} < (\uparrow \text{SUBJ})(\uparrow \text{OBJ}) > \text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Peter:1'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Buch'} \end{bmatrix} \\ \text{PASS} & - \end{bmatrix} \\ \text{PASS} & - \end{bmatrix}$$

2. *Bad alternation F-structure*:

   (8) <empty generation output>

$$\begin{bmatrix} \text{PRED} & \text{'sollen} < (\uparrow \text{SUBJ})(\uparrow \text{XCOMP}) > \text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Peter:1'} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'lesen} < (\uparrow \text{SUBJ})(\uparrow \text{OBL-AG}) > \text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Buch'} \end{bmatrix} \\ \text{OBL-AG} & \begin{bmatrix} \text{PRED} & \text{'Peter:1'} \end{bmatrix} \\ \text{PASS} & + \end{bmatrix} \\ \text{PASS} & - \end{bmatrix}$$

3. *Required alternation F-structure*:

   (9) Das Buch soll von Peter gelesen werden.

$$\begin{bmatrix} \text{PRED} & \text{'sollen} < (\uparrow \text{SUBJ})(\uparrow \text{XCOMP}) > \text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Buch:2'} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'lesen} < (\uparrow \text{SUBJ})(\uparrow \text{OBL-AG}) > \text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Buch:2'} \end{bmatrix} \\ \text{OBL-AG} & \begin{bmatrix} \text{PRED} & \text{'Peter:1'} \end{bmatrix} \\ \text{PASS} & + \end{bmatrix} \\ \text{PASS} & - \end{bmatrix}$$

Figure 5.5: F-structure candidates that illustrate the problem of surface cues that block the generation of an alternation candidate

c. ??Marie hat Äpfel geklaut und hat die Polizei erwischt.
Marie was apples stolen and has the police caught.

5.1.3 First Results and Error Analysis

We run a first small-scale experiment to test whether we can produce the additional alternation candidates using the simple transfer grammar from Figure 5.4. We considered a set of 156 German sentences extracted from the HGC, a huge German corpus of newspaper text. All contain a ditransitive verb that instantiates its three arguments. We expect that the size of candidate set for each sentence should at least be twice as big as compared to candidate set produced by generating from the F-structure.

In Table 5.1, we compare the number of surface realizations that are produced in generation from meaning representations and generation from usual F-structures. In both cases, the total average of surface realizations is very high due to some very long sentences in our test set. For the candidate sets obtained by generating from semantic representations the number of realizations increases by a factor of 2.8 on average as compared to the candidate sets generated from F-structures.

Avg. number of realizations for semantic input	25092.16
Avg. number of realizations for syntactic input	14168.57
Avg. increase of realizations per sentence	284%
Sentences with no increase in realizations	64
Total number of sentences	156

Table 5.1: Candidate sets produced in the pilot approach to extended LFG-based surface realization

However, Table 5.1 also shows that in 40% of the sentences, the number of surface realizations did not increase at all, which means that no voice alternations could be generated. Assuming that this small test set is representative to some extent, we have to expect that we will loose a substantial portion of our corpus data for the statistical modeling, due to the fact that the source representation for generation contains contextually specified surface cues that block the generation of alternation candidates.

Besides theses issues that concern the reconstruction of structural and lexical information for producing well-formed alternation F-structures, we

encountered some rather technical issues with the XLE generator. We found a number cases where the alternation F-structures could not be reconstructed due to specific, idiosyncratic constraints stemming from the grammar.

As an example, consider the sentence pair in Table 5.6. The analyses are produced by a German LFG grammar whose lexicon does not have an entry for the proper noun *Karthago*. XLE provides a "guessing" mechanism for unknown words. In this case, the German grammar has been set up to assume that unknown capitalized word forms are proper names, leaving the gender and number feature unspecified (since there are proper names for all genders and in singular and plural – like *Beatles*). As a consequence, the F-structure for *Karthago* in the passive sentence does not have a NUM feature since the number of the noun cannot be inferred from the syntax. By contrast, the F-structure for *Karthago* in the active sentence does have a NUM feature which comes from the inflectional morphology of the verb. So the two sentences have different meaning representations (if the meaning construction takes number into account).

Unfortunately, the XLE generator is very sensitive to slight changes in the F-structure input. If the mapping rules add a NUM feature to the F-structure in the passive sentence in Table 5.6 (which may seem to be a reasonable move), the generator fails because the structure that the grammar assigns to the sentence is no longer subsumed by the input representation. In practice, it is difficult to foresee and debug such problems.

A possible strategy for dealing with these idiosyncrasies would be to allow the mapping rules to heavily overgenerate and derive a range of hypothetical F-structures from the meaning representation. This strategy would exploit the fact that the XLE generator filters the ill-formed inputs anyway and in the final surface realization, these F-structures will not produce any surface sentence. However, we observe substantial performance problems, when the XLE generator has to deal with massive overgeneration and ill-formed input structures. By way of illustration, we contrast generation from an identical meaning representations based on two different reverse rewrite grammars that generate active and passive alternations for transitive verbs.

The transfer rules in Figure 5.4 and in Example (11) perform the same F-structure mappings in different ways. The transfer grammar in Figure 5.4 incorporates a notion of argument frames and will mostly produce F-structures that are well-formed and can be generated from. The rules in (11) on the will produce a lot of F-structures that are not compatible with LFG assumptions or specific grammatical/lexical constraints, e.g., F-structures

Rom wurde von Karthago erobert.
Rome was by Carthage conquered.

$$\begin{bmatrix} \text{PRED} & \text{'erobern} < (\uparrow \ldots)(\uparrow \ldots) > \text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Rom'} \\ \text{PERS} & 3 \\ \text{NUM} & \text{sg} \end{bmatrix} \\ \text{OBL}_{AG} & \begin{bmatrix} \text{PRED} & \text{'Karthago'} \\ \text{PERS} & 3 \end{bmatrix} \\ \text{PASS} & + \end{bmatrix}$$

Karthago eroberte Rom.
Carthage conquered Rome.

$$\begin{bmatrix} \text{PRED} & \text{'erobern} < (\uparrow \ldots)(\uparrow \ldots) > \text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Karthago'} \\ \text{PERS} & 3 \\ \text{NUM} & \text{sg} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Rom'} \\ \text{PERS} & 3 \\ \text{NUM} & \text{sg} \end{bmatrix} \\ \text{PASS} & - \end{bmatrix}$$

Figure 5.6: F-structure pair for passive-active alternation that illustrates the problem of idiosyncrasies in LFG-based generation inputs with XLE

with two subjects or without a subject.

(11)

```
AGENT(%V,%AG)
?=>
OBL-AG(%V,%AG).

PATIENT,(%V,%PA)
?=>
SUBJ(%V,%PA).

AGENT(%V,%AG)
?=>
SUBJ(%V,%AG).

PATIENT(%V,%PA)
?=>
OBJ(%V,%PA).
```

We used the set of 156 sentences for testing the grammars. In Table 5.2, we report the respective generation performance based on two different inputs for the surface realizer. The timeout parameter was set to 500 seconds.

As can be seen, the generator cannot easily deal with the F-structure chart input that contains a lot of ill-formed structures. It times out in 30% of the cases and the average generation time is dramatically increased compared to generation from mostly well-formed input.

	# F-structures	avg. generation time	# timeouts
Naive Rules	156	246.14 (110.68)	53
Informed Rules	156	36.20 (27.04)	3

Table 5.2: Evaluation of the generation performance for two possible F-structure transfer grammars used in the pilot approach

While some of the problems and examples mentioned in this Section might be specific to the XLE generator, other grammar-based generators seem to have similar problems with robustness and flexibility when processing externally defined inputs. Belz et al. (2011) report on the First Surface Realisation Shared Task where the grammar-based systems generally obtained poor results due to coverage problems.

5.1.4 Missing Arguments Block Candidate Generation

So far, our study of the passive alternation only considered corpus sentences were he verbs instantiate all their arguments. Thus, we ignored the fact that passive realizations of transitive verbs often occur without the optional, oblique agent.

Figure 5.7 presents two F-structures and meaning representations that would be derived by the LFG grammar and Crouch and King (2006)'s meaning construction for an active/passive paraphrase. In the semantic representation that is derived for the sentence which realizes a passive without an oblique agent in the original sentence, the representation does not specify an agent role. Our grammar-based realizer would only generate passive sentences from this representation as it needs an argument to comply with the syntactic constraints of the grammar.

To assess the frequency of passives that do not realize agents, we created a bigger data set with sentences extracted from the HGC corpus. The data set is described in more detail in Section 5.3. Table 5.3 summarizes the distribution of agents in passives and actives for 8044 F-structures from the data set. The overwhelming majority of passivized transitive verbs are "1-role" realizations, i.e. they do not realize an agent in a by-phrase. As a

Active: "Someone saw the chancellor."

$$\begin{bmatrix} \text{PRED} & \text{'see} < (\uparrow \text{SUBJ})(\uparrow \text{OBJ}) > \text{'} \\ \text{SUBJ} & [\ \text{PRED} \quad \text{'someone'}\ ] \\ \text{OBJ} & [\ \text{PRED} \quad \text{'chancellor'}\ ] \\ \text{TOPIC} & [\ \text{'someone'}\ ] \\ \text{PASS} & - \end{bmatrix}$$

HEAD (see)
PAST (see)
ROLE (agent,see,someone)
ROLE (patient,see,chancellor)

Passive: "The chancellor was seen."

$$\begin{bmatrix} \text{PRED} & \text{'see} < (\uparrow \text{SUBJ})(\uparrow \text{OBJ}) > \text{'} \\ \text{SUBJ} & [\ \text{PRED} \quad \text{'chancellor'}\ ] \\ \text{TOPIC} & [\ \text{'chancellor'}\ ] \\ \text{PASS} & + \end{bmatrix}$$

HEAD (see)
PAST (see)
ROLE (patient,see,chancellor)

Figure 5.7: F-structure and meaning representation pair for passive-active alternation: the argument realization specifies the syntactic voice of the verb in the meaning representation

	Active	Passive
2-role	82%	2%
1-role	0%	16%

Table 5.3: Distribution of transitive verb arguments (1-role transitive verbs realize the patient role, 2-role transitive verb realize the agent and patient role) and voice paraphrases in meaning representations derived without a treatment of implicit agents (SEM_n), data set used for extended LFG-based surface realization

consequence, the generator will produce active paraphrases for a very small portion of passive sentences (2%).

5.1.5 Discussion and Outlook

This pilot study shows that the extension of an established surface realization system that produces word order variants to syntactic alternations like the passive is non-trivial in a number of ways. Due to the fact that the generate-and-rank setting allows for direct inspection of the produced candidate sets, we discovered that F-structures derived in the intermediate

transfer generation often tend to encode certain contextual, structural or lexical, cues, that block generation of a surface realization when being passed to the grammar-based generator. Moreover, the procedure does not deal with implicit arguments that do, however, frequently occur in corpus data: Most occurences of transitive verbs are active, and most occurences of passive verbs do not realize the optional agent, see Section 5.2. This leaves a tiny portion of sentences where active and passive candidates can actually be generated when adopting straightforward mappings between meaning analysis and F-structures.

It is important to note that the effect of these interactions also depends on the underlying generation system. In a rule-based generate-and-rank architecture where hard constraints are directly modeled in the grammar, interactions like the *man* pronoun and passive voice are directly observable: given a naive way of mapping the underlying meaning representations to F-structure candidates, the generator will simply filter certain output candidates that would lead to ungrammatical output. As a consequence, the input to the statistical ranker does not contain much more paraphrases when it was produced from F-structures or meaning representations. In a completely statistical system, such as Bohnet et al. (2010), the number of possible, grammatical output candidates cannot be counted and compared between different types of input representations. In this set-up, it is expected that the system more or less trivially learns certain patterns as it never sees actual competitors in the data.

Thus, the sensitivity of the grammar-based generation with respect to the input representation it expects has methodological advantages and disadvantages for being extended to more abstract levels of generation. On the one hand, it is extremely useful to be able to test whether a generation input provides the required abstractions for generation and whether it yields the required candidates in the generation output. Moreover, we should not forget that the grammar is a rich resource of morpho-syntactic knowledge which allows for high output quality and grammatical candidates. On the other hand, in the light of our pilot experiments, it is not realistic to scale our extended LFG-based generation architecture to a deep semantic representation that would abstract from more phenomena involving more complex alternations in the predicate argument structure, such as nominalizations.

To adress this ambivalent situation, the subsequent Chapters propose two complementary methodologies and generation set-ups. Chapter 5.3 directly builds on our pilot study and implements a heuristic approach that is engi-

neered for the specifities of the active-passive alternation in our particular LFG-based surface realization architecture. The approach is heuristic as we deal with missing agents in passive corpus sentences and contextual cues in terms of a set of general rules that might not always produce a perfect reference to an agent of a passive verb. However, the approach yields highly accurate and grammatical candidate sentences. Chapter 5.4 adopts a more principled approach as we systematically integrate surface realization and referring expression generation. This approach is annotation based such that we obtain the correct representations of predicates and their explicit and implicit arguments. The annotations will be used in a statistical generation framework that does not model grammatical knowledge in terms of hard syntactic constraints and does not explicitly yield candidate sets.

5.2 Implicit Arguments

In Section 5.1.4, we have seen that the distribution of transitive verbs that instantiate both arguments is highly imbalanced between the active and the passive. Thus, if we want to build a model that deals with alternation, we have to find a solution for how to treat the problem of implicit agents in passive constructions. Consequently, our analysis procedure that abstracts from the syntactic realization information in an F-structures needs to, somehow, reconstruct implicit arguments to provide an appropriate source for the generator. But what does it mean for an argument to be implicit and how should a semantic representation account for it?

From a syntactic perspective, the deleted agent of a passive can be called implicit as it is not overtly realized, but still syntactically present (Bhatt and Pancheva, 2006). This implicit presence can be illustrated by the contrasts between passive and unaccusative uses of certain verbs in English. Example (12) shows that passives license *by*-phrases, whereas the unaccusative in (13) does not license the *by* phrase. On the basis of such tests, Bhatt and Pancheva (2006) argue that the agent in passives is "understood" or present in one way or the other, whereas it is not present in other constructions.

(12) a. The ship was sunk.
b. The ship was sunk by Bill.

(13) a. The ship sank.
b. *The ship sank by Bill.

Bhatt and Pancheva (2006) provide similar tests to show the presence of implicit arguments in nominalizations. In Example (14-a-b), the agent of the nominalized verb *attempt* can control the subject of *leave*.

(14) a. The attempt to leave
b. John made an attempt to leave.

From a semantic perspective, implicit arguments fall at least into two classes with respect to the understood entity that they refer to. Condoravdi and Gawron (1996) distinguish existential interpretations of implicit arguments from anaphoric interpretations. Example (15-a) would illustrate a case where the implicit object of *eat* receives an existential interpretation, which means that it does not refer to a specific entity in context. A possible paraphrase is sentence (15-b).

(15) a. There was a piece of bread on the table, but John didn't eat.
b. There was a piece of bread on the table, but John didn't eat anything.

Example (16-a) shows a case of an anaphoric implicit argument. The object of *apply* clearly refers to *the good job*, so sentence (16-b) is a paraphrase of (16-a).

(16) a. There was a good job available here but Fred didn't apply.
b. There was a good job available here but Fred didn't apply for it.

The main challenge posed by an empirical treatment of implicit arguments is that it is not always easy to infer the underlying meaning or referent of the implicit argument. Fillmore (1986) proposed an analysis for implicit objects that relates the type of referent to the semantics of the verb. He argues that certain verbs like *eat* that can easily omit the object would trigger implicit objects of the existential type whereas verbs like *apply* can only have an implicit object if its reference can be resolved to some entity mentioned in the previous discourse:

(17) a. John eats.
b. ?John applies.

However, Fillmore (1986) himself notes that this explanation does not seem

to generalize equally well over all verbs. In Examples (18) and (19), we see that the verb *return* allows the omission of certain anaphoric objects, whereas other objects cannot be easily made implicit.

(18) a. I returned to the camp.
b. I returned.

(19) a. I returned to the task.
b. *I returned.

Moreover, Condoravdi and Gawron (1996) state that even if the interpretation of the implicit argument is existential, the context can impose additional restrictions on the meaning of the argument. They give the Example in (20-a) where the object of the verb *bake* is implicit (it does not have an anaphoric referent in the previous context), but the context imposes the restriction that the existential argument has to be interpreted as *pastries*. Thus, the existential paraphrase in (20-b) is not entirely natural.

(20) a. I have been baking all week.
We needed a lot of pastries for the party.
b. ?I have been baking something all week.
We needed a lot of pastries for the party.

In computational and corpus-based NLP applications, implicit verb arguments has recently received attention for the analysis of semantic roles. Gerber and Chai (2010) annotate implicit arguments of some specialized nominalizations in the English NomBank by linking them to previous mentions in the context. Ruppenhofer et al. (2010) annotate an English novel with different types of implicit arguments in different contexts. Following Fillmore (1986), they distinguish indefinite null complements (i.e. existential implicits) and definite null complements (i.e. anaphoric implicits). Roth and Frank (2012) acquire automatic annotations of implicit roles for the purpose of studying coherence patterns in texts restricting themselves to the anaphoric type. For generation, we are only aware of Belz and Varges (2007) who treat subjects in coordinated verb phrases as implicit referents and annotate them in their REG data set. Thus, they only deal with anaphoric implicits in a very restricted context where the reference of the argument is completely determined by the syntactic construction.

5.3 A Context-aware Heuristic Approach

In an ideal corpus-based model of the active-passive alternation, one would like to use a meaning representation that perfectly captures the reference of implicit arguments. However, as we have discussed in Section 5.2, a representation of implicit arguments has to distinguish various types of implicit reference. When analyzing corpus data, it is not always possible to recover the missing agent of a passive since the author of the text decided not to mention it. We conducted some pilot annotation experiments where we asked annotators to identify types of implicit agents for passive verbs. But we could not achieve a satisfactory agreement among annotators. Example (21) presents a typical corpus sentence found in the Tüba-D/Z where it is hard to characterize the type of implicit agent of the verb *verletzen* (*injure*).

(21) *Context:* Ein gebrochenes Heiratsversprechen sorgte am Sonntag für Zoff bei zwei türkischen Großfamilien: Insgesamt 60 Personen prügelten sich auf einer Straße in Vegesack, so die Polizei .
"On Sunday, a broken promise of marriage caused trouble at two Turkish families: According to the police, 60 people in total had a fight on a street in Vegesack."

a. Vier Personen wurden leicht **verletzt**.
Four people were slightly injured.

b. ?*Paraphrase:* Dabei verletzten einige Teilnehmer der Prügelei vier Personen leicht.
"Some participants of the fight slightly injured 4 people."

c. ?*Paraphrase:* Man verletzte vier Personen leicht.
One slightly injured 4 people.

d. ?*Paraphrase:* Jemand verletzte vier Personen leicht.
"Somebody slightly injured 4 people."

e. *Paraphrase:* Ein besonders brutal agierender Cousin des Bräutigams verletzte 4 Personen leicht.
"A particularly brutal cousin of the groom slightly injured 4 people."

Given the previous discourse context in (21), it is clear that the agent of the injury refers to some subset of the fight's participants. However, the exact identity of the agent is not known and a natural active paraphrase is, maybe, Sentence (21-e), where more specific (invented) information about the injury

is given. However, this information cannot be generated or annotated if it is not specified in the context of the discourse.

In order to be able to exploit the rich morpho-syntactic knowledge encoded in the broad-coverage German LFG for extended surface realization, we carefully engineered a transfer grammar that maps F-structures which realize a transitive verb to their corresponding alternation F-structures. These rules are designed to account for a precise mapping between actives and passives, being aware of a range of contextual syntactic and lexical cues discussed in Section 5.1.2. Moreover, these rules deal with implicit agents in passive sentences by heuristically adding arguments to the alternation F-structure such that we achieve a less imbalanced distribution in our candidate data.

5.3.1 Transfer Rules for Context-aware Heuristics

In order to obtain identical underlying meaning representations for active and passive sentences, this work adopts a heuristic approach. We do not aim at annotating the exact reference of a missing agent in a passive sentence. The main idea of the heuristics is that we can identify a set of contexts for active sentences where we generate a passive paraphrase that deletes the agent. We treat these actives as "1-role" realizations. Moreover, we can identify a set of contextually specified agents for occurrences of passivized verbs, increasing the number of "2-role" passives. We implement:

- heuristics identifying potential agents in the context of a passivized verb
- heuristics identifying "deletable" agents of active verbs.

Concerning the heuristics that identify the reference of an implicit agent, we observe that in many corpus examples of passives where the agent is not specified in a *von* (*by*) phrase, the surrounding sentence context provides a concrete interpretation or reference for the implicit argument. In our pilot annotation experiment, we found corpus examples where the agent of a passivized verb is realized as an adjunct PP of the verb. This pattern seems to be frequent in cases where the agent is not animate, e.g. refers to some abstract or group entity:

(22) Danach wurde sie *am staatlichen Konservatorium für Musik und Gesang in Istanbul* **ausgebildet**.

"Afterwards she was **trained** *at the state academy for music and chant in Istanbul.*"

The transfer rules in Appendix A.1 list a set of prepositions that we mark as heading an agent of a passive verb. Below, we show the rule that applies to a passive F-structure were an adjunct of the verb has been marked as agent, and maps the adjunct to the subject of an active verb frame.

(23)

```
+GEN-CAT(%N,PASSIVE-ADJ-AGENT),
+ADJUNCT(%N,%Adjuncts),
in_set(%Adj,%Adjuncts), @agent_prep(%Adj),
PRED(%Adj,%%),PTYPE(%Adj,%%), arg(%Adj,%%,%LoSUBJ), OBJ(%Adj,%LoSUBJ),
SUBJ(%N,%SUBJ),
arg(%N,1,NULL), arg(%N,2,%SUBJ),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-ARG),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
OBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,-).
```

Example (24) illustrates the case of a passivized verb that is embedded under a *say* verb. This syntactic context triggers the interpretation that the agent of the saying is also the agent of the embedded verb.

(24) Eine internationale Ausschreibung der Anteile werde jetzt **eingeleitet**, erklärte *die Finanzbehörde* gestern.
"An international offering of the shares will be initiated now, declared the tax authorities yesterday."

a. *Paraphrase: Die Finanzbehörde* erklärte gestern, eine internationale Ausschreibung der Anteile werde sie jetzt **einleiten**.
"The tax authorities declared yesterday that they would initiate an international offering of the shares."

Correspondingly, the heuristics mark pronominal agents of embedded active verbs that can be deleted for generating passive paraphrases of the active sentence. Other evidence for agent deletions can be found, for instance, in parallel corpora. In Example (25), we show a translation pair from the Europarl corpus where a German *man* subject has been translated to an English passive.

(25) a. Hier muss man sich stärker auf die Kontrolle der Gase
Here must one self more on the control of the gases
konzentrieren, die bereits hergestellt wurden.
concentratem that already produced were.
b. More focus must be put upon controlling the gases that have already been produced.

Interestingly, in our analysis of the generation testsuite in Section 5.1.2, we found a number of cases where pronominal agents in active realizations block the generation of a passive paraphrase. On the other hand, the pronoun *man* seems to be an appropriate paraphrase for passives with implicit agents that receive an existential interpretation, as is illustrated in Example (26):

(26) Das Deutsche Rote Kreuz bittet jeweils von 15.30 bis 19.30 Uhr zum Aderlaß: Heute kann in Horn-Lehe, Bergiusstraße 125, Blut **gespendet werden**.
"The German Red Cross asks people to come to a bloodletting between 15.30 and 19.30 o'clock: Today, blood can be donated in Horn-Lehe, Bergiusstraße 25."
a. *Paraphrase:* Heute kann man in Horn-Lehe, Bergiusstraße 125, Blut **spenden**.

The transfer rule in (27) takes a "1-role" realization of a passive verb, and maps it to an active paraphrase that realizes *man* as the subject.

(27)

```
+GEN-CAT(%N,PASSIVE-1-ARG),
SUBJ(%N,%SUBJ),
arg(%N,1,NULL), arg(%N,2,%SUBJ),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-ARG-GENERIC),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
PRED(%LoSUBJ,pro), PRON-FORM(%LoSUBJ,man), NUM(%LoSUBJ,sg), PERS(%LoSUBJ,3),
OBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,-).
```

Furthermore, the transfer grammar shown in Appendix A deals with structural cues that might block paraphrase generation in a particular syntactic context. The following rule deals with agents of active verbs that are

part of coordination. In the corresponding passive alternation, the agent referent is deleted.

```
+GEN-CAT(%N,ACTIVE-2-AGENTCOORD),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
+OBJ-TH(%N,%ObjTh), +arg(%N,2,%ObjTh),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), nonarg(%N,1,%SUBJ),
NUM(%SUBJ,sg), PERS(%SUBJ,3),
PASSIVE(%N,+).
```

From an engineering perspective, the design of these transfer rules is time-consuming and requires careful inspection of the F-structure specifications and proper parameter settings of the XLE generator. While this is feasible for a controlled phenomenon such as the active-passive alternation, this solution does not seem promising for paraphrase types that require more modifications of the morpho-syntactic predicate properties, such as e.g. nominalizations.

5.3.2 The Data Set

The transfer grammar which is completely given in Appendix A is summarized below. The heuristics define:

- a set of pronouns that correspond to deletable agents of active realizations:
  - arbitrary reference and existential pronouns: *man, jemand, irgendjemand, etwas, irgendetwas*
  - neutral, third person pronouns: *das, es*
  - pronominal agents of embedded verbs, see Example (24)
  - pronominal agents of coordinated verbs
- a set of prepositional adjuncts in passive sentences that mapped to agents in the meaning representations;

We use the transfer grammar to parse corpus sentences and produce voice alternation candidates for these. In the following, we refer to it as SEM_h

SEM_h	Passive	Agent Type	Active	SEM_h
2-role	1.4%	Referential Argument	71%	2-role
	6.9%	Adjunct	-	
1-role	-	Generic	5.5%	1-role
	-	Neutral pronoun	0.5%	
	-	Embedded Pronoun	5.7%	
	8.9%	None	-	

Table 5.4: Distribution of transitive verb arguments (1-role transitive verbs realize the patient role, 2-role transitive verb realize the agent and patient role) and voice paraphrases in meaning representations derived with a heuristic treatment of implicit agents (SEM_h), data set used for extended LFG-based surface realization

for heuristic semantics. The candidates generated via the extended surface realization pipeline based on SEM_h cosntitute the training set for the surface realization ranker, which will be evaluated in Chapter 6.2 and 7.2.

We want to focus our experiments on the effect of voice alternations in a surface realization scenario. Therefore, in contrast to Cahill et al. (2007a) or the experiment described in Chapter 4.2, we do not use the TIGER treebank, the standard data set for German parsing and generation. Instead, we built our own set of input sentences and representations, making sure that all sentences of the data set contain at least one transitive verb that can be passivized. As a consequence, we do not have access to treebank-compatible annotations as Cahill et al. (2007a), but carry out surface realization on automatically disambiguated structures.

We extracted 19,905 sentences, all containing at least one transitive verb, from the HGC, a huge German corpus of newspaper text (204.5 million tokens). The sentences were parsed and automatically disambiguated. The resulting F-structure parses are transferred to meaning representations and mapped back to F-structure charts. For our generation experiments, we only use those F-structure charts that the XLE generator can map back to a set of surface realizations. This results in a total of 1236 test sentences and 8044 sentences in our train set. The data loss is mostly due to the fact the XLE generator often fails on incomplete parses, and on very long sentences.

Table 5.4 summarizes the distribution of active and passive paraphrases when we generate our candidates with the SEM_h semantics. The heuristic

rules change the distribution of alternations in our data, since they detect a lot of additional passives with an overt agent, alternating with an active. This change in distribution will have a strong effect on how the generation models learns to predict the alternations, as we demonstrated in Chapter 6.2 and 7.2.

A substantial portion of active occurrences is now treated as a 1-role realization, such that the number of "2-role" active decreases from 82% in SEM_n to 71% in SEM_h. Moreover, the number of "2-role" passive increases from 2% in SEM_n to 8% in SEM_h. Thus, for the "2-role" realizations, we still have a big majority for the active, but the passive voice not constitutes a more reasonable subset. For the "1-role" realizations, actives and passives are almost balanced in the distribution.

We are aware of the fact that these approximations introduce some noise into the data and do not always represent the underlying referents correctly. For instance, the implicit agent in a passive must not be "trivial" but can refer to an actual discourse referent in the context. However, if we did not treat the passives with an implicit agent on a par with certain actives, we would have to ignore a major portion of the passives occurring in corpus data.

5.4 A Multi-level Annotation-based Approach

The experiments with a "naively" extended surface realization architecture in Section 5.1 have shown that general-purpose semantic representations are not a suitable source for generating verb alternations. These representations do not deal with implicit arguments, and tend to specify a range of surface cues that block the realization of candidates, such as e.g. generic subject pronouns in active sentences. The XLE-based surface realizer is helpful for detecting such effects, but due to its very low robustness it is, ultimately, not realistic to extend it to a large-scale generation procedure that accounts for more syntactic alternations than the passive. In this Section, we describe a generation setting that goes beyond the extended LFG-based surface realization architecture from Section 5.3 along several dimensions:

First, we use a more flexible syntactic representation for our combined framework. We basically follow recent work in statistical linearization where the syntactic input to the generator is an unordered dependency tree (Bohnet et al., 2010, 2012).

Second, we propose to combine the two well-known paradigms of corpus-based referring expression generation and surface realization. Thus, most of the problems we have observed relate to interactions between the realization of verbal voice and the referring expressions of its arguments. In Section 5.3, we have developed heuristic rules that add, delete or map certain verb arguments in the underlying semantic representations. The approach presented in the current Section adopts a more principled solution to the problem.

Third, as we are not dealing with a reversible architecture where an analysis module produces an input for the generator, we create representations by annotating corpus sentences for referring expressions on the one hand and some type of semantic or deep syntactic representation on the other hand. This method has gained increasing popularity in recent approaches to data-driven generation and allows us to address exactly those choice phenomena that we want to model. While we have argued in Section 5.2 and 5.3 that the reference of implicit arguments is generally a hard annotation problem, our referent annotations are inspired from the GREC paradigm presented in Chapter 2.4. Following GREC annotation style, we focus on a text type that is strongly entity-centric and annotate explicit and implicit mentions of the central entities in this data.

Essentially, we argue that a meaning representation that only abstract from syntactic realization phenomena is not a proper source for inducing realistic corpus-based models of syntactic alternations. Instead, we need to reconstruct a abstract information layer where both types of choices, syntactic and referential expressions, are not specified in order to obtain a meaningful candidate space. We expect that the combined approach circumvents the problem of contextual cues that block the generation of candidates.

In the following, Section 5.4.1 defines the representations that will figure as input for the combined generation task and motivates the design decisions for the annotation style. Section 5.4.2 details the annotation of referents, Section 5.4.3 explains the dependency annotations designed to capture verb alternations.

5.4.1 Combining REG and Surface Realization

The main idea of our combined REG and surface realization framework is to define meaning representations as input for the generator that do not only lack information about surface syntactic structure, but also about the realization of referring NPs. Consequently, the candidates produced from a

particular source represent choices from several stages of the generation process, namely syntactic and a referential choice. To this end, we integrate the GREC-style referent annotations with deep syntactic representations commonly used for surface realization. Our syntactic annotations include deep and shallow syntactic relations similar to the representations used in recent surface realization shared tasks Belz et al. (2011).

Generally, the generation of subsequent references in discourse - which corresponds to the GREC paradigm - is especially relevant and challenging for text types with long entity chains where a certain referent is mentioned in a lot of contexts, such that a considerable variation in the corresponding referring expressions can be observed. Belz and Kow (2010) selected Wikipedia articles that have a single, human or non-human referent as their topic. For our case of multi-level generation, we had the specific interest of investigating interactions between referring expression choice and syntactic alternations. We decided on a narrow set of alternations (mainly voice and nominalization) which can be derived from the surface syntax by a set of rules. These alternations typically occur with transitive verbs, expressing a relation between two discourse referents. Therefore, we chose a text type that typically involves two main referents where many of the transitive verbs will express a relation between these: short newspaper articles describing crimes and robberies. Another important extension of the GREC-style annotations is that we include empty referents, as e.g. in passives and nominalizations directing attention to the phenomenon of implicit reference, which is largely understudied in NLG.

The data set for our generation experiments consists of 200 newspaper articles about robbery events. The articles were extracted from a large German newspaper corpus and are restricted to texts that describe an event involving two main referents, a victim and a perpetrator (and sometimes an additional source, see below). A complete example text is given in Figure 5.8. Each sentence contains at least one mention of the victim or perpetrator entity.

5.4.2 Annotating Referring Expressions

In the following, we describe the annotation of mentions of referents in our data set. The RE annotations mark explicit and implicit mentions of referents involved in the robbery event described in an article. The brat tool (Stenetorp et al., 2012) was used for annotation. We had 2 annotators with a computational linguistic background, provided with annotation guidelines.

(28) a. (Junge Familie)$_{v:0}$ auf dem Heimweg$_{poss:v}$ ausgeraubt$_{ag:p}$
(Young family) on the way home$_{poss:v}$ robbed$_{ag:p}$

b. Die Polizei sucht nach
The police looks for
[zwei ungepflegt wirkenden jungen Männern im Alter von etwa 25 Jahren]$_{p:0}$.
[two shabby-looking young men of about 25 years].

c. [Sie]$_{p:0}$ sollen am Montag gegen 20 Uhr
[They] are said to on Monday around 20 o'clock
(eine junge Familie mit ihrem sieben Monate alten Baby)$_{v:0}$ auf dem
(a young family with their seven month old baby) on the
Heimweg$_{poss:v}$ von einem Einkaufsbummel überfallen und ausgeraubt
way home$_{poss:v}$ from a shopping tour attacked and robbed
haben.
have.

d. Wie die Polizei berichtet, drohten [die zwei Männer]$_{p:0}$
As the police reports, threatened [the two men]
(dem Ehemann)$_{v:1}$, (ihn)$_{v:1}$ zusammenzuschlagen.
(the husband) (him) beat up.

e. [Er]$_{v:1}$ gab deshalb [seine]$_{v:1}$ Brieftasche ohne Gegenwehr$_{ag:v,the:p}$
[He] gave therefore [his] wallet without resistance$_{ag:v,the:p}$
heraus.
out.

f. Anschließend nahmen [ihm]$_{v:1}$ [die Räuber]$_{p:0}$ noch die Armbanduhr$_{poss:v}$
Afterwards took [him] [the robbers] also the watch$_{poss:v}$
ab und flüchteten$_{ag:p}$.
off and fleed$_{ag:p}$.

Figure 5.8: Example text with RE annotations in the robbery data set, oval boxes mark *victim* mentions, square boxes mark *perp* mentions, heads of implicit arguments are underlined

(29) a. Überfall auf (Tankstelle am Opelkreisel)$_{s:0}$
Attack on service station at Opelkreisel

b. Die Kasse (der Tankstelle am Opelkreisel)$_{s:0}$
The cash register the.GEN
ist jetzt bei einem nächtlichen Überfall
service station at Opelkreisel is now during a nighttime
ausgeraubt worden.
attack robbed been.
"The cash register of the service station at Opelkreisel has now been robbed during a nighttime attack."

c. [Der Täter]$_{p:0}$ bedrohte (den 22-jährigen Angestellten)$_{v:0}$ gegen 0.30
The perpetrator threatened the 22-year old clerk around 0.30
Uhr mit gezogener Pistole und forderte [ihn]$_{v:0}$ auf, das Geld
o'clock with pulled out pistol and asked him PART the money
herauszugeben.
give out.
"The perpetrator threatened the 22-year old clerk around 0.30 o'clock at the point of the pistol and asked him to give him the money."

Figure 5.9: Example text with RE annotations in the robbery data set, including *source* mentions

They were trained on a set of 20 texts.

Explicit mentions of referents are marked as spans on the surface sentence, labeled with the referent's role and an ID. We annotate the following referential roles: (i) *perpetrator* (*perp*), (ii) *victim*, (iii) *source*, according to the core roles of the *Robbery* frame in English FrameNet. As an example, in Figure 5.8, all mentions of the *victim* referent are marked with an oval box and the ID $v : 0$.

We include *source* referents since some texts do not mention a particular *victim*, but rather the location of the robbery (e.g. a bank, a service station). Thus, in contrast to *victim* and *perp*, this referent is always non-human. In Figure 5.9, we give an example of a text that mentions all three roles of the robbery frame.

In a lot of texts in our data set, the *perp* referent is not a single person, but refers to a group of people. Subsequent mentions of the referent in the text can then refer to a subset of the people included in the group. This phenomenon also occurs in the text in Figure 5.8, where the *victim* refers to the "young family" in Sentences (29-a-c). In Sentence (29-d), only the

(30) a. [19 Mann]$_{p:0}$ überfielen (Fußgänger)$_{v:0}$
19 men robbed pedestrian

b. Auf frischer Tat hat die Polizei am frühen Dienstagmorgen im
In fresh act has the police on early tuesday morning in the
Bahnhofsviertel [einen Franzosen und einen Algerier]$_{p:1}$ festgenommen,
Bahnhofsviertel a French and an Algerian arrested,
[die]$_{p:1}$ gemeinsam mit [einigen anderen Nordafrikanern]$_{p:2}$ einen
who together with some other North Africans a
37jährigen Fußgänger überfallen und beraubt hatten.
37-year old pedestrian attacked and robbed have.
"On early Tuesday morning, the police caught a French and an Algerian in the act in the Bahnhofsviertel, who had attacked and robbed 3 37-year odl pedestrian with some other North Africans."

c. Wie Polizeisprecher Jürgen Linker mitteilte, hatten Beamte der
As police spokesman Jürgen Linker announced, have officers the.GEN
SoKo Mitte gegen 4.40 Uhr [fünf Nordafrikaner]$_{p:3}$ beobachtet, als
SoKo Mitte around 4.40 o'clock 5 North Africans observed, as
[sie]$_{p:3}$ aus der Gallusanlage kamen und Richtung Hauptbahnhof gingen.
they from the Gallusanlage came and direction main station walked.
"Jürgen Linker, the police' spokesman, announced that the officers from SoKo Mitte observed 5 North Africans around 4.40 o'clock as they came from the Gallusanlage came and walked in the direction of the main station."

Figure 5.10: Example text with RE annotations in the robbery data set, illustrating a complex case of split antecedents

"husband", who is part of the family, is mentioned. To be able to deal with such cases, we introduce IDs that distinguish referents that have the same role, but do not have the same identity.

The overt referent instances in our data set occur in a variety of syntactic functions, such as in arguments of verbs, in prepositional adjuncts, as modifiers of nominalized verbs or as possessive modifiers. For an example of a possessive modifier, see the phrase "his wallet" in Sentence (28-e) in Figure 5.8. In Sentence (29-a), we have an example of an RE mentioning the *source* that modifies the nominalization "robbery". There are also cases where an RE occurs in a predicative construction, as is illustrated in Example (31). In these constructions, we only annotate the subject as a mention of a referent ("zwei Männer"), the predicative phrase "Stümper auf ihrem Gebiet" is not included in the annotation.

(31) Als Stümper auf ihrem Gebiet erwiesen sich zwei Männer$_{p:0}$,
As dilettantes in their field proved REFL two mean,
die$_{p:0}$ am Mittwoch gegen 20 Uhr versucht hatten, die
who on Wednesday around 20 o'clock tried have, the
Metro Tankstelle Am Riederbruch auszurauben.
Metro service station Am Riederbruch rob.
"Two men who tried to rob the Metro service station Am Riederbruch on Wednesday 20 o'clock turned out to be dilettantes in their field."

For each RE span, we annotate its syntactic head and its syntactic relation. For instance in Example (31), the verb *erwiesen* would be annotated as a verbal head that is linked to the RE "zwei Männer" via a subject relation. This complies with t he GREC data sets, and is also useful for further annotation of the deep syntax level (see Section 5.4.3). We distinguish the following types of syntactic heads:

- verb: subject, direct object, indirect object, oblique argument
- noun: PP modifier, possessive modifier
- nominalized verb: PP modifier, possessive modifier

The RE implicit mentions of *victim*, *perp*, and *source* are annotated as attributes of their syntactic heads in the surface sentence. We consider the following types of implicit referents:

- agents in passives, e.g. "robbed" in (31-a)
- arguments of nominalizations, e.g. "resistance" in (31-e)
- possessives, e.g. "watch" in (31-f)
- missing subjects in coordinations, e.g. "flee" in (31-f)

Table 5.5 reports inter annotator agreement that we on a set of 15 texts: the simple pairwise agreement for explicit mentions is 95.14%-96.53% and 78.94%-76.92% for implicit mentions.[1] We think that this is a very satisfactory agreement since the general annotation of implicit arguments is subject

[1]Standard measures for the "above chance annotator agreement" are only defined for task where the set of annotated items is pre-defined.

	Explicit ref.		Implicit ref.	
	A1	A2	A1	A2
A1	-	95.14	-	78.94
A2	96.53	-	76.92	-

Table 5.5: Pairwise annotator agreement for the annotation of explicit and implicit referents in the robbery data set

to subtle semantic effects (see Section 5.2). The GREC paradigm that focuses on central entities in a text seems to provide an effective solution to the problem.

5.4.3 Deriving Deep from Shallow Dependencies

We now define the syntactic annotation of our data which will provide the input representations for the surface realization and linearization component. The representation includes two layers: shallow and deep, labeled dependencies, similar to the representation used in surface realization shared tasks Belz et al. (2011). We use the Bohnet (2010) dependency parser to obtain an automatic annotation of shallow or surface dependencies for the corpus sentences.

In Figure 5.11, we show an example shallow dependency structure for a sentence from our data set. Each word is represented as a node in the dependency tree. Formally, a node is defined as a tuple t_s = *(id, lemma, pos$_s$, label, head, morph)*. In a dependency parse, the id of a word corresponds to its position in the sentence. For our generation purposes, we sort the nodes according to their position in the tree (see Section 5.4.1), so that the ids do not reflect the original surface order of the sentence. The *head* of a node corresponds to the id of its syntactic head or mother node and *label* to the corresponding syntactic relation.

Even when the information about the order of nodes is removed from the shallow dependencies, they contain a certain amount of surface syntactic information that needs to be removed in order to generate paraphrases like verb alternations. For instance, in contrast to F-structures, the dependency tree represents function words such as auxiliaries as nodes in the tree. As a consequence, the noun *Mann* (*man*) in Figure 5.11, is linked via a subject relation to the auxiliary node and not to the main verb *ausrauben* (*rob*). We want to obtain a deeper syntactic representation, similar to the F-structure

Shallow dependency annotation:

```
1  Ein  _  ein  _  ART  _  nom|sg|masc  -1  3  _  NK  _  _
2  53jähriger  _  53jährig  _  ADJA  _  nom|sg|masc|pos  -1  3  _  NK  _  _
3  Mann  _  Mann  _  NN  _  nom|sg|masc  -1  4  _  SB  _  _
4  ist  _  sein  _  VAFIN  _  sg|3|pres|ind  -1  0  _  --  _  _
5  am  _  an  _  APPRART  _  dat|sg|masc  -1  15  _  MO  _  _
6  Mittwoch  _  Mittwoch  _  NN  _  dat|sg|masc  -1  5  _  NK  _  _
7  abend  _  abend  _  ADV  _  _  -1  6  _  MNR  _  _
9  gegen  _  gegen  _  APPR  _  _  -1  15  _  MO  _  _
10  19  _  19  _  CARD  _  _  -1  11  _  NK  _  _
11  Uhr  _  Uhr  _  NN  _  *|*|fem  -1  9  _  NK  _  _
13  im  _  in  _  APPRART  _  dat|sg|masc  -1  15  _  MO  _  _
14  Rotschildpark  _  Rotschildpark  _  NN  _  dat|sg|masc  -1  13  _  NK  _  _
15  ausgeraubt  _  ausrauben  _  VVPP  _  _  -1  16  _  OC  _  _
16  worden  _  werden  _  VAPP  _  _  -1  4  _  OC  _  _
```

Deep dependency annotation with RE slots:

```
1  ausrauben  ausrauben  ausrauben  VV  VV  past|ind  past|ind  0  0  --  --
2  gegen  gegen  gegen  APPR  APPR  _  _  1  1  MO  MO  _  _
3  Uhr  Uhr  Uhr  NN  NN  _  _  2  2  NK  NK  _  _
4  19  19  19  CARD  CARD  _  _  3  3  NK  NK  _  _
5  an  an  an  APPRART  APPRART  _  _  1  1  MO  MO  _  _
6  Mittwoch  Mittwoch  Mittwoch  NN  NN  _  _  5  5  NK  NK  _  _
7  abend  abend  abend  ADV  ADV  _  _  6  6  MNR  MNR  _  _
8  in  in  in  APPRART  APPRART  _  _  1  1  MO  MO  _  _
9  Rotschildpark  Rotschildpark  Rotschildpark  NN  NN  _  _  8  8  NK  NK  _  _
10  victim  victim  victim  _  _  _  _  1  1  SEM_OA  SEM_OA  ref:6  _
11  perp  perp  perp  _  _  _  _  1  1  SEM_SB  SEM_SB  ref:empty  _
```

Figure 5.11: Deep and shallow dependency annotation for a passive sentence from the robbery data set

where the predicate arguments are directly linked to the main verb.

The deep syntactic dependencies are derived from the shallow layer by a set of hand-written transformation rules. The goal is to link referents to their main predicate in a uniform way, independently of the surface-syntactic realization of the verb. We address passives, nominalizations and possessives corresponding to the contexts where we annotated implicit referents (see above). The transformations are defined as follows:

Verb finiteness Auxiliary nodes are removed from the tree, all dependents of an auxiliary are attached to the verb. A simple tense feature distinguishes past and present for verb nodes, e.g. "haben:AUX überfallen:VVINF" (*have attacked*) maps to "überfallen:VV:PAST" (*attack:PAST*).

This transformation is a necessary preliminary to assign similar structures to the possible alternations of a verb as these usually involve differences in the morpho-syntactic realization. For instance, a passivized verb always has an auxiliary head in the shallow dependency representation which is not the case for active realizations of verbs in the present or past:

(32) a. Der Kanzler wurde vom Fotografen erwischt.
'The chancellor was catched by the photographer.'
The chancellor was by the photographer catched.

b. Der Fotograf erwischte den Kanzler.
The photographer catched the chancellor.

Note that this transformation is not necessary when a deeper syntactic representation, such as LFG F-structures in the previous Section 5.3, is used as the basis for deriving the input. In an F-structure, only the main predicates figure as "nodes" or PRED values in the representation whereas auxiliaries determine features like tense and mood. Thus, the transformation that removes auxiliaries from dependency trees can be seen as a simple approximation of a number of operations that are modeled in the projection between c-structures and F-structures in LFG.

Particles Particle nodes are merged with the node of their head verb lemma, e.g. "nahm" ... "fest" in (32-e) is mapped to "festnehmen" (*arrest*) . Similar to the previous transformation, this is necessary to underspecify the original morpho-syntactic realization of the verb.

(33) a. Der Täter wurde von der Polizei festgenommen.
The perpetrator was by the police arrested.
The perpetrator was arrested by the police.
b. Die Polizei nahm den Täter fest.
The police arrested the perpetrator PART.

Syntactic Functions Subjects in actives and oblique agents in passives are mapped to "semantic subjects" (i.e. agents). Objects in actives and subjects in passive to "semantic objects" (i.e. themes). e.g. *victim/subj was attacked by perp/obl-ag* maps to *perp/agent attack victim/theme*. A similar transformation is implemented in Section 5.1, for mapping LFG f-structures to meaning representations.

Figure 5.11 shows the final deep syntactic representation that we derive from the shallow dependencies for a passive sentence. The auxiliaries *sein* (node 4) and *werden* (node 14) are removed from the tree so that victim entity is a direct dependent of the main predicate *ausrauben* (*rob*). The perpetrator, which has been annotated as an implicit agent referent, is added as a semantic subject node to the tree.

Possessives We normalize prenominal and genitive postnominal possessives which have distinct labels in the shallow dependency representation. e.g. "seine (NK) Brieftasche" (*his wallet*) and "die Brieftasche (AG) des Opfers" (*the wallet of the victim*) map to "die Brieftasche POSS victim" (*the wallet of victim*). This transformation only applies if the possessive is included in the RE annotation, i.e. if we can vary the corresponding realization of the referring expression.

Nominalizations We map nominalizations to their verbal lemmas. For instance "Festnahme:NN" (*arrest*) maps to "festnehmen:VV". In contrast to the previous transformation, this mapping cannot be defined on the basis of the shallow dependency structure. Instead, we create a list that includes all the nominalizations that the annotators marked in the data set and map each nominal lemma to its verbal counterpart. The list contains mappings for 105 nominal types. Some frequent examples are given below:

(34)

Verfolgung (*prosecution*)	verfolgen (*prosecute*)
Überfall (*attack*)	überfallen (*attack*)
Fahndung (*search*)	fahnden (*search*)
Suche (*search*)	suchen (*search*)
Verurteilung (*condemnation*)	verurteilen (*condemn*)
Flucht(*escape*)	fliehen (*escape*)

In addition to these standard nominal types that directly correspond to a verb lemma, our annotators marked a couple of complex nominalizations which are in fact compounds of verb lemmas and some other, quite variable elements. Some examples are given below.

(35)
a. Zugüberfall → überfallen ("SEM_OA",Zug,NN)
train robbery → rob ("SEM_OA",train,NN)
b. Sofortfahndung → fahnden ("MO",sofort,ADV)
immediate search → search ("MO",immediately,ADV)
c. Einbruchsversuch → versuchen ("OC",einbrechen,VV)
attempted robbery → attempt ("OC",rob,VV)

For these complex nominalizations, we extend the binary mapping between nominal and verbal lemmas to include additional elements. These are represented as triples that specify the dependency relation with the verbal head, the lemma, and the PoS-tag.

The transformation for nominals involves two further important steps in order to create similar deep syntactic structures for verbal and nominal realizations. First, the prepositional and genitive arguments of nominalizations often correspond to the semantic subject or object of the underlying verb. e.g. *attack on victim* has to be mapped to *attack victim/theme*. The problem, especially with genitive arguments, is that they are ambiguous with respect to the underlying semantic function or role. This is illustrated in Example (36), where the genitive modifier of *Festnahme* (*arrest*) can either correspond to the agent or theme of the verb.

(36)
a. Festnahme der Polizei → festnehmen Polizei/agent
arrest the.GEN police → arrest police/agent
b. Festnahme der Täter → festnehmen
arrest the.GEN perpetrator → arrest
Täter/theme
perpetrator/theme

We solve this problem heuristically by defining a set of rules that assign a default role to genitive and prepositional arguments of nominalizations. The main idea is to first look for typical prepositional arguments that specify agents, e.g. *durch* or *von*. If the agent is not found in a prepositional argument, genitives are always mapped to agents, and to themes otherwise. Finally, we look for prepositional arguments that indicate an underlying object such as *Suche nach* or *Überfall auf*. These rules do not always establish the gold argument frame of the verb. In order to obtain a clean representation, manual encoding of all representations would be necessary.

Second, we have to apply transformations to the syntactic context of the nominalization. Due to their non-finite, nominal morpho-syntax, they typically occur in other structures as embedded verbs or main verbs of a clause. As an example, consider the paraphrases in Example (37), where an embedded clause head by the conjunction *nachdem* can be alternatively realized as a nominalization headed by the preposition *nach*.

(37) a. Die Polizei kam 2 Stunden nachdem die Tankstelle
The police came 2 hours after the service station
überfallen wurde.
attacked was.
"The police came 2 hours after the service station was attacked.

b. Die Polizei kam 2 Stunden nach dem Überfall auf die
The police came 2 hours after the attack on the
Tankstelle.
service station.

If we do not deal with the preposition *nach* in our transformation, the generator would have a strong clue for a nominal realization of *überfallen* in Example (37). In fact, this is just another instance of contextual cues specified representations in corpus-based NLG (see Section 5.1.2): the surrounding context of a certain choice often determines its syntactic realization if the procedure for deriving an underlying semantic representation does not explicitly deal with this context. In order to obtain representations of nominalizations that "look like" verbal representations we define rules for three types of contexts:

- nominalizations headed by prepositions are mapped to embedded clauses head by a conjunction, we define a mapping between a set of prepositions and their corresponding conjunctions

- nominalizations headed by light verbs are merged with this verbal node, figuring as an alternative realization of a main verb:

(38) a. Die Männer ergriffen die Flucht.
The men went on escape.
b. Die Männer flohen.
The men escaped.

- nominalizations headed by certain clause-embedding verbs are mapped to embedded *dass* (*that*) clauses:

(39) a. Die Jugendlichen bestreiten den Überfall auf die Tankstelle.
The adolescents deny the attack on the service station.
b. Die Jugendlichen bestreiten, dass sie die Tankstelle überfallen haben.
The adolescents deny, that they the service station attacked have.

Again, these rules have to be seen as heuristics that do not always yield perfect representations. For instance, the preposition *nach* is ambiguous in a way that it does not always correspond to the conjunction *nachdem*, e.g. in the phrase *nach Angaben des Opfers, ...*

The nominalization is mapped to a verb and we change the PP embedded by *durch* to an embedded clauses headed by *indem*. Moreover, we included the RE annotation for the sentence so that the verb *überfallen* has a semantic subject and object.

5.4.4 The Data Set

The representations of our data set that will constitute the input for the generator are defined in terms of several layers. We mainly distinguish a deep and a shallow dependency layer such that the generation process can be modularized in a way that linearization can be treated separately from alternation modeling. In both layers, the realization of referring expressions

# sentences	2030
# explicit REs	3208
# implicit REs	1778
# passives	383
# nominalizations	393
# possessives	1150

Table 5.6: Basic annotation statistics for the robbery data set

is underspecified, i.e. RE subtrees are replaced by abstract role labels.[2] The extracted RE subtrees are kept in a separate layer which is basically a list of all possible candidates for a referent. These candidates are aligned with their original slots in the deep and shallow dependency tree. The candidate list for each referent which is initialized with three default REs: (i) a pronoun, (ii) a default nominal (e.g. "the victim"), (iii) the empty RE. In contrast to the GREC data sets, our RE candidates are not represented as the original surface strings, but as non-linearized subtrees.

The resulting multi-layer representation for each text is defined as follows:

1. unordered deep trees with RE slots ($deepSyn_{-re}$)
2. unordered shallow trees with RE slots ($shallowSyn_{-re}$)
3. unordered RE subtrees
4. linearized, fully specified surface trees ($linSyn_{+re}$)
5. alignments between nodes in 1., 2., 4.

Table 5.6 summarizes some statistics on the data set.

5.5 Conclusions

This Chapter has presented research on deriving or reconstructing input representations that constitute an appropriate source for generation systems in a

[2]Since we use automatic dependency annotations for our shallow dependency layers, certain manually annotated RE spans do not correspond to a proper subtree. In these cases, we take the left-most head of the span according to the dependency annotation, attach the following heads and replace this subtree by the role label.

corpus-based setting. Already McDonald (1993) stated that "The most vexing question in natural language generation is 'what is the source'– what do speakers start from when they begin to compose an utterance?" While this view offers a psycholinguistic perspective on the problem, the recent literature on corpus-based generation has discussed input representations from the perspective of system comparability and evaluation (White and Rajkumar, 2009; Belz et al., 2010; Guo et al., 2011; Wanner et al., 2012).

Our discussion of extended surface realization systems that deal with syntactic alternations provides a perfect illustration for the vexing question of choosing a source for generation. While the previous Chapter 4 has analyzed interactions between choices on the level of factors encoded in a context model and found that a range of sentence-internal morpho-syntactic cues reflect discourse-level context, we have now described the same basic problem for the dimension of the generation source. Since the usage of sentence-level, syntactic constructions and their discourse-level function is so closely intertwined, corpus-based representations obtained from NLU analysis tools tend to reflect these interactions. Therefore, we have proposed techniques to deal with paraphrases like passive or nominalizations in generation that does not only abstract from the immediate morpho-syntactic properties of a verb, but also deals with the referential choice of verb arguments and other contextually specified cues.

The careful study of syntactic alternations for corpus-based generation suggests that the definition of input representations is much more than a matter of establishing annotation standards that would ensure system comparability. Actually, it turns out to be a fundamental research question that relates to the well-known problem of interacting discourse-level and sentence-level choices in language production. Standard methods for deriving generation inputs from syntactic analyses more or less assume that the source of syntactic choice can be appropriately captured by removing surface syntactic cues from an analysis representation. This assumption would basically imply that syntactic choices do not interact with generation choices of other types.

Moreover, this Chapter has shown that the pathway to more abstract generation inputs and settings where more choice phenomena can be studied involves non-trivial decisions at the level of the generation framework. On the one hand, we have seen a grammar-based generator that encodes a rich inventory of morpho-syntactic constraints and produces sound, grammatical sets of candidates. However, in order to deploy the potential of an underlying grammar, we have to produce very specific input representations

and manually engineer mapping rules between semantic and syntactic representations. As a more general and scalable solution that circumvents the sensitivity of grammar-based generation, we have presented an annotation-based approach where representations are created for the specific needs for the generation. The framework which lends itself to such generation inputs are statistical generators that can be trained on virtually every pairing of input representations and output sentence. On the other hand, such systems do not explicitly represent hard syntactic constraints on argument realization and is based on a more shallow syntactic representation, it is an open question how they are able to deal with additional uncertainty introduced by e.g. implicit arguments.

More generally, the work presented in this Chapter calls into question the common assumption that existing surface realization systems are based on syntactic inputs that can be easily interfaced with by other application-oriented systems. As we have sketched in the Introduction of this thesis, the practical motivation for surface realization systems is to provide off-the-shelf, large-scale models that deal with a range of subtle, context-dependent linguistic choices and that can be simply plugged into small-scale domain-specific systems. Instead, our observations suggest that these practical systems would already need to predetermine a range of linguistic decisions (the realisation of referents) in order to produce representations that can be processed in state-of-the-art surface realizers. This is undesirable, as this would ultimately mean that relatively subtle surface decisions such as the realization of voice would be implicitly specified by deeper modules in the domain-specific application that is likely to abstract from the whole range of possible context factors. Thus, clearly, syntactic representation as they are used and produced in NLU tasks and systems do not necessarily provide an appropriate common ground for NLG systems. In that respect, the approach presented in this Chapter can be seen as a way to make surface realization more attractive for practical systems, as it is aware of the various interactions between choices – a well-known and notorious problem in NLG research and applications. More work is needed to establish whether the representations we propose here can ease the generation task for actual, domain-specific NLG systems.

Chapter 6

Evaluating the Source in Models of Extended Choice

In the previous Chapter 5, we have presented procedures for deriving representations from the corpus-based analysis of sentences, that provide a more abstract source for generation than the commonly used representations in surface realization and referring expression generation (REG). We have extensively argued that the source for generating syntactic alternations like passives and nominalizations needs to be carefully derived and annotated in order to yield a consistent set of candidates representing a particular set of choices. Thus, we suggested that input representations for generating syntactic alternations should capture implicit arguments as well as a range of interactions between syntactic and referential choices.

The aim of the experimental work described in the current Chapter is to test whether these annotation decisions at the level of the generation source pay off at the level of statistical modeling for candidate selection. Whereas our argument in Chapter 5 was based on the assumption that an appropriate source should lead to a significant extension of the choice space at the stage of candidate generation, we now look at the corresponding effects for the ranking component in surface realization or REG.

But how can we assess whether extended candidates produced from a,

theoretically, more appropriate source actually improve candidate selection in some way? While this question directly follows from our discussion in Chapter 5, it has intricate implications for a corpus-based evaluation of our generation frameworks: standard evaluations in NLG and NLU are usually carried out so that competing systems are compared on a given input. In our case, we want to compare a particular system on alternative inputs. This is even more delicate when we consider that a more appropriate source will yield more candidates, meaning that the selection task is, a priori, harder as compared to selection from smaller candidate sets.

Evaluation and Frameworks Moreover, it has to be kept in mind that the generation frameworks that we work with in this thesis - an LFG-based generate-and-rank system, a statistical dependency-based realiser and an REG system - are based on different notions of generation candidates. In the LFG-based system, we could directly assess whether a certain generation inputs yields a wider set of candidates for the ranking step, showing the direct impact of a treatment of implicit referents in the generation input. The evaluation aims to show that the resulting candidate sets also lead to a more appropriate candidate selection model for syntactic alternations. In the dependency-based setting, we have purposefully defined and annotated a representation for extended surface realization, but the system does not explicitly represent fully-fledged candidates as sentences. Here, we will show how an actual empirical evaluation can assess the effect of representing implicit referents in the generation input. For an REG system, the decision to include implicit referents adds one particular candidate (i.e. the empty realization) to the candidate sets for each referent slot. In this case, the evaluation will show that implicit referents substantially increase the uncertainty of the REG problem and pose new requirements on the involved feature models.

Section 6.1 discusses the hypothesis of our evaluation and the effects that we expect in more detail. Section 6.2 compares surface realization ranking for word order and voice alternations generated from naive versus heuristically derived representations in the extended LFG-based framework. Section 6.3 evaluates classifiers that predict passives and nominalization on dependency trees, assessing the effect of implicit referents. In Section 6.4, we present an REG experiment that extends the GREC set-up and includes implicit referents.

6.1 Evaluating in the Presence of Variable Input

A major motivation for corpus-based generation is to test and evaluate different generation methods in a common, broad-coverage domain on a generally accessible data set. Thus, the rise of statistical methods in NLG has led to a considerable amount of surface realizers being tested and compared on annotations of the Penn Treebank (Langkilde-Geary, 2002; Ringger et al., 2004; Zhong and Stent, 2005; Cahill and Van Genabith, 2006; White and Rajkumar, 2009; Guo et al., 2011). However, an objective comparison between all these approaches still turns out to be difficult as the various generators use different transformations of the Penn Treebank annotations (White and Rajkumar, 2009; Belz et al., 2010; Guo et al., 2011).

Some typical sources for divergences between surface realization inputs are the specificity of syntactic labels, the presence of function words, or the specificity of morphological realization. For instance, some more surface-oriented syntactic representations distinguish labels for premodifiers or postmodifiers such that the order of modifiers is already specified to a certain extent. Similarly, some representations, such as the CoNLL dependency format, distinguish labels for the first and second conjunct in a coordination. In a deep syntactic annotation, such as an LFG F-structure, the order of conjuncts and modifiers is not specified.

In the following, we will summarize some studies in corpus-based generation that have reported results obtained from applying a particular generation system, and statistical model for candidate selection, on different inputs.

Langkilde-Geary (2002) systematically evaluates the HALogen surface realization ranker, the successor of Langkilde and Knight (1998)'s system, on Section 23 of the PennTreebank, using inputs that instantiate several degrees of underspecification. She reports that the BLEU scores vary between 92.4 for the most specific input and 51.4 for the minimally specific input.

Velldal (2008) reports on HPSG-based generation experiments for English where he contrasts generation from meaning representations that are underspecified or specified for voice and topicalization. As one would expect, the underspecified representations trigger much more (about twice as many) surface realization candidates. In their case, the decrease in accuracy of the ranking component is less striking than the effects found by Langkilde-Geary (2002). The ranker achieves a NEVA score (a variant of BLEU) of 94.1 on

the specified input, and 92.3 on the underspecified version.

Guo et al. (2011) test a general dependency-based generator on two different dependency versions of the treebank, an LFG-based and a CoNLL-based input, and find a difference of 8 BLEU points - 80.65 on the more abstract LFG input and 88.2 on the CoNLL input - when applying the same model.

Given these insights about differences between generation inputs and lack of comparability between systems, there have been some recent efforts in the NLG community to establish representation and annotation standards. Belz et al. (2011) develop a common ground representation derived from the Penn Treebank, as a basis for the First Surface Realization Shard Task where 5 systems have been compared on an identical input. The representation has a shallow and a deep syntactic layer where the shallow input corresponds to the CoNLL 08 dependencies and the deep layer is derived from this by removing information about part-of-speech and function words. Bohnet et al. (2011) submitted the only system that could be applied to both the shallow and the deep annotation layer. Their deep sentence surface realizer achieves a BLEU score of 89.6 on the shallow input, and 79.6 on the less specified input.

Wanner et al. (2012) suggest that Belz et al. (2011)'s common ground input is not completely satisfactory as a standard for surface realization. According to their view, a representation for surface realization should abstract from any type of syntactic information whereas the Belz et al. (2011) input still specifies certain syntactic phenomena such as the order of conjuncts in coordination. Similar to our discussion in the previous Chapter 5, Wanner et al. (2012) argues that an appropriate representation for surface realization should be created through clean and careful, manual annotation that targets the generation application. They propose a range of principles ("Semanticity","Informativity","Connectedness") that served as a guide for a deeper semantic annotation of the Penn Treebank.

In terms of evaluation, Wanner et al. (2012) state that "the removal of syntactic features from a given standard annotation, with the goal to obtain an increasingly more semantic annotation, can only be accepted if the quality of (deep) stochastic generation does not unacceptably decrease". They report a BLEU score of 64, which is, however, hard to put in perspective as comparable experiments on the same input do not exist.

What this small review of evaluations that have looked at variable input definitely shows is that decisions at the level of syntactic annotations can have a big impact on the difficulty of the generation problem and as a result the expected performance of the system can vary substantially. However, it

seems to be an open question in the NLG community how this variability and the interdependency between information specified in a source for generation and its effects on candidate selection can be addressed and interpreted systematically.

6.2 Experiment 4: Extended Candidates in Realization Ranking

This Section describes a study on statistical modeling of choice for active-passive alternations and word order variation in a grammar-based generate-and-rank scenario. As explained in Chapter 5.1, we extended Cahill et al. (2007a)'s LFG-based surface realization architecture with an intermediate mapping between F-structures and a semantic representation. The semantic representation removes syntactic information about argument realization from LFG F-structures such that voice alternations can be generated in addition to word order variation. Example (1) illustrates a candidate set that is generated from the standard F-structures and comprises only word order variation.

(1) `schenken< SUBJ(Thomas), OBJ-TH(Maria), OBJ(Buch)>`

a. Maria schenkt ein Buch Thomas.
b. Maria schenkt Thomas ein Buch.
c. Ein Buch schenkt Thomas Maria.
d. Ein Buch schenkt Maria Thomas.
e. Thomas schenkt ein Buch Maria.
f. Thomas schenkt Maria ein Buch.

Being able to generate from a more abstract representation that does not specify voice, we additionally obtain the surface realizations in (2) illustrating all possible permutations in passive voice (where in German only the theme can be turned into the passive subject).

(2) `schenken(agent:Thomas,theme:Buch,beneficient:Maria)`

a. Maria wird von Thomas ein Buch geschenkt.
   Maria.DAT is by Thomas a book.NOM given.
b. Maria wird ein Buch von Thomas geschenkt.
c. Ein Buch wird Maria von Thomas geschenkt.

d. Ein Buch wird von Thomas Maria geschenkt.
e. Von Thomas wird Maria ein Buch geschenkt.
f. Von Thomas wird ein Buch Maria geschenkt.

The impact of syntactic alternations like voice on realization ranking in free word order languages has so far not been investigated in computational frameworks working with reversible grammars.

6.2.1 Inputs

In Chapter 5.1, we have first described a naive procedure for semantic representations to F-structures, which basically maps grammatical arguments to semantic roles and removes some syntax-internal features such as case. This semantics will be called "SEM$_{shallow}$" in the following.

The "SEM$_{shallow}$" transformation on F-structures does not account for the fact that many passive sentence in corpus data do not overtly realize the agent argument. Moreover, it does not capture certain interactions between the referential surface form of a verb argument that blocks the generation of a respective voice alternation, e.g. the case of the *man* (*one*) pronoun which cannot be mapped to an oblique agent.

We addressed these observations in a second heuristic derivation procedure that will be called "SEM$_{deep}$" in the following. The semantic representations derived by "SEM$_{deep}$" heuristically capture some implicit arguments and additionally remove some contextual cues such that a wider range of candidates can be generated, see Chapter 5.3.

Thus, we have 3 alternative sources for producing candidates sets and training candidate selection in our extended surface realization architecture:

- FS: candidates generated from the F-structure
- SEM$_{shallow}$: candidates generated from the naive meaning representations
- SEM$_{deep}$: candidates generated from the heuristically underspecified meaning representation.

The main idea of the experiment is to keep the set of original corpus sentences constant, but train and test the model on different candidate sets: The FS candidate sets only comprise word order variation. The SEM$_{shallow}$ sets

represent choice on the level of voice and word order, but only in structures where all verb arguments can be instantiated from the original realization in the corpus sentence, meaning that most of the sentences where both voices can be realized were originally produced in active voice. The candidate sets generated from the SEM_{deep} source show a different distribution of active and passive sentences, as it comprises also active paraphrases of sentences that were originally realized in passive with an implicit agent, and passive paraphrases of originally active sentences were $\text{SEM}_{shallow}$ does not capture certain blocking effects.

We train the ranking model on a set of 8044 sentences and test it on a set of 1236 sentences. See Chapter 5.3.2 for details on the data set.

6.2.2 Experimental Set-up

Labeling For the training of our ranking model, we have to tell the learner how closely each surface realization candidate resembles the original corpus sentence. We distinguish the ranks: "1" identical to the corpus string, "2" identical to the corpus string ignoring punctuation, "3" small edit distance (< 4) to the corpus string ignoring punctuation, "4" different from the corpus sentence. The intermediate ranks "2" and "3" are useful since the grammar does not always regenerate the exact corpus string, see Cahill et al. (2007a) for explanation.

Features Given a set of surface realizations for an input meaning representation, we annotate each candidate with features extracted from the underlying F-structure and meaning representation, the surface string of the sentence and a language model score.[1] The feature model is built as follows: for every lemma in the F-structure, we extract a set of morphological properties (definiteness, person, pronominal status etc.), the voice of the verbal head, its syntactic and semantic role, and a set of informations status features following Cahill and Riester (2009). These properties are combined in two ways: a) Precedence features: relative order of properties in the surface string, e.g. "theme $<$ agent in passive", "1st person $<$ 3rd person"; b) Non-precedence features: combinations of voice and role properties with morphological properties, e.g. "subject is singular", "agent is 3rd person in

[1]The language model is trained on the German data release for the 2009 ACL Workshop on Machine Translation shared task, 11,991,277 total sentences.

active voice" (these are surface-independent, identical for each alternation candidate).

Evaluation Measures In order to assess the general quality of our generation ranking models, we use several standard measures:

- Exact match: how often does the model select the original corpus sentence
- BLEU: n-gram overlap between top-ranked and original sentence
- NIST: modification of BLEU, gives more weight to less frequent n-grams

Second, we are interested in how well our model predicts voice alternations in particular. Since the above measures only capture n-gram overlap, BLEU scores will not be affected very much if e.g. the model always predicts the incorrect voice, but produces the right word order. Therefore, we report the following accuracy:

- Voice: how often does the model select a sentence from the original F-structure

6.2.3 Results

In Table 6.1, we report the average number of strings for each item that the ranker has to deal with in the different generation inputs. In the SEM_{deep} input, the average number of candidates is more than doubled as compared to the standard FS input.

We also report the performance of a random choice and a language model (LM) baseline. A first interesting effect is that the language model performs almost equally well on the FS and the $\text{SEM}_{shallow}$ input although there is a clear drop in the random choice baseline due to the increased number of candidates. However, we observe a clear decrease in performance of the language model on the SEM_{deep} input while the drop in the random choice baseline is less sharp from $\text{SEM}_{shallow}$ to SEM_{deep}. This indicates that the SEM_{deep} semantics introduces some variation into the candidates that cannot be easily captured by local surface-oriented features.

	Avg. #	Random	LM		
Input	strings	Exact Match	Exact Match	BLEU	NIST
FS	36.7	16.98	15.45	0.68	13.01
$\text{SEM}_{shallow}$	68.2	10.72	15.04	0.68	12.95
SEM_{deep}	75.8	7.28	11.89	0.65	12.69

Table 6.1: Experiment 4: Language model baseline evaluation on candidate sets for realization ranking generated from F-structure input (FS), meaning representations not mapping implicit arguments ($\text{SEM}_{shallow}$), meaning represenations that heuristically specify deep argument frames (SEM_{deep})

Compared to the huge variance in BLEU score that has been reported in other articles that compared statistical candidate selection on different inputs (see Section 6.1), we still observe a relatively small decrease in BLEU for our SEM_{deep} candidate sets. However, the extension of our candidates is very controlled, only affecting the morpho-syntactic realization of a particular predicate in the sentence. It is expected that the BLEU score is not very sensitive to these changes, as it only measures n-gram overlap. Thus, in our case, the Exact Match measure might be more telling as it really shows that the language model's predictions are less exact in terms of the sentences it selects.

The performance of the linguistically informed model on the candidates sets is shown in Table 6.2. Generally, it largely outperforms the language model on all generation inputs. The differences in BLEU between the candidate sets and models are statistically significant.[2] Moreover, the linguistic model is less sensitive to the additional confusion introduced by the SEM_{deep} input. Its BLEU score and Exact Match accuracy decrease only slightly (though statistically significantly).

As the BLEU and Exact Match measures are very coarse-grained, we compare the models with respect to the voice accuracy. Since the $\text{SEM}_{shallow}$ input contains voice alternations only for a subset of the sentences, we also computed the accuracy on the subset $\text{SEM}_{shallow}*$ that excludes items where the voice is specified. Table 6.3 shows the voice accuracies, the proportions of items where the voice is specified and the majority baseline which always predicts the most frequent voice (active). According to the way we designed the heuristics for dealing with implicit arguments, we distinguish 2-role al-

[2]According to a bootstrap resampling test, $p < 0.05$

	Linguistic Model		
Input	Exact Match	BLEU	NIST
FS	27.91	0.764	13.18
$\text{SEM}_{shallow}$	27.66	0.759	13.14
SEM_{deep}	26.38	0.747	13.01

Table 6.2: Experiment 4: Evaluation of the linguistically informed ranking model on candidate sets for realization ranking generated from F-structure input (FS), meaning representations not mapping implicit arguments ($\text{SEM}_{shallow}$), meaning represenations that heuristically specify deep argument frames (SEM_{deep})

ternations where both arguments of transitive verb had been realized in the corpus sentence, and 1-role alternations where the SEM_{deep} semantics only represents one referential argument, see Chapter 5.3.

In the $\text{SEM}_{shallow}$ input, the majority baseline is very high, and the model does not outperform this baseline, even if the evaluation is restricted to items where a voice alternation can be generated. It is not able to make predictions for the 1-role alternations as the representations do not capture implicit and non-referential arguments. In contrast, the SEM_{deep} model largely outperforms the active baseline, achieving a much higher accuracy on the prediction of passives. In particular, the accuracy on the 1-role alternations is 90.3%, largely exceeding the majority baseline of 59.2%. The prediction on 2-role alternations is harder as the majority baseline is at 88% for the active realization, but the model also clearly outperforms the baseline for these cases.

Example (3) illustrates a case from our development set where the $\text{SEM}_{shallow}$ model incorrectly predicts an active (3-a), and the SEM_{deep} correctly predicts a passive (3-b). The preference for the passive follows from the markedness hierarchies discussed in Chapter 3.3: the patient is singular and definite, the agent is plural and indefinite.

(3) a. 26 kostspielige Studien erwähnten die Finanzierung.
26 expensive studies mentioned the funding.

b. Die Finanzierung wurde von 26 kostspieligen Studien erwähnt.
The funding was by 26 expensive studies mentioned.

Paraphrase	Input	Acc.	Spec.	Majority BL
All	FS	100	100	-
	$\text{SEM}_{shallow}$	98.06	22.8	98.49
	$\text{SEM}_{shallow}$*	97.59	0	98.23
	SEM_{deep}	91.05	0	83.4
2-role	FS	100	100	-
	$\text{SEM}_{shallow}$	97.7	8.33	98.49
	$\text{SEM}_{shallow}$*	97.59	0	98.23
	SEM_{deep}	91.8	0	88.5
1-role	FS	100	100	-
	$\text{SEM}_{shallow}$	100	100	-
	$\text{SEM}_{shallow}$*	100	100	-
	SEM_{deep}	90.0	0	53.9

Table 6.3: Experiment 4: voice accuracy of the top-ranked surface realizations predicted by the linguistically informed models on different generation inputs with respect to the argument frame in the meaning representation, "Spec." is the proportion of generation inputs where the representation predetermines the voice of the transitive verbs, the Majority baseline corresponds to always predicting "active voice"

On the one hand, the rankers can cope surprisingly well with the additional realizations obtained from the meaning representations. According to the global sentence overlap measures, their quality is not seriously impaired. This result should be treated with care, however, as these measures are completely agnostic about the underlying syntax of the selected candidate. Basically, BLEU scores will only decrease due to mismatches between the surface forms of the verbal and auxiliary tokens, assuming that the additional choice introduced by voice alternations does not seriously affect the prediction of word order. This assumption, which we have not explicitly addressed in this experiment, is far reaching as it directly touches upon the interaction between word order and voice in our model of choice. This issue is investigated in another experiment reported in Chapter 7.3.

The more detailed phenomenon-oriented evaluation in terms of Voice Accuracy has clearly shown that the design of the generation source has a substantial effect on the prediction of choice. The model trained on the $\text{SEM}_{shallow}$ candidates does not learn contextual preferences that distinguish between the two alternations, because of the extremely imbalanced distribution in the input data. If the underlying distribution is reasonably changed by sentences that have to be ignored in a shallow treatment of argument realization - note that the majority of sentences still realizes the active voice - the model seems to grasp certain contextual preferences attested in the literature, as is illustrated in the above Example (3).

We believe that this phenomenon-oriented perspective sheds some interesting light on questions related to defining and evaluating an appropriate source for generation, which is a notorious problem in state-of-the-art corpus-based NLG (see Section 6.1). While it is often difficult to make sense of the BLEU score in terms of how well it reflects the accuracy of a candidate selection model in terms of particular choice points, it is instructive to evaluate how a generator deals with particular paraphrases. In Chapter 7.4, we discuss some evaluation methods in more detail, and also present some human judgements collected for the data used in this experiment.

6.3 Experiment 5: Dependency-based Classification of Alternations

The experiment in the previous Section 6.2 modeled the prediction of voice alternations in a grammar-based surface realization ranking scenario where the representation adopted as a source for the generator and the information it makes available in its input is directly linked to the number of candidate realizations that the ranking component has to deal with. The experiment presented in the current Section deals with the prediction of verb alternations in purely statistical, dependency-based generation framework, on our data set of German robbery articles, described in Chapter 5.4. The texts in this corpus have been annotated for mentions of referents, including implicit referents, a shallow dependency representation, corresponding to parses produced by a probabilistic dependency parser, and a deep dependency representation, that abstracts from surface syntax as it removes information about e.g. the realization of auxiliaries and syntactic functions.

In order to analyze the prediction of verb alternations in a dependency-based setting, we simplify the problem of mapping deep to surface syntactic dependencies and cast it as a classification task: For each verb node in the data set, a statistical classifier has to predict the labels "active", "passive" or "nominalization". In our complete generation pipeline implemented for combined surface realization and REG (see Chapter 7.2), the task is modeled in a more complex architecture. The main goal of this experiment is to assess the effect of the annotation of implicit arguments, on the prediction of voice alternations and nominalizations in a purely statistical setting, comparing it to the grammar-based surface realization scenario evaluated above in Section 6.2. We will train and test the verb classifier on different versions of the deep syntactic dependencies from our robbery data set: a version that includes annotation of implicit referents and a version that simply includes referents as they were realized in the original corpus.

6.3.1 Comparison with Grammar-based Realization

First, we will summarize the most important differences between our grammar-based surface realization experiment from Section 6.2 and the dependency-based classification for verb alternations. These differences do not only concern the underlying generator but also the representation of the syntactic

input, such that we expect that the effects of underspecification in the input will be manifested differently in the two generation scenarios.

Syntactic Annotation Chapter 5.4.3 gives a detailed overview of the shallow and deep syntactic annotation of our data set. A big difference to the LFG-based analyses is that the purely statistical dependency parser used for obtaining the basic shallow annotations is not restricted by any hard constraints that concern the analysis of the argument frame of a verb. It does not have to assign a subject relation to a verb, or can even assign several subjects or objects to a verb. By contrast, in an LFG analysis, the grammar always has to find a subcategorization frame taken from the lexicon of the grammar that is satisfied by actual arguments in the sentence.

Transitivity In the previous Section 6.2, we constructed the data set in a way that we only included sentences with a transitive verb given in the lexicon of the LFG grammar. In the robbery data set, however, the types of verbs are not restricted and we are considering all v erbal nodes in the dependency annotation for generation. Note that this property of the data interacts with the previously described automatic dependency annotation: If an instance of an active verb in the corpus has no object dependent, this could be due to a parsing error or the fact that the verb is intransitive. Without any underlying grammatical knowledge, these two cases cannot be kept apart. For this reason, we cannot reasonably restrict our verb alternation classification to verbs that actually do have two arguments.

Classification vs. Grammar-based Realization In surface realization ranking, the input for the statistical module that models choice is a set of competing output c andidate sentences. During training, the ranker has to optimize its parameters in a way such that the original surface and gold output sentence always receives the best score given the feature model. The exact number of candidates for each training item and its surface forms depends on the length and structure of the original corpus sentence and the underlying grammar. Thus, for short sentences, the ranker typically has to deal with a smaller number candidates as compared to long sentences. The classification approach adopted for the current experiment is computationally less expensive. Instead of modeling choice between full output sentences, we model the choice of alternation separately for each verb. Moreover, instead of

predicting actual surface realizations, we predict very coarse-grained labels (active, passive, nominalization) that would correspond to specific grammatical patterns for surface realization. Thus, for each training item, the number of possible output labels is the same (and much smaller as compared to the ranking) and does not depend on the original corpus context.

Another perspective on the difference between these two approaches is that the grammar-based surface realizer has a hard-constrained grammatical "pre-filter" that only produces certain candidates if it is really syntactically possible in the given context. Thus, if the verb does not have a semantic object in the input representation, a passive output realization can never be produced and the realization ranker will never see the pattern in the data. By contrast, the alternation classifier can, in principle, predict all possible labels regardless of the verb's argument frame. If the argument frame is encoded as a feature for the classifier and if it often observes a certain label with certain argument frame features, it will learn that the argument frame is a strong predictor for the output alternation label.

6.3.2 Features and Set-up

We design several feature templates for the classifier that describe different syntactic and contextual properties of a verb node. Due to the reasons explained above, we also have to encode very basic features concerning the presence of arguments, the lemma and the label of the verb. The working hypothesis for the experiment is that these basic features are especially effective if the classifier is trained on an input where we exclude the implicit referents since we expect that the presence of certain arguments is then a strong predictor for the correct alternation. We expect that the other, contextual features are more important for the scenario where the classifier is trained on the annotation that includes implicit referents, as it should see the same argument frames with different labels more often.

The feature templates are specified as follows:

- **verb arguments**: boolean features for encoding whether verb has a semantic subject, object, or both
- **lemma**: lexical feature for verb lemma
- **label**: syntactic relation to the verb's head

- **children, grandchildren**: properties (label, PoS, lemma) of all the verb's children and grandchildren nodes

- **context**: boolean features for encoding whether verb lemma or verb arguments were previously mentioned in the sentence or text

We train two binary classifiers for predicting a) nominalized vs. non-nominalized verbs, and b) passive vs. non-passive verbs. In both cases, the majority baseline refers to relative frequency of non-nominalized and non-passive verbs. For evaluation, we will use the 10-fold split of the data set and report results from crossvalidation over the splits. Our evaluation measure is Accuracy, referring to the proportion of correctly predicted verb alternation labels.

6.3.3 Results

First, we look at the performance of the nominalization classifier, reported in Table 6.4 for the different feature templates. The "+Impl" column refers to the model trained on annotations that include implicit referents, "-Impl" excludes the implicit referents. The baseline classifier only integrates features for the argument frame of the verb. In the "+Impl" case, these argument frames include implicit arguments, whereas the "-Impl" input only represents the surface referents. Surprisingly, in both cases, the models do not improve over the majority baseline, meaning that neither the "-Impl" nor the "+Impl" model can trivially take advantage of arguments annotated in its input.

When adding two other basic features to the baseline, the verb lemma and its syntactic label, the classifiers clearly beat the majority baseline still with a very small difference between the two scenarios. Interestingly, the impact of the annotation of implicit referents is most visible when we add more features from the local and the wider context. Here, we obtain further improvements in both cases, but with a stronger positive tendency in the "-Impl" scenario.

Table 6.5 shows the performance of the passive classifier and its corresponding majority baseline which is lower than the nominalization baseline. Compared to the nominalization classifier, we observe clearer differences between the two annotation scenarios for the passive prediction. In the "+Impl" scenario, the baseline classifier is clearly below the majority baseline, whereas its equals the majority baseline in the "-Impl" case. When we add the lemma

Nominalization Classifier

	Accuracy		Majority BL
	+Impl	-Impl	
verb arguments	90.16	90.16	90.16
+ lemma	91.8	92.5	
+ label	94.15	94.37	
+ children, grandchildren	96.67	97.74	
+ context	96.54	97.77	

Table 6.4: Results for Experiment 5: Accuracy of the nominalization classifier, including implicit referents (+Impl) and excluding implicit referents (-Impl)

Passive Classifier

	Accuracy		Majority BL
	+Impl	-Impl	
verb arguments	85.94	88.68	88.68
+ lemma	88.96	88.71	
+ label	91.7	92.36	
+ children, grandchildren	91.89	93.81	
+ context	91.92	93.81	

Table 6.5: Results for Experiment 5: Accuracy of the classifier for passives on representation that includes implicit referents (+Impl) and excludes implicit referents (-Impl)

feature, both classifiers are on par with the majority baseline. Adding the label feature leads to an improvement for both classifiers such that they beat the baseline. Finally, the classifier further improves by adding contextual features, but only in the "-Impl" scenario.

These results suggest that the classifiers in the "-Impl" scenario cannot simply take advantage of the features that represent the arguments of the verb node, and consequently, that the classifiers in the "+Impl" scenario deal with more uncertainty due to implicit arguments. It is only when these features are combined with other properties of the context that we find certain advantages in terms of prediction accuracy for the "-Impl" and against the "+Impl" classifier. In particular, we showed that the argument features are only effective when they are combined with lexical and syntactic features for

	Total instances	2-role instances	
		+ Impl	-Impl
Active	2770	1174	957
Passive	383	170	79
Nominalization	333	143	5

Table 6.6: Experiment 5: proportion of active, passive and nominalized verb instances and their argument realization in the dependency-based generation inputs

the verb. This is basically the opposite of what we expected (also see the discussion in Section 6.3.1): we hypothesized that the contextual features would be more important for the "+Impl" models, whereas the "-Impl" more or less trivially achieve some satisfactory performance.

We interpret this effect as follows: as the set of verb is not restricted to transitive verbs and the classifier does not have access to hard grammatical constraints for voice generation, it is very important to model some basic lexical and syntactic knowledge so that the classifier is able to pick up certain patterns for the alternations in the data set. But even if when we take into consideration that the dependency-based classifier has to model certain grammatical constraints which the realization ranker does not have to model, it is still surprising that the differences between the "-Impl" and the "+Impl" scenario are not more pronounced. In order to get a better understanding of these effects, we analyze the distribution of verb arguments in our data set and count for all verb instances (active, passive and nominalized) whether they encode a semantic subject and object, see Table 6.6.

The distribution of verb instances and their arguments in Table 6.6 clearly shows why the argument features alone are not effective for classification: In the "-Impl" scenario, only a third of the verb instances in active voice actually have a subject and object dependent in the deep dependency representation. For passive and nominalization, these portions are even lower, but the fact that there are so few active verbs with two arguments prevents the feature from being a strong predictor (in isolation). In the "+Impl" scenario, we have very similar portions (between 42% and 44%) of 2-role instances for all 3 alternations types.

The fact that we have a relatively small portion of 2-role actives is due to the reasons we have discussed in Section 6.3.1: first, we are generalizing

over transitive and intransitive verbs. Second, the dependency parser used for obtaining the basic shallow dependency representation does not resolve any long distance dependencies related to control structures etc. Third, the dependency parser makes mistakes with respect to parsing predicate arguments. The reason for the fact that there is still a considerable number of passives and nominalizations with less than 2 roles even in the "+Impl" scenario is that we do not annotate all types of implicit referents in this data set, but only those that instantiate the roles of *perp*, *victim*, or *source*.

This analysis indicates that the deep dependency representation obtained from shallow, automatically predicted dependency analyses is too noisy to provide a good basis for an in depth study of linguistically motivated alternation modeling. However, we think that the comparison between the two referent annotations is still useful, as it provides a way to exactly test the argument representation in a variable setting. Thus, the finding that there are no substantial differences between the classifiers for the two referent annotations suggests that the classifier does not learn some basic syntactic distinctions, e.g. between transitives and intransitives. This also explains why the very basic features, the label and the lemma of the verb, are so important as they provide lexical and syntactic indicators for transitivity and e.g. control contexts where the verbs tend to have only one argument in the dependency representation.

6.4 Experiment 6: Implicit Referents in REG

Experiment 4 and 5 in the previous Sections have dealt with implicit referents in the context of syntactic alternations. In this Section, we model implicit referents in an REG framework, using the data set of German robbery articles described in Chapter 5.4. We will basically extend the results on REG from Experiment 3 in Chapter 4.4 where we exploited only the annotation of overt mentions of referents in the data set. While we found that the quality of the underlying syntactic input has a considerable impact on the contextual factors relevant for the REG task, we expect that implicit referents might have an even bigger impact on the underlying models for REG: When integrating annotations of implicit referents in an REG input, the surface slots of the reference chain are not completely specified and the system decides whether delete or to fill a certain slot with a referring expression. Consequently, some important sentence-external features, namely

the distance to the last mention and the category of the last mention, etc. are not determined by the input, but have to be modeled by the system. In parallel to the previous experimemts, the goal is to investigate the effect of an extended candidate choice, i.e. one that includes the empty realization, on the effectiveness of modeling techniques for REG.

6.4.1 Data and Set-up

In this experiment, we use the full RE annotation for our data set of German robbery articles. We compare the performance of the REG module on two different inputs:

- the standard input excluding annotations of implicit referents
- the extended input where implicit referents are included

In the first scenario, the reference chain that the RE model has to realize corresponds to the original reference chain in the corpus text. In the extended scenario, the model can change the overt realization of the chain by deleting certain slots from the surface.

In parallel to Experiment 3 in Chapter 4.4, the input to the REG module is a sequence of sentences represented as trees with underspecified REs and a list of candidate REs for each referent involved in the robbery event (a victim, a perpetrator, possibly a source). For each referent, the candidate list is specified as a list of dependency subtrees that have to inserted in the appropriate slots in the sentence trees. The candidates consists of all realizations of the referent in the original corpus text, plus a pronominal realization and a default definite description for each referent. For this experiment, each candidate list is extended with the empty subtree, which corresponds to deleting an RE from the surface sentence.

6.4.2 Implicit Referents and Linearized Input

The previously described REG experiment in Chapter 4.4 was based on the assumption that the RE module has access to a (gold or predicted) ordered dependency tree in its input. The fact that the input tree is ordered has an effect on two aspects of the model: First, the model realizes the REs in the order specified by the tree, thereby determining the context for the

consequent RE slots. Second, the feature model incorporates precedence information.

When the implicit referent annotations from the robbery data-set are added to the REG task, these assumptions about available linearization information have to be revised. In the data set, the implicit referents are annotated as properties of their syntactic heads. Thus, the annotation does not associate them with a unique surface syntactic position. While it would be possible to infer or manually annotate the corresponding position in certain contexts (such as implicit subjects in coordinated sentences, as they are annotated in Belz and Varges (2007)'s data set), but generally, the potential realization of an originally implicit argument can have several positions in the sentence, due to German free word order or variation in the order of modifiers of e.g. a nominalization.

As a consequence, the input for the extended REG task that includes implicit referents has to be defined as an unordered dependency tree. This tree can be used directly so that RE realizations are not predicted in an order corresponding to the linear surface order. Another option is to use automatically linearized trees, in parallel to the "predicted syntax" scenario in Chapter 4.4. However, the perfect syntax scenario that we investigated in Chapter 4.4, cannot be addressed in this setting since we simply do not have syntactic information about implicit referents. From this perspective, the extension of REG that includes implicit referents, does not simply extend the search space of the underlying model, but it affects the set-up of the task as a whole.

6.4.3 Results

The main goal of the experiment is to assess the effect of the extended implicit RE annotations on the accuracy of REG. In particular, we are interested in the way how the RE ranking module can deal with the contextual uncertainty introduced by the possibility to delete certain RE slots in the surface sentence. In the following, we will refer to the RE ranker that deals with explicit and implicit RE slots as the "extended model", whereas the ranker previously implemented in Chapter 4.4 is called the "standard model".

In parallel to the evaluation for Experiment 3 in Chapter 4.4, we use the following accuracies to evaluate the performance of RE realization, adding an accuracy measure that targets implicit referents:

	Exact	Type	Empty
Local slot + local ref	51.25	67.12	86.62
+ current sent ref	50.79	66.43	85.94
+ prev sent ref	49.65	66.66	85.48
+ hard context constraints	51.92	67.8	84.1
+ global ent	53.28	68.25	85.49

Table 6.7: Experiment 6: Feature ablation for extended REG (predicting implicit and explicit REs) on deep non-linearized dependency trees, results for the entire development split

1. Exact Accuracy, proportion of REs selected by the system with a string identical to the RE string in the original corpus, as in Belz et al. (2010)

2. Type Accuracy, proportion of REs selected by the system with an RE type identical to the RE type in the original corpus, as in Belz et al. (2010)

3. Empty Accuracy, proportion of REs predicted correctly as implicit/non-implicit

It is important to point out that the extended RE model is trained and tested on a version of the data that contains more RE slots per sentence and text such that the comparison with the accuracies obtained in Experiment 3 is not directly possible. For this reason, we also evaluate the extended model on a restricted annotation of the evaluation set that excludes the implicit referents.

In Table 6.7, we report the performance of the extended RE model on the deep, non-linearized input representations. The main observation is that the contextual features relating to the previously generated REs perform poorly and do not improve the purely local sentence-internal baseline. However, the hard constraints and the global entity features are effective.

Table 6.8 shows the performance of the RE ranker when it is trained and applied on shallow, automatically linearized input. In this case, the local, sentence-internal baseline performs worse when we compare its Exact accuracy to the Exact accuracy achieved by the local model trained on deep syntax. The Type and Empty accuracies of the two models are very similar. We observe a big difference as to the efficiency of context-based features in the two scenarios. Basically, the REG model trained on predicted syntax

	Exact Match	Type Match	Empty Match
Local slot + local ref	48.97	68.03	86.39
+ current sent ref	50.79	69.84	87.3
+ prev sent ref	52.15	70.29	87.98
+ hard context constraints	55.55	70.07	85.94
+ global ent	57.82	73.47	88.43

Table 6.8: Experiment 6: Feature ablation for extended REG (predicting implicit and explicit REs) on shallow predicted and linearized dependency trees, results for the entire development split

trees can achieve a much better overall performance due to the fact that the "current sent ref" and the "prev sent ref" features are informative.

In order to compare our extended REG model to the standard model, we carry out an evaluation on a version of the development set where all slots for implicit referents have been removed, i.e. this evaluation set is the same as the one used in Chapter 4.4. We also train the standard REG model on deep syntactic input with unordered dependencies. This comparison is shown in Table 6.9. The Empty Accuracy of the Standard Model is always 100% as it never predicts an empty RE slot.

Looking at the local baseline of the Standard Model, we observe that it is only slightly better than the local baseline of the Extended Model whereas the Type Accuracy clearly outperforms the Extended Model. This confirms an effect found in the previous experiments from Chapter 4.4: For a subset of the RE slots, the standard RE ranker can easily assign the exact referring expression used in the original corpus sentence. The drop in Type Accuracy of the Extended Model is clearly related to the fact that it makes roughly 8% false predictions for the explicit/implicit realization of an RE slot.

Most importantly, the feature ablation analysis in Table 6.9 shows that the relatively small amount of falsely predicted empty realizations by the Extended Model has a big effect on the way it can exploit its previous generation context: adding the context features decreases the Type Accuracy by 2% and improves the Exact Accuracy by 2%. By contrast, in the Standard Model, the context features are still useful for improving the Exact Accuracy by 4% at a constant Type Accuracy.

Finally, Table 6.10 presents the ablation analysis that compares the standard and the extended model on the restricted development set when the

	Extended Model			Standard Model	
	Exact	Type	Empty	Exact	Type
Local slot + local ref	43.17	65.4	92.7	46.67	73.02
+ current sent ref	42.86	64.76	92.06	47.93	72.7
+ prev sent ref	43.17	66.98	93.33	47.3	72.7
+ hard context constraints	43.17	65.39	88.25	49.52	72.38
+ global ent	45.71	66.67	90.79	50.79	73.02

Table 6.9: Experiment 6: Feature ablation for REG on deep, non-linearized dependency trees, results for restricted development set (excluding implicit mentions)

	Extended Model			Standard Model	
	Exact	Type	Empty	Exact	Type
Local slot + local ref	39.68	66.35	92.06	45.08	74.6
+ current sent ref	41.59	68.25	92.69	51.11	79.37
+ prev sent ref	44.12	69.52	94.29	55.23	79.37
+ hard context constraints	47.36	67.18	88.85	57.14	79.37
+ global ent	47.37	68.73	89.16	57.46	78.73

Table 6.10: Experiment 6: Feature ablation for REG,on shallow predicted and linearized dependency trees, results for restricted development set (excluding implicit mentions)

deep syntax input is automatically mapped to shallow syntax and also automatically linearized before REG. As we have shown on the full development set for the extended model, the RE ranker is more effective in this setting as the context features are more informative. However, we also find that there is an even more substantial performance difference in this setting between the Standard and the Extended Model. A first observation is that the Exact Accuracy of local feature baseline for the Extended Model drops by 4% as compared to the local feature baseline trained on the deep syntax. Thus, it seems that the Extended Model is more sensitive to mistakes made by the syntax prediction. The second observation is that the Standard Model can, overall, take more advantage of the context features, improving the Exact Accuracy by 12%.

As a summary, we have found that implicit referents massively increase the contextual uncertainty that an REG model has to deal with, although

the candidate sets as such are not massively extended, i.e. they basically add an empty candidate and some additional slots in the surface sentence. If the REG task is set up in a way that the coreference chain is not fully specified in the input, certain basic contextual dependencies are much harder to capture. Thus, in contrast to the alternation experiments in Section 6.2 and 6.3 where the performance of the candidate selection was not seriously impaired (looking at the broad measures for sentence quality), we now find very pronounced contrasts between the information available in the source and the accuracy of candidate selection. Moreover, this scenario sheds light on the close interactions between linear order and RE realization. The fact that accurate RE prediction seems to very dependent on the quality of the local syntactic context rises the question about how these two levels of choice should be dealt with in a full generation pipeline. This issue will be investigated in Chapter 7.2.

6.5 Conclusion

In the literature on corpus-based NLG, the variety of inputs that are used in state-of-the-art systems is often seen as problem for evaluation: the source for candidate generation directly or indirectly specifies the amount of variation that the statistical candidate selection model will encounter, such that even slight changes in the input can lead to remarkable differences in performance for the same system, which, in turn, makes it hard to draw meaningful conclusions from comparisons between systems.

Our experiments reported in this Chapter have basically exploited controlled and meaningful manipulations of generation inputs as a tool for evaluating to what extent a particular candidate selection model is dependent on the information available in the source, and to what extent it captures particular choice phenomena.

The surface realization experiment in Section 6.2 has shown that if a generation input does not provide an appropriate abstraction from the original realization of corpus sentences, e.g. by including implicit arguments, certain alternations can simply not be modelled as the underlying grammar-based generator cannot produce reasonable candidate sets for learning the contextual preferences.

However, this rationale does not carry over to the generation scenario from Section 6.3 where implicit arguments have a different status since the

purely statistical system does not implement hard syntactic constraints and does not have a deep representation of argument frames. In this experiment, the comparison between a deep and a more shallow input representation has been useful for establishing that both sources are rather noisy and, in the current state, not suited for an in-depth, linguistically informed investigation of contextual choice.

Finally, the same annotation decision leads to yet another result for the set-up and the performance of an REG system. Thus, the treatment of implicit arguments determines that the system cannot be trained on input that represents linearization information from the original corpus sentence, which is, however, standard practice in e.g. the GREC shared tasks (Belz et al., 2008, 2009; Belz and Kow, 2010). Moreover, similar to the REG experiment in Chapter 4.4, we have shown in Section 6.4 that certain types of contextual factors have different effects, depending on the underlying source.

In the light of these observations, we conclude that evaluations and comparisons of different sources in corpus-based NLG can be meaningful and instructive, if the corresponding results can be traced back to particular decisions of the generation system. Even more so, we argue that such evaluations are necessary for establishing whether a certain generation input allows for reasonable models capturing a particular range of well-defined choices.

Chapter 7

Modeling Interactions in Architectures for Combined Choice

Generating well-formed linguistic utterances from an abstract non-linguistic input involves making a multitude of conceptual, discourse-level as well as sentence-level, lexical and syntactic decisions. Work on rule-based natural language generation (NLG) has explored a number of ways to combine these decisions in an architecture, ranging from integrated systems where all choices are taken jointly (Appelt, 1982) to sequential pipelines (Reiter and Dale, 1997) where choices are modeled in well-delimited, separated modules. The main problem that underlies these architectural decisions is that the joint modeling or combination of all possible choices typically leads to an explosion of the search space whereas the separate modeling fails to capture interactions between different levels of choice.

The multitude of corpus-based generation approaches which address surface realization as an isolated task basically assume an underlying pipeline architecture, where sentence-level choice is modeled independently of higher-level generation decisions. However, the work presented in the preceding chapters of this thesis demonstrates that the interactions between syntactic and referential choice are deeply reflected in state-of-the art context models (Chapter 4) and generation inputs (Chapter 5 and 6), suggesting that the separation of surface realization from the rest of the NLG pipeline is artificial to a certain extent and that the results obtained in these isolated settings might not carry over to more abstract generation sources.

The set-up that we suggest in the current Chapter can be seen as a middle ground between full data-to-text generation where a whole range of complex interactions have to be captured at the same time and single-task generation where certain linguistic phenomena can be carefully, but also somewhat artificially, modeled. As a basis for this set-up, we use our data sets from Chapter 5 to study a broader range of choices than in standard surface realization, while keeping control on the growth of the search space resulting from a combination of choices, such that interactions between particular choice phenomena can be assessed and modeled.

Interactions and Frameworks In the preceding Chapters of this thesis, we have seen that the choice of generation framework – such as a grammar-based ranking architecture or a shallow statistical realizer – has far reaching consequences for the underlying implementation of context factors, derivation and annotation of input representations, and consequences for automatic evaluation. This thematic thread will be continued in this Chapter, from the perspective of modeling interactions between syntactic and referential choice. The extended systems and representations for surface realization and REG that we have developed and evaluated in Chapter 5 and 6 are also very different in terms of their architecture: the LFG-based surface realization ranker is essentially an integrated architecture where all choices (word order and alternations, in our case) are predicted jointly on the sentence-level, whereas our dependency-based combined system is a modular architecture where specific choices/parts of a sentence are modeled in designated sub-components. Consequently, these two system provide different means of accounting for interactions between choices.

Section 7.1 provides a general discussion of complex NLG tasks where several choices have to be modeled in combination. Section 7.2 presents an experiment on the robbery data set described, integrating the realization of referring expressions and the prediction of syntactic alternations. This shows that interactions between choices can be accounted for in a flexible architecture where modules interact via a revision mechanism. The experiment described in Section 7.3 expands on our experiments with the extended LFG-based surface realization where word order is combined with voice alternations. This shows that decisions in feature design and labeling at the level of realization ranking have to account for interactions between voice and word order. Section 7.4 discusses issues related to evaluation of com-

bined NLG tasks more generally, and presents results from a pilot human evaluation.

7.1 Architecture and Choice

In generation tasks where choice is not limited to a particular phenomenon like word order, the architectural design of the NLG system defines the way a complex choice process is computationally organized and modeled, e.g. through the the decomposition of the process into modules or components which reflect the relations and interactions between choices. This organization is a complex matter, as various types of knowledge and linguistic constraints on different levels have to be handled (De Smedt et al., 1996): Discourse planning is subject to pragmatic constraints, lexical choice requires semantic knowledge, syntactic sentence planning has to obey grammatical constraints, etc. An intuitively appealing way to manage the various knowledge resources is a modular architecture where choices requiring different types of knowledge are assigned to separate components and stages of the generation process, leading to a systematic and transparent model of the underlying linguistic constraints. The modular organization of language generation processes is also endorsed by some classical theories in psycholinguistic research (Fodor, 1983; Levelt, 1989).

Although the modular organization of linguistic constraints accounting for different types of choice being is clearly conceptually and intuitively appealing, it has been challenged since the early days of NLG, and it can be considered as one of the main threads of the research carried out in the field. An important insight from implementations of complex NLG systems is that the high-level organization of discourse is not independent from lexical and syntactic knowledge (Danlos, 1984; Hovy, 1990; Scott and de Souza, 1990; Mellish et al., 2000). In the following, we discuss a range of examples that have been described in the literature, in order to clarify the complexities of the problem and its implications for the set-up of an NLG system.

Example (1) illustrates a case discussed by Scott and de Souza (1990), where a discourse relation (EVIDENCE) between two propositions ('My car is not British', 'My car is a Renault') is supposed to be realized in a complex sentence:

(1) a. Since my car is a Renault, it is not British.

b. My car is a Renault, therefore it is not British.
c. *Therefore my car is not British, it is a Renault.
d. *Since my car is not British, it is a Renault.

In order to generate a coherent and meaningful sentence for the evidence relation between the two facts in (1), specific lexical constraints exhibited by the connectives 'since' and 'therefore', and the linear order of the sentence embeddings have to be modeled. Similar examples can be found in other works on sentence planning: Stent et al. (2004) defines a range of sentence aggregation operations that require syntactic and lexical knowledge about the involved clauses, e.g. in order to merge the two propositions in (2-a) to form a sentence like (2-b), MERGE is defined such that it applies to two clauses with identical matrix verbs and all but one identical arguments, presupposing lexical and syntactic structure of the involved clauses:

(2) a. MERGE(Above has good service; Carmine's has good service)
b. Above and Carmine's have good service.

But even for the construction of simple sentences, it is clear that syntactic processes cannot be treated independently from lexical knowledge. Rubinoff (1992) mentions Example (3) where the sentence planning component needs to have access to the lexical knowledge that "order" and not "home" can be realized as a verb in English.

(3) a. *John homed him with an order.
b. John ordered him home.

The relation between parts of an abstract representation and constituents in a linguistic utterance need not be univeral for all languages. Stede (1996) demonstrates that lexicalization patterns can vary across languages which has immediate implications for the design of generators, for instance in a multi-lingual translation setting. A well-known type of pattern relates to lexical incorporation phenomena, which have been investigated in the seminal work by Talmy (1985). He shows that English motion verbs tend to express MANNER, whereas motion verbs in Romance languages prefer to encode PATH. For instance, the English lexical concept for "swim across" would be translated to French by means of the verb "traverse" (*to cross*) and a gerund for "nager" (*swim*):

(4) a. He swam across the river.

b. Il a traversé la fleuve en nageant.

But even when looking at a single language, lexicalization and syntactic constraints can interact and vary depending on the context. Elhadad et al. (1997) illustrate what they call "floating constraints" by the following Examples which show different ways of linguistic realizing a description of an event:

(5) a. Wall Street Indexes opened strongly. *(time in verb, manner as adverb)*
b. Stock indexes surged at the start of the trading day. *(time as PP, manner in verb)*

Although most of these examples show interactions between conceptualization and lexicalization, the use of syntactic constructions is similarly concerned. Danlos (1984) shows that the passive construction for *kill* in (6) is related to the order of sentences in the discourse.

(6) a. Mary was killed by John. She was shot.
b. *Mary was shot. She was killed by John.

Generally, the problem of modularity has been most intensively discussed in the context of microplanning, which can be seen as the critical NLG component where an abstract level of knowledge representation needs to be connected to linguistic structures (Stone et al., 2003). According to Stone et al. (2003), microplanning subsumes the tasks of referring expression generation, lexical choice, and sentence aggregation. The linguistic choices involved relate to quite different levels of linguistic description and their combination leads to an intricate interplay of constraints between the separate levels. Following Appelt (1982), the problem of sentence planning has often been phrased as a constraint satisfaction problem: given some elementary linguistic items and their constraints, find a combination that satisfies the individual requirements of the elementary linguistic items and expresses exactly the semantic content that fulfills the communicative goal (Hovy, 1990; Stone et al., 2003; Koller and Stone, 2007; Banik, 2009).

A nice example that illustrates interactions between referring expression generation and sentence planning is reported by Banik (2009) who provides an in-depth study of parenthetical constructions. In a corpus study, she established that many parentheticals in free text follow the pattern in (7-a),

where the parenthetical needs to realize a pronominal subject, directly following the constituent that co-refers with the subject:

(7) a. Elixir, since it contains Gestodene, is banned by the FDA.
b. *The FDA, since Gestodene is an ingredient of Elixir, bans Elixir.
c. *Elixir, since Elixir contains Gestodene, is banned by the FDA.
d. *It, since Elixir contains Gestodene, is banned by the FDA.
e. *The FDA, since it contains Gestodene, banned Elixir.

In order to capture the wellformedness of (7-a), in contrast to the range of inappropriate realizations in (7-b-e), Banik (2009) encodes the various syntactic, and referential constraints in an integrated discourse grammar. Similar cases showing the interactions between referential and syntactic realization have been modeled in a Centering-based generation framework by Kibble and Power (2004). Also Nakatsu and White (2010) propose an integrated discourse grammar in the framework of CCG for generating multi-sentence paraphrases, being aware of constraints stemming from discourse connectives.

In line with our claims made in the Introduction in Chapter 1, most of the above approaches, concerned with generating from more abstract representations and structuring linguistic sentences, mainly target interactions between choices in terms of constraints and wellformedness. The aim of the integrated approaches to microplanning is to rule out text and sentence realizations that do not express an input in a way that it can be understood by a reader. Since the underlying dependencies between syntactic, lexical and referential choice can be so complex, less emphasis is put on distinguishing between several well-formed candidates in a particular communication context.

A slightly different perspective is suggested by Marciniak and Strube (2005), who address a complex generation tasks as a set of classification problems. They use a global optimization technique to find a good combination of the single classifiers and show that this architecture outperforms a sequential pipeline where modules are ordered such that the discourse-level choices are modeled prior to sentence-level choices. Figure 7.1 shows an example from their paper, where separate choices (verb form, connectives, sentence structure) can be combined in different ways.

While Marciniak and Strube (2005)'s approach does not call into question the modular design of an NLG system, it addresses the dimension of organization of modules and challenges the common organization principle of the

Discourse Unit	T_3	T_4	T_5
Pass the First Union Bank ...	null	vp	bare inf.
It is necessary that you pass ...	null	np+vp	bare inf.
Passing the First Union Bank ...	null	vp	gerund
After passing ...	after	vp	gerund
After your passing ...	after	np+vp	gerund
As you pass ...	as	np+vp	fin. pres.
Until you pass ...	until	np+vp	fin. pres.
Until passing ...	until	vp	gerund

Figure 7.1: Example sentences from a multi-level generation task due to Marciniak and Strube (2005) where several choices are combined: connective (T_3), sentence expansion (T_4), verb form (T_5)

pipeline where modules are arranged sequentially. Indeed, the pipeline seems to be the prevailing architecture in practical NLG applications (De Smedt et al., 1996; Cahill et al., 1999), since integrated systems typically face performance issues and do not easily scale to more than a few example inputs, see e.g. Koller and Petrick (2011) for some recent experiments that construe NLG as a planning task.

A typical problem that NLG pipelines face and that is nicely shown by Marciniak and Strube (2005)'s experiments are error propagation effects: if modules are set up to make generation decision in a strictly sequential fashion, they cannot correct or compensate for errors made in early modules, such that the accuracy necessarily decreases when descending the pipeline. Moreover, in rule-based sequential generators with a strict separation of modules and a unidirectional information flow, interactions between choices can lead to a so-called *generation gap*, where a down-stream module cannot realize a text or sentence plan generated by the preceding modules (Meteer, 1991; Wanner, 1994).

Of course, the modular set-up of a rule-based generation system does not necessarily imply that modules need to be arranged in a pipeline. As the two most successful systems that deviate from the pipeline, De Smedt et al. (1996) mention Hovy (1990)'s generator PAULINE, adopting an interleaved architecture, and a so-called revision-based architecture put forward by Robin (1993). He develops a system for describing sports events, where basic facts are generated in a first pass, and revised to include background information

in a second pass. Similar to the integrated system mentioned above, both Hovy (1990)'s and Robin (1993)'s approach seem to be quite domain specific.

Some meta-studies have tried to categorize existing applied generation systems and distill some general insights from a cross-system overview. Cahill et al. (1999) compare a set of 19 practical NLG systems, and find that most systems actually follow the standard pipeline layout of "content determination → sentence planning → surface realization". However, they also investigate whether the modular organization also corresponds to similar functional divisions, i.e. whether similar modules deal with similar choice phenomena. Here, they discover substantial discrepancies for a number of phenomena. For instance, the treatment of referring expressions is more or less equally split between the three stages among the systems. This suggests that the pipeline architecture does not provide a natural and principled solution when modeling combined choices in generation, which, in practice, seems to lead to highly idiosyncratic solutions that depend to a great extent on the application background.

Recently, a range of corpus-based methods applied on some classical domains, such as weather forecast generation, have been proposed that alleviate the need to decompose a complex data-to-text generation task into separate generation modules. For instance, Angeli et al. (2010) breaks down the generation process into a sequence of local, incremental decisions computed by log-linear classifiers. Each decision can be conditioned on the entire generation history, integrating features from content selection, lexical choice and template selection. On the other hand, Bohnet et al. (2011) deal with multilevel generation in a broad-coverage domain where templates for syntactic realization are too simple. They adopt a standard sequential pipeline approach.

This overview of NLG research that looks at systems architectures amply demonstrates that the treatment of a set of combined choices in a complex generation task is far from being a settled question. The general problem of microplanning, where various constraints coming from different sources of choice have to be handled in combination, is so complex that rule-based or corpus-based methods adopting principled solutions for dealing with interactions between choices immediately face efficiency and scalability problems, and have been developed for small domains and specific applications. This state-of-the art provides the underlying methodological motivation for looking at generation set-ups that cut down the space of possible choices in a reasonable way, by, at the same time, going beyond a specific domain and

adopting a broad-coverage approach to the corpus-based analysis of combined choice models. Thus, we argue that the generation experiments presented in this Chapter, which are devoted to more restricted types of interaction, i.e. interaction between syntactic and referential choice, are representative of a challenging and far reaching problem for any type of complex generation task, but allow for a more transparent treatment of interactions between choices.

7.2 Experiment 7: Flexible Pipelines for Multi-Level Generation

The experiments presented in the following bring together the two tasks of surface realization and REG. The goal is to study interactions between syntactic and referential choices in a broad-coverage corpus-based generation scenario. The experiment is based on the robbery data set described in Chapter 5.4. The input to the generator is a text which is defined as an ordered list of sentences, and lists of RE candidates for each *victim*, *perp*, or *source* referent that occurs in the text. Each sentence is defined as an unordered deep dependency tree with underspecified RE slots, including the annotations of implicit referents.

To illustrate the complexity of the task, we go back to the example input in Figure 2.11 in Chapter 2.4.2, which embeds two propositions 'the perpetrator entity is on trial', 'the perpetrator entity attacked the victim entity'. Example (8) (which is translated, for convenience) shows some possible surface realization combining different decisions on the level of REG, syntactic alternations and linearization:

(8)
- a. ???[They]$_p$ are on trial because of an attack by [two italians]$_p$ on [a young man]$_v$.
- b. Because [two italians]$_p$ attacked a young man, [they]$_p$ are on trial.
- c. ?Because of an attack by [two italians]$_p$ on a young man, [they]$_p$ are on trial.
- d. ?[The two men]$_p$ are on trial because [a young man]$_v$ was attacked by [them]$_p$.
- e. [The two men]$_p$ are on trial because [they]$_p$ attacked [a young man]$_v$.
- f. ??Because a young man was attacked, [two italians]$_p$ are on trial.

Note that due to the fact that we are dealing with a complex sentence, Example (8) is fairly reminiscent of some cases discussed for microplanning in Section 7.1. For instance, (8-a) is basically incoherent, as the the reader cannot establish that the subject of the main proposition is coreferent with the subject of the embedded clause. If the linear order of clauses is switched as in (8-b), the sentence is coherent, but as is shown by (8-c), the realization of the verb has to be modified as well.

The integrated modeling of REG and surface realization leads to a considerable expansion of the choice space as compared to a generation task that addresses this subproblems in isolation. Therefore, we assume that the basic generation framework is decomposed into the following three modules:

- SYN, a module that maps deep to shallow dependencies
- REG, a module that inserts RE subtrees into the tree of a sentence and deletes RE slots that are predicted to be implicit
- LIN, a module that linearizes the unordered dependency tree

The components are implemented in a way that they can be arranged in different architectures and orders. This means that the components have to be trained and applied on varying inputs, depending on the pipeline. In the following, we describe the basic set-up of our components and sketch the architectures that correspond to different arrangements of the single components.

7.2.1 Architectures

Our main goal is to investigate different architectures for combined surface realization and referring expression generation. Our flexible architectural setting is based on the fact that the three generation modules are trainable and basically expect an input in a general dependency format with no specific requirements on the exact specification of the dependencies (see Section 7.2.2). Consequently, each of the modules can be trained on slightly different representations in the multi-layer annotation of the data set. Below we define the different annotation layers in our data set that we will be used for describing the set-up of the architectures below.

- $deepSyn_{-re}$: deep dependencies with RE slots

- $deepSyn_{+re}$: deep dependencies with specified REs
- $deepSyn_{+impl}$: deep dependencies with deleted slots for implicit REs
- $shallowSyn_{-re}$: shallow dependencies with RE slots
- $shallowSyn_{+re}$: shallow dependencies with specified REs
- $shallowSyn_{+impl}$: shallow dependencies with deleted slots for implicit REs
- $linSyn_{+re}$: linearized shallow dependencies with specified REs

Note that the nodes in the deep and shallow dependency trees correspond to lemmas. The output of the NLG system will not be inflected. We experimented with the morphology component described by Bohnet et al. (2010), but could not get satisfactory results. Therefore, we decided not to treat morphology generation in order to ease evaluation, and exclude effects that would be introduced from an incorrect morphology. See Section 7.2.5 for some discussion.

In the following, we define the architectures that we will evaluate on our data set. For each architecture, we define the training and the testing step.

First Pipeline The first pipeline corresponds most closely to a standard generation pipeline in the sense of (Reiter and Dale, 1997). REG is carried out prior to surface realization, i.e. it is trained on pairs of deep dependency trees with RE slots and specified REs. The SYN component is trained on pairs of deep and shallow dependency trees with specified REs. Thus, the RE component does not have access to surface syntax or word order whereas the SYN component has access to fully specified referring NPs.

- training
  1. train REG: $(deepSyn_{-re}, deepSyn_{+re})$
  2. train SYN: $(deepSyn_{+re}, shallowSyn_{+re})$
- prediction
  1. apply REG: $deepSyn_{-re} \rightarrow deepSyn_{+re}$
  2. apply SYN: $deepSyn_{+re} \rightarrow shallowSyn_{+re}$
  3. linearize: $shallowSyn_{+re} \rightarrow linSyn_{+re}$

Second Pipeline In the second pipeline, the order of the REG and SYN component is switched. In this case, REG has access to surface syntax without word order but the surface realization is trained and applied on trees with underspecified RE slots.

- training
  1. train SYN: $(deepSyn_{-re}, shallowSyn_{-re})$
  2. train REG: $(shallowSyn_{-re}, shallowSyn_{+re})$
- prediction
  1. apply SYN: $deepSyn_{-re} \rightarrow shallowSyn_{-re}$
  2. apply REG: $shallowSyn_{-re} \rightarrow shallowSyn_{+re}$
  3. linearize: $shallowSyn_{+re} \rightarrow linSyn_{+re}$

Third Pipeline While the first and second pipeline model the prediction of implicit referents in the general REG module, the third pipeline separates the prediction of implicit referents in a single module. In this case, the SYN module will have access to slightly different deep dependency representation as edges for implicit RE slots are removed. The REG module has to predict appropriate explicit REs having access to the surface syntax. Thus, the empty RE is removed from the candidate sets for all RE slots.

- training
  1. train IMPL: $(deepSyn_{-re}, deepSyn_{+impl})$
  2. train SYN: $(deepSyn_{+impl}, shallowSyn_{+impl})$
  3. train REG: $(shallowSyn_{+impl}, shallowSyn_{+re})$
- prediction
  1. apply IMPL: $(deepSyn_{-re}, deepSyn_{+impl})$
  2. apply SYN: $deepSyn_{+impl} \rightarrow shallowSyn_{+impl}$
  3. apply REG: $shallowSyn_{+impl} \rightarrow shallowSyn_{+re}$
  4. linearize: $shallowSyn_{+re} \rightarrow linSyn_{+re}$

In this statistical multi-stage set-up, we use the so-called jackknifing technique for training the downstream modules of a pipeline. This technique accounts for the problem that, during testing, a downstream module of pipeline receives input that has potential prediction errors from the previously applied modules. If we were training the downstream modules on gold inputs, the systems would see high quality input during training and, potentially, low quality input during testing. In order to adapt the modules to predicted input, we create jackknifed or cross-annotated versions for the different layers of the data set. For instance, for the first pipeline, we train and apply the REG module on all 10 folds of the robbery articles. The SYN module is trained on pairs of deep dependency trees with predicted REs and shallow dependencies with predicted REs. In the second pipeline, the REG module is trained on pairs of predicted shallow trees with RE slots and with specified REs.

7.2.2 Modules

In the following, we describe the implementation of the syntactic component (SYN), the REG component, and the linearizer (LIN).

SYN: Deep to Shallow Syntax

The SYN component models the mapping from deep to shallow dependency trees. It is implemented as a ranker that learns transformations between nodes in the deep dependency tree and patterns of dependency relations in the shallow dependency tree. Basically, this ranker learns to revert the transformations (defined in Section 5.4.3) which map shallow dependency trees obtained from the parser to deep syntactic relations. The component extends the alternation classifier from Experiment 5 (Chapter 6.3), as it predicts an actual surface syntactic structure for a deep dependency node, instead of a label standing for a particular alternation.

The general problem of learning transformations between aligned trees requires relatively complex formalisms, depending on the type of transformations found in the data. In our case, the transformations are restricted to single verb nodes in the deep tree (possessives are handled in the RE module) that can be aligned to a set of nodes in the shallow dependency tree. This means that there are no dependencies between the transformations of two verb nodes in the same tree such that we can learn the transformation

patterns easily by looking at the alignments. However, it has to be noted that this straightforward approach would not carry over to some simple extensions of the transformation from Section 5.4.3. For instance, if we would treat coordinations in our syntactic transformations, the assumption that two verb nodes can always be mapped independently of each other would not hold anymore.

The learner is implemented as a ranking component, trained with SVMrank Joachims (2006). During training, each instance of a verb node has one optimal shallow dependency alignment and a set of distractor candidates. During testing, the module has to pick the best shallow candidate according to its feature model.

During training, the syntax generator first processes the alignment between the deep and the surface trees in the training set. For each verbal node in the deep tree, a corresponding verbal subtree in the surface tree is extracted. Given a pair of deep node and shallow subtree, we try to construct a generalized verb transformation rule where as much lexical information as possible is removed. The generalization step replaces information that is identical on both rule sides by a placeholder. These generalized rules allow for transformations of verb nodes when we have never seen a verb lemma in the training data. As an example, consider the rule in (9), that applies to every verb lemma and maps its underspecified PoS tag to a finite verb tag.

(9) (x,`lemma`,VV,y) → (x,`lemma`,VVFIN,y)

The rules have different degrees of lexicalization. The rule in (10-a) only applies to verbal nodes with the lemma *überfallen* (*rob*) and maps it to a nominalization. This mapping involves a transformation of a verbal to a nominal lemma such that the rules cannot be generalized. Similarly, we need lexicalized transformation for the mapping to finite realization of particle verbs, as in (10-b).

(10) a. (x,überfallen/*attack*,VV,y) →
(x,bei/*at*,PREP,y), (z,Überfall/*attack*,NN,x),(q,der/*the*,ART,z)
b. (x,abnehmen,VV,y) → (x,nehmen,VVFIN,y), (z,ab,PART,x)

When we split the data set into 10 folds, we extract, on average, 374 transformations from the training sets. This set subdivides into non-lexicalized and lexicalized transformations. Most transformation rules (335 out of 374 on average) are lexicalized for a specific verb lemma and mostly transform

nominalizations as in rule (10-b) and particles (see Section 5.4.3).

When we observe rules like (9) in the training set (which is very likely), we will be able to transform verbal nodes whose lemma we have not observed in the training set. Most of the nominalization rules will be lexicalized such that we cannot predict them for unknown verbs in the test set. After the first iteration through the training set, we unify all the rule right hand sides that have an identical left hand side. On the basis of the resulting verb transformation grammar, we obtain all possible candidate surface trees for each instance of a verbal node in the training set.

In all the pipelines we have experimented with, the input to the syntax generator is a non-linearized tree. The verbal nodes are traversed according to the order defined in Section 5.4.1. This order is deterministic: matrix verbs will always be processed before embedded verbs, heads of coordinations will be processed before coordinated verbs. In a more abstract syntax, this order would be less deterministic.

SYN baseline The baseline for the verb transformation component is a two-step procedure: 1) pick a lexicalized rule if available for that verb lemma, 2) pick the most frequent transformation. This baseline has the effect that, e.g. particle verbs will be mapped to their finite realization with a split node for the particle.

REG: Realizing Referring Expressions

Similar to the syntax component, the REG module is implemented as a ranker that selects surface RE subtrees for a given referential slot in a deep or shallow dependency tree. The candidates for the ranking correspond to the entire set of REs used for that referential role in the original text. The basic RE module is a joint model of all RE types, i.e. nominal, pronominal and empty realizations of the referent. In Chapter 4.4 and 6.4, we have already reported on various evaluations of this module considering REG as an isolated task.

For testing the third pipeline as defined above, we use an additional separate classifier for implicit referents, also trained with SVMrank. It uses the same feature model as the full ranking component, but learns a binary distinction for implicit or explicit mentions of a referent. The explicit mentions will be passed to the RE ranking component.

REG baseline The baseline for the REG component is defined as follows: if the preceding and the current RE slot are instances of the same referent, realize a pronoun, else realize the longest nominal RE candidate that has not been used in the preceding text. This corresponds to the intuition that the first mentions of a referent in a text will be relatively long nominal descriptions. For the subsequent mentions, the procedure predicts shorter NPs or pronouns.

LIN: Linearization

For linearization, we use the state-of-the-art dependency linearizer described in Bohnet et al. (2012). We train the linearizer on an automatically parsed version of the German TIGER treebank Brants et al. (2002). This version was produced with the dependency parser by Bohnet (2010), trained on the dependency conversion of TIGER by Seeker and Kuhn (2012).

For our generation experiments, we have investigated architectures that apply linearization to intermediate steps of the generation process. When we linearize trees that have underspecified RE slots, we apply a simple preprocessing to the trees: each slot is replaced with the default RE for the given role label.

Feature Models

Depending on the way the generation components are combined in an architecture, they will have access to different layers of the input representation. This means that the underlying feature models of the components also vary with the architecture.

The implementation of the feature models is based on a general set of templates for the SYN and REG component. The exact form of the models depends on the input layer of a component in a given architecture. For instance, when SYN is trained on $deepSyn_{-re}$, the properties of the children nodes are less specific for verbs that have RE slots as their dependents. When the SYN component is trained on $deepSyn_{+re}$, lemma and PoS of the children nodes are always specified.

The feature templates for SYN combine properties of the shallow candidate nodes (label, PoS and lemma for top node and its children) with the properties of the instance in the tree: (i) lemma, tense, (ii) sentence is a

header, (iii) label, PoS, lemma of mother node, children and grandchildren nodes (iv) number, lemmas of other verbs in the sentence.

The feature templates for REG combine properties of the candidate RE (PoS and lemma for top node and its children, length) with properties of the RE slot in the tree: lemma, PoS and labels for the (i) mother node, (ii) grandmother node, (iii) uncle and sibling nodes. Additionally, we implement a small set of global properties of a referent in a text: (i) identity is known, (ii) plural or singular referent, (iii) age is known, and a number of contextual properties capturing the previous referents and their predicted REs: (i) role and realization of the preceding referent, (ii) last mention of the current referent, (iii) realization of the referent in the header.

7.2.3 Baselines, Pipelines and Upper Bounds

For each input text, the generation system outputs a list of surface sentences. We will evaluate the system by comparing the original text in the corpus against the predicted output text on every sentence. We split our data set into 10 splits of 20 articles. We use one split as the development set, and crossvalidate on the remaining splits.

Evaluation Measures: We use a number of evaluation measures familiar from previous generation shared tasks:

1. BLEU, sentence-level geometric mean of 1- to 4-gram precision, as in (Belz et al., 2011)

2. NIST, sentence-level n-gram overlap weighted in favor of less frequent n-grams, as in (Belz et al., 2011)

3. RE Accuracy on String, proportion of REs selected by the system with a string identical to the RE string in the original corpus, as in (Belz et al., 2010)

4. RE Accuracy on Type, proportion of REs selected by the system with an RE type identical to the RE type in the original corpus, as in (Belz et al., 2010)

Second, we define a number of measures motivated by our specific set-up of the task:

1. BLEU_r, sentence-level BLEU computed on post-processed output where predicted referring expressions for *victim* and *perp* are replaced in the sentences (both gold and predicted) by their original role label, this score does not penalize lexical mismatches between corpus and system REs

2. RE Accuracy on Impl, proportion of REs predicted correctly as implicit/non-implicit

3. SYN Accuracy on String, proportion of shallow verb candidates selected by the system with a string identical to the verb string in the original corpus

4. SYN Accuracy on Type, proportion of shallow verb candidates selected by the system with a syntactic category identical to the category in the original corpus

The first evaluation addresses the performance of the "standard" generation pipelines, i.e. the first, second and third pipeline defined above. First, we will ask whether a certain sequential order of the modules is beneficial for the final output quality. Second, the evaluation is intended to assess to what extent the performance of the single modules affects the predictions by other modules. Therefore, we will compare the pipelines against some upper bounds where certain layers of the input representation are specified as in the original analysis of the corpus sentence. As a sanity check, we also compare against a baseline generation system that combines the baseline version of the SYN component and the REG component respectively.

As we report in Table 7.1, all pipelines largely outperform the baseline. Otherwise, they obtain very similar scores in all measures with a small, weakly significant tendency for the BLEU score of the third pipeline which separates the prediction of implicit and explicit referents. However, the BLEU_r score for the third pipeline is lower than for the two other pipelines which seems to be contradictory. This effect can be better understood by looking at the upper bounds.

The three bottom rows in Table 7.1 report the performance of the individual components and linearization when they are applied to inputs with an REG and SYN oracle, providing upper bounds for the pipelines applied on $deepSyn_{-re}$. When REG and linearization are applied on $shallowSyn_{-re}$ with gold shallow trees, the BLEU score is lower (60.57) as compared to the

		Sentence overlap		
Input	System	BLEU	NIST	$BLEU_r$
$deepSyn_{-re}$	Baseline	42.38	9.9	47.94
$deepSyn_{-re}$	1st pipeline	54.65	11.30	59.95
$deepSyn_{-re}$	2nd pipeline	54.28	11.25	59.62
$deepSyn_{-re}$	3rd pipeline	55.38	11.48	59.52
gold $deepSyn_{+re}$	SYN→LIN	63.9	12.7	62.86
gold $shallowSyn_{-re}$	REG→LIN	60.57	11.87	68.06
gold $shallowSyn_{+re}$	LIN	79.17	13.91	72.7

Table 7.1: Experiment 6: Generation performance achieved by the standard pipeline architectures on robbery articles including implicit referents; at the bottom, scores for upper bounds (i.e. generation inputs that predetermine REs, or syntactic realization, or both)

system that applies syntax and linearization on $deepSyn_{+re}$, deep trees with gold REs (BLEU score of 63.9). However, the $BLEU_r$ score, which generalizes over lexical RE mismatches, is higher for the REG→LIN components than for SYN→LIN. Moreover, the $BLEU_r$ score for the REG→LIN system comes close to the upper bound that applies linearization on $linSyn_{+re}$, gold shallow trees with gold REs ($BLEU_r$ of 72.4), whereas the difference in standard BLEU and NIST is high. This effect indicates that the RE prediction mostly decreases BLEU due to lexical mismatches, whereas the syntax prediction is more likely to have a negative impact on final linearization. This could also be an explanation for the contradictory results of the third pipeline. Because of prediction errors made by the IMPL module, the quality of the SYN prediction decreases, so that we observe a drop in the $BLEU_r$ score. On the other hand, the REG module has a slight benefit from the fact that the candidate set is reduced so that the BLEU score increases.

The error propagation effects that we find in the first and second pipeline architecture clearly indicate that the predictions made by the single modules affect each other, pointing to interactions between choices at the level of syntax, referring expressions and word order. Table 7.2 provides another analysis of these propagation effects and shows the accuracy measures achieved by the REG and SYN module depend on the pipeline. Generally, the accuracy of the individual components is, in each case, lower when they are applied as the second step in the pipeline. Thus, the RE accuracy suffers from mistakes

		SYN Accuracy		RE Accuracy		
Input	System	String	Type	String	Type	Impl
$deepSyn_{-re}$	Baseline	35.66	44.81	33.3	36.03	50.43
$deepSyn_{-re}$	1st pipeline	57.09	68.15	54.61	71.51	84.72
$deepSyn_{-re}$	2nd pipeline	59.14	68.58	52.24	68.2	82
gold $deepSyn_{+re}$	SYN→LIN	60.83	69.74	100	100	100
gold $shallowSyn_{-re}$	REG→LIN	100	100	60.53	75.86	88.86
gold $shallowSyn_{+re}$	LIN	100	100	100	100	100

Table 7.2: Experiment 6: Accuracies achieved by single modules in the standard pipeline architectures on robbery articles including implicit referents

from the predicted syntax in the same way that the quality of syntax suffers from predicted REs. In particular, the REG module seems to be affected more seriously from prediction errors made by the SYN module. The RE String Accuracy decreases from 60.53 on gold shallow trees to 52.24 on predicted shallow. trees whereas the Verb String Accuracy decreases from 60.83 on gold REs to 57.04 on predicted REs.

In Table 7.3, we report the performance of our generation pipelines when we apply them on a version of the input representations that exclude implicit referents. Experiment 6 in Chapter 6.4 has shown that the inclusion of implicit referents in the REG input has a considerable effect on the performance of the module. In Table 7.3, we see exactly that the exclusion of implicit referents has a parallel effect for the two pipelines, i.e. the BLEU scores increase from around 54 to 57.9. The final linearization also benefits from the simpler inputs, the $BLEU_r$ scores increases to 62.08. This $BLEU_r$ score almost reaches the score achieved by the upper bound that has gold RE input (SYN → LIN in Table 7.1.) This suggests that the SYN module mostly suffers from prediction mistakes that concern the implicit or explicit realization of an RE whereas the other dimensions of referring expression realization (e.g. pronoun vs. definite description) have a less pronounced effect.

7.2.4 Beyond the standard pipeline

In order to account for the error propagation effects found in each of our pipeline set-ups, we investigated two alternative organization modes for our

		Sentence overlap		
Input	System	BLEU	NIST	BLEU_r
$deepSyn_{-re}$	1st pipeline	57.93	11.73	62.08
$deepSyn_{-re}$	2nd pipeline	57.91	11.73	61.92

Table 7.3: Experiment 6: Generation performance achieved by the standard pipeline architectures onr tobbery articles excluding implicit referents

generation components:

Parallel System In our parallel system, the components are trained independently of each other, i.e. REG is trained on pairs of deep dependencies and SYN is trained on trees with RE slots. In the testing step, we first compute the transformations predicted by SYN and REG and apply them in parallel on the deep syntactic input with underspecified REs.

- training
  1. train SYN: $(deepSyn_{-re}, shallowSyn_{-re})$
  2. train REG: $(deepSyn_{-re}, deepSyn_{+re})$
- prediction
  1. apply REG and SYN:
     $deepSyn_{-re} \rightarrow shallowSyn_{+re}$
  2. linearize: $shallowSyn_{+re} \rightarrow linSyn_{+re}$

Revision-based System Our revision-based system is basically a pipeline system that first does the SYN step and then applies a preliminary linearization on shallow dependency trees with RE slots. As a consequence, the RE component has access to surface syntax and a preliminary linearization, called *prelinSyn*. This set-up adds two new annotation layers to our data set:

- $prelinSyn_{-re}$: pre-linearized shallow dependencies with RE slots
- $prelinSyn_{+re}$: pre-linearized shallow dependencies with specified REs

Input	System	BLEU	NIST	BLEU_r
$deepSyn_{-re}$	1st pipeline	54.65	11.30	59.95
$deepSyn_{-re}$	Parallel	54.78	11.30	60.05
$deepSyn_{-re}$	Revision	56.31	11.42	61.30

Table 7.4: Experiment 6: Comparing the generation performance of the best standard pipeline against a parallel and a revisions-based architecture

In this set-up, we first apply the linearizer on trees with underspecified RE slots. For this step, we insert the default REs for the referent into the respective slots. After REG, the tree is linearized once again. Training and testing are defined as follows:

- training
  1. train SYN on gold pairs of $(deepSyn_{-re}, shallowSyn_{-re})$
  2. train REG on gold pairs of $(prelinSyn_{-re}, prelinSyn_{+re})$
- prediction
  1. apply SYN: $deepSyn_{-re} \rightarrow shallowSyn_{-re}$
  2. linearize: $shallowSyn_{-re} \rightarrow prelinSyn_{-re}$
  3. apply REG: $prelinSyn_{-re} \rightarrow prelinSyn_{+re}$
  4. linearize: $prelinSyn_{+re} \rightarrow linSyn_{+re}$

In this evaluation, we investigate two methods for addressing the error propagation effects previously found with strictly sequential pipelines. We compare the first pipeline against the parallel and the revision-based architecture introduced in Section 7.2.4. The evaluation in Table 7.4 shows that the parallel architecture improves only marginally over the pipeline. By contrast, we obtain a clearly significant improvement for the revision-based architecture on all measures. The fact that this architecture significantly improves the BLEU, NIST and the BLEU_r score of the parallel system indicates that the REG benefits from the predicted syntax when it is approximatively linearized. The fact that also the BLEU_r score improves shows that a higher lexical quality of the REs leads to better final linearizations.

		RE Accuracy		
Input	System	String	Type	Impl
$deepSyn_{-re}$	RE	54.61	71.51	84.72
$deeplinSyn_{-re}$	RE	56.78	72.23	84.71
$prelinSyn_{-re}$	RE	58.81	74.34	86.37
gold $linSyn_{-re}$	RE	68.63	83.63	94.74

Table 7.5: Experiment 6: Accuracy of the RE module depending on its input, i.e. the previously applied generation modules

Table 7.5 shows the performance of the REG module on varying input layers, providing a more detailed analysis of the interaction between REG, syntax and word order. In order to produce the $deeplinSyn_{-re}$ layer, deep syntax trees with approximative linearizations, we preprocessed the deep trees by inserting a default surface transformation for the verb nodes. We compare this input for REG against the $prelinSyn_{-re}$ layer used in the revision-based architecture, and the $deepSyn_{-re}$ layer used in the pipeline and the parallel architecture. The REG module benefits from the linearization in the case of $deeplinSyn_{-re}$ and $prelinSyn_{-re}$, outperforming the component trained applied on the non-linearized deep syntax trees. However, the REG module applied on $prelinSyn_{-re}$, predicted shallow and linearized trees, clearly outperforms the module applied on $deeplinSyn_{-re}$. This shows that the RE prediction can actually benefit from the predicted shallow syntax, but only when the predicted trees are approximatively linearized. As an upper bound, we report the performance obtained on $linSyn_{-re}$, gold shallow trees with gold linearizations. This set-up corresponds to the GREC tasks. The gold syntax leads to a huge increase in performance.

These results strengthen the evidence from the previous experiment that decisions at the level of syntax, reference and word order are interleaved. A parallel architecture that simply "circumvents" error propagation effects by making decisions independent of each other is not optimal. Instead, the automatic prediction of shallow syntax can positively impact on RE generation if these shallow trees are additionally processed with an approximative linearization step.

(11) Generated by sequential system:

a. Deshalb gab [dem Täter] (seine) Brieftasche ohne daß
Therefore gave [to the robber] (his) wallet without that
(das Opfer) Widerstand leistet heraus.
(the victim) resistance shows out.

b. [Er] nahm anschließend (dem Opfer) die Armbanduhr ab und
[He] takes afterwards (the victim) the watch off and
[der Täter] flüchtete.
[the robber] fleed.

(12) Generated by revision-based system:

a. (Das Opfer) gibt deshalb (seine) Brieftasche ohne
(The victim) gave therefore (his) wallet without
Widerstand zu leisten heraus.
resistance to show out.

b. Anschließend nahm [der Täter] (dem Opfer) die Armbanduhr
Afterwards took [the robber] (the victim) the watch
ab und flüchtete.
off and fleed.

Figure 7.2: Two automatically generated output texts from Experiment 6, (see Figure 5.8 for the original sentences)

7.2.5 Discussion

The results presented in the preceding evaluations consistently show the tight connections between decisions at the level of reference, syntax and word order. These interactions entail highly interdependent modeling steps: Although there is a direct error propagation effect from predicted verb transformation on RE accuracy, predicted syntax still leads to informative intermediate linearizations that improve the RE prediction. Our optimal generation architecture thus has a sequential set-up, where the first linearization step can be seen as an intermediate feedback that is revised in the final linearization. This connects to the revision-based set-up proposed by (Robin, 1993).

In Figure 7.2, we compare two system outputs for the last two sentences of the text in Figure 5.8.[1] The output of the sequential system is severely incoherent and would probably be rejected by a human reader: In sentence (5a) the *victim* subject of an active verb is deleted, and the relation between the possessive and the embedded *victim* RE is not clear. In sentence (5b) the first conjunct realizes a pronominal *perp* RE and the second conjunct a nominal *perp* RE. The output of the revision-based system reads much more natural. This example shows that the extension of the REG problem to texts with more than one main referent (as in the GREC data set) yields interesting inter-sentential interactions that affect textual coherence.

Concerning our evaluation methodology, we found that the distinction of BLEU and BLEU_r provides some differentiated insights as to the dimensions of the output quality. As these measures contradict each other for the third pipeline, we cannot reach a conclusion as to whether implicit referents should be modeled jointly with other REs or not. From a more general perspective, this problem indicates that our evaluation does not necessarily allow for definite conclusions about the actual fluency and coherence of the generation output. The issue will be discussed more thoroughly in Section 7.4 below.

[1] To make the output readable, we manually edited the text transforming lemmas to inflected words.

7.3 Experiment 8: Realization Ranking as an Integrated Model for Word Order and Voice

Due to the large number of combinations of choices in combined surface realization and REG, the previous experiment in Section 7.2 has adopted a modular generation approach where certain types of choices are treated in separate components. In the current Section, we come back to our grammar-based, extended surface realization architecture that we previously discussed in Chapter 6.2. This architecture can be considered as an integrated model where all possible combinations of voice and word order alternations are generated. The main goal of the analyses reported in the following is to assess the effects of interaction between voice and word order in the surface realization ranking model.

The empirical investigation of interaction between alternation phenomena embodies some interesting theoretical questions. As discussed in Chapter 3.1, the use of argument alternations has been related to markedness hierarchies in the linguistic literature (Aissen, 1999, 2003). These hierarchies associate argument functions with certain properties for person, animacy, or definiteness; e.g., subjects tend to be higher on the person scale than objects (1st person $<$ 3rd person). (Bresnan et al., 2001) find that there is a statistical tendency in English to passivise a verb if the patient is higher on the person scale than the agent. Bresnan et al. (2007) correlate the use of the English dative alternation to a number of features such as givenness, pronominalization, definiteness, constituent length, animacy of the involved verb arguments. These features are assumed to reflect the discourse accessibility of the arguments. Interestingly, the properties that have been used to model argument alternations in strict word order languages like English have been identified as factors that influence word order in free word order languages like German, see Chapter 3.1 for a number of pointers. Cahill and Riester (2009) implement a model for German word order variation that approximates the information status of constituents through morphological features like definiteness, pronominalization etc. We are not aware of any corpus-based generation studies investigating how these properties relate to argument alternations in free word order languages.

7.3.1 Data and Set-up

We use our set of transitive sentences extracted from the German newspaper corpus HGC which divides into 8044 training sentences and 1236 test sentences (see Experiment 4 in Chapter 6.2). For each of the sentences, we produce surface realization candidates, with the help of our extended surface realization architecture (see Chapter 5.3). While we experimented with different ways of generating surface realization candidates in Chapter 6.2, we work with a fixed meaning representation in the current experiment. In particular, we use our heuristic meaning representation (SEM_h) which captures implicit agents in passives and yields the largest number of average surface realization candidates (75.8 surface strings on average).

In Example (12), we show a candidate set from our data. The example illustrates that the candidates realize all possible combinations of word order and voice for a given sentence. The heuristic meaning representation for implicit agents captures the fact that the "in"-PP modifier in the passive (12-a) corresponds to the agent in the passive.

(13)
a. In der Fernsehsendung wurde keiner der Namen erwähnt.
   In the TV show was none of the names mentioned.
b. Die Fernsehsendung erwähnte keinen der Namen.
   The TV show mentioned none of the names.
c. Keinen der Namen erwähnte die Fernsehsendung.
   None of the names mentioned the TV show.
d. Keiner der Namen wurde in der Fernsehsendung
   None of the names was in the TV show
   erwähnt.
   mentioned.

In parallel to Experiment 4, we use the following evaluation measures:

- Exact Match: how often does the model select the original corpus sentence
- BLEU: n-gram overlap between top-ranked and original sentence
- Voice: how often does the model select a sentence from the original f-structure

- Precedence: how often does the model generate the right order of the verb arguments (agent and patient)

- Vorfeld (VF): how often does the model correctly predict the verb arguments to appear in the sentence initial position before the finite verb, the so-called *Vorfeld*.

7.3.2 Features and Labels for Ranking

The results reported in the following will provide an analysis of the feature model used in the integrated ranking for word order and voice. We divide the features from Chapter 6.2 into two main classes:

- Precedence: features that capture the relative linear order of two constituents, constituents are represented in terms of their morphosyntactic properties, e.g. "object precedes subject", "plural object precedes singular subject"

- Scale Alignment: combinations of voice and syntactic role of a constituent with its morphological properties (corresponding to the markedness hierarchies, see Chapter 3.1), e.g. "agent in passive is singular", "patient in passive is 1st person"

Note that, in the integrated surface realization ranking, we aggregate all precedence and scale alignment features of the constituents in a sentence in a single feature vector. Each feature is integer-valued, representing the count of the feature in the sentence.

For the current experiment, we also varied the labeling scheme of our ranking model, which tells the learner how closely each surface realization candidate resembles the original corpus sentence. Previously, we distinguished the ranks: "1" identical to the corpus string, "2" identical to the corpus string ignoring punctuation, "3" small edit distance (< 4) to the corpus string ignoring punctuation, "4" different from the corpus sentence. In the following, we introduce the additional rank "5" to explicitly label the surface realizations derived from the alternation f-structure, that does not correspond to the parse of the original corpus sentence. All surface realization candidates ranked "5" thus contain a verb which is realized in a voice that does not correspond to the original realization.

Features	Exact Match	BLEU	Voice	Prec.	VF
Precedence	16.3	0.70	88.43	64.1	59.1
Scale Alignment	10.4	0.64	90.37	58.9	56.3
Union	26.4	0.75	91.50	80.2	70.9

Table 7.6: Experiment 7: Feature ablation for the integrated surface realization ranker according to generation performance and voice, precedence and Vorfeld accuracy

7.3.3 Results

We examine the impact of certain feature types on the prediction of the variations types in our data. We are particularly interested in the interaction of voice and word order (precedence) since linguistic theories predict similar information structural factors guiding their use, but usually do not consider them in conjunction.

In Table 7.6, we report the performance of ranking models trained on the different feature subsets. The union of the features corresponds to the model trained on SEM_h in Experiment 4. At a very broad level, the results suggest that the precedence and scale alignment features interact both in the prediction of voice and word order. The union model performs best with respect to the sentence overlap measures and also the prediction of voice, the relative precedence of the verb arguments and the Vorfeld occupant.

The most pronounced effect of interaction can be seen when comparing the precedence model to the union model. Adding the surface-independent features to the precedence features leads to a big improvement in the prediction of word order. This is not a trivial observation since a) the surface-independent features do not discriminate between the realization candidates of an f-structure and b) the precedence features are built from the same properties. Thus, the SVM learner discovers dependencies between relative precedence preferences and abstract properties of a verb argument which cannot be encoded in the precedence alone.

It is worth noting that the precedence features improve the prediction of voice. This indicates that it should not be specified at a stage prior to word order, but rather in conjunction with the constituent linearization. Example (14) is taken from our development set, illustrating a case where the union model predicted the correct voice and word order (14-a), and the scale alignment model top-ranked the incorrect voice and word order. The active

verb arguments in (14-b) are both case-ambiguous and placed in the non-canonical order (object < subject), so the semantic relation can be easily misunderstood. The passive in (14-a) is unambiguous since the agent is realized in a PP (and placed in the Vorfeld).

(14) a. Von den deutschen Medien wurden die Ausländer nur
By the German media were the foreigners only
erwähnt, wenn es Zoff gab.
mentioned, when there trouble was.

b. Wenn es Zoff gab, erwähnten die Ausländer nur die
When there trouble was, mentioned the foreigners only the
deutschen Medien.
German media.

These results show that inter-dependencies between word order and voice are reflected in the feature model of a surface realization ranker. We now address this interaction by varying the way the training data for the ranker is labelled. We contrast two ways of labelling the sentences: a) all sentences that are not (nearly) identical to the reference sentence have the rank "4", irrespective of their voice (referred to as unlabelled model), b) the sentences that do not realize the correct voice are ranked lower than sentences with the correct voice (rank "4" vs. "5"), referred to as labelled model. Intuitively, the latter way of labelling tells the ranker that all sentences in the incorrect voice are worse than all sentences in the correct voice, independent of the word order. Given the first labelling strategy, the ranker can decide in an unsupervised way which combinations of word order and voice are to be preferred.

In Table 7.7, it can be seen that the unlabelled model improves over the labelled on all the sentence overlap measures. The improvements are statistically significant. This suggests that the surface realization ranker really takes advantage of the integrated setting where there is no conceptual difference between voice and word order paraphrases. The fact that we observe a particularly pronounced effect of the Exact Match improvement in Table 7.7 indicates that the unlabelled ranker is able to better predict certain special combinations of paraphrases. Moreover, we observe a more pronounced effect of the labeling scheme on the Precedence accuracy than on Voice accuracy. Thus, it seems to confuse the ranker that certain sentences are labelled with "5" although they have the correct word order.

Model	Match	BLEU	NIST	Voice	Prec.
Labelled, no LM	21.52	0.73	12.93	91.9	76.25
Unlabelled, no LM	26.83	0.75	13.01	91.5	80.19
Unlabeled + LM	27.35	0.75	13.08	91.5	79.6

Table 7.7: Experiment 7: Assessing the impact of language model scores and labelling scheme in an integrated surface realization ranking model for voice and word order

7.4 Evaluation for Combined Choices

Throughout this thesis, we have used a number of automatic evaluation measures for assessing the quality of our generation output. These measures intend to capture, or at least approximate the contextual fluency and appropriateness of a sentence in a discourse. In the paradigm of automatic evaluation, we compared the predicted sentence or referring expression with its original realization in the corpus. This method follows the standard practices in corpus-based NLG research, and it has been shown that measures like BLEU or NIST correlate with judgements made by native speakers. However, it should not be forgotten that this paradigm has well-known and maybe severe limitations. This situation is not special to NLG research, but also arises in other domains of NLP such as Machine Translation where the general, "human-like" quality of a linguistic output has to be assessed.

Thus, we are more or less certain that our automatic evaluation might only partially render certain effects, especially with respect to textual coherence. For instance, concerning the two texts in Figure 7.2, a human reader probably finds that the first discourse is not coherent and fluent at all whereas the second text snippet can be more or less easily understood. It is likely that the BLEU scores do not capture the magnitude of the differences in text quality as it only computes an n-gram overlap measure between the reference and the output. The example also shows that our integrated set-up rises a number of questions with respect to evaluation design. Thus, in the first discourse of Figure 7.2, it is not easy to tease apart the effects produced by syntactic ungrammaticalities and soft choices that are not appropriate in the context.

Section 7.4.1 and 7.4.2 propose two small studies that address the issue of evaluation in our extended surface realization scenario. They basically aim at showing some methodological issues that are particularly relevant

for our integrated generation scenarios. Section 7.4.3 points to some recent developments for methods of human evaluation, and discusses them in the light of our generation tasks.

7.4.1 N-best Evaluation

An important insight from studies of NLG evaluation is that the comparison of a single reference against a single output does not necessarily account for the acceptable variation that humans show in their judgements. Thus, an ideal data set for evaluation does not only give a single reference for a certain sentence or text, but several outputs produced by humans, as is done by Belz and Varges (2007). However, the collection of such data can be expensive and time-consuming again (similar to the collection of human judgements on generated outputs), so that a major advantage of corpus-based evaluation is lost. Therefore, we pursue the other possibility in this small pilot evaluation: instead of using several references, we evaluate several generated output sentences against the single reference.

We think that this method of n-best evaluation is reasonable for our surface realization ranking set-up: the ranker actually does not only select a single output sentence as optimal, but produces a global ranking over all candidates produced by the generator. The global ranking is completely ignored in the single-best evaluation. Thus, the rationale of the n-best evaluation is as follows: if the ranker did not select the exact candidate as the top-ranked realization, but as the second or third best, it can be assumed that the ranking still reasonably captures certain properties related to appropriateness or fluency.

In Table 7.8, we compare the n-best accuracies achieved by the labelled and unlabelled ranker for the joint prediction of voice and argument order. We focus on the accuracy measures for voice and precedence. The unlabelled model is very flexible with respect to the word order-voice interaction: the accuracy dramatically improves when looking at the top 3 sentences. The labelled model also improves by a large step, however, its prediction errors are not reduced as much as in the unlabelled model. Table 7.8 also reports the performance of an unlabelled model that additionally integrates LM scores. Surprisingly, these scores have a very small positive effect on the sentence overlap features and no positive effect on the voice and precedence accuracy. The n-best evaluations even suggest that the LM scores negatively impact the ranker: the accuracy for the top 3 sentences increases much less as compared

Model	Top 1 Prec.+Voice	Top 2 Prec.+Voice	Top 3 Prec.+Voice
Labelled, no LM	71.01	78.35	82.31
Unlabelled, no LM	74.51	84.28	88.59
Unlabeled + LM	73.92	79.74	82.89

Table 7.8: Experiment 7: n-best evaluation of realization rankers for accuracy of voice and precedence prediction

to the model that does not integrate LM scores.

This suggests that the n-best evaluation method can at least shed some light on the linguistic "behavior" of two comparable ranking models with respect to the range of possible variation. In particular, the language model scores seem to have an effect on the flexibility of the model by fixing certain surface patterns.

The n-best performance of a realization ranker is practically relevant for re-ranking applications such as Velldal (2008). We think that it is also conceptually interesting. Previous evaluation studies suggest that the original corpus sentence is not always the only optimal realization of a given linguistic input. Humans seem to have varying preferences for word order contrasts in certain contexts. The n-best evaluation could reflect the behavior of a ranking model with respect to the range of variations encountered in real discourse. The pilot human evaluation in the next Section deals with this question.

7.4.2 Human Evaluation

In this thesis, we have repeatedly looked at accuracy measures for quantifying the quality of the surface realization ranking output. For instance, we used 'Voice Accuracy' for measuring the proportion of test sentences where the generator ranked a candidate sentence on top that realizes the main verb in the same voice as in the original corpus sentence. This accuracy measure reduces the question whether voice of a particular verb in a particular sentence is natural and contextually appropriate to a binary decision. Thus, it is a relatively strict measure as we know that the usage of grammatical alternations like voice or word order is, in many contexts, a soft rather than a hard decision.

For this reason, human judgements are an important evaluation method

in NLG as the quality of the generation output can be assessed more directly. We conducted a human evaluation where we asked participants to judge the quality of the top-ranked sentences predicted to be natural by the surface realization ranking in Experiment 7. The main goal of this evaluation is to measure the naturalness of the voice decisions made by the generator. But, of course, when humans read a sentence and judge its naturalness and clarity, they do not focus on particular linguistic phenomena. Their judgement is based on many intuitive factors. In particular, in our data, their judgement will not only relate to the quality of voice decisions but also to the word order in the automatically generated sentence. Generally, such interacting phenomena are notoriously hard to tease apart in NLG evaluation (see discussion below).

In addition to the challenge posed by interacting choices, the n-best evaluation in Section 7.4.1 suggests that the surface realization model is, to a certain extent, flexible with respect to its voice decisions. The results presented in Section 7.4.1 have shown that the voice accuracy of our linguistically informed ranking model increases by more than 10% when we consider the three best candidates rather than only the top-ranked sentence. Thus, there is a certain proportion of generation inputs in our validation data where the model makes variable voice predictions, looking at sentences in the top-3 list. Table 7.10 illustrates two example cases (used in the questionnaire detailed below) where the surface realizer ranked sentences with different voices on top. These cases will be called "mixed items". On the other hand, there are cases where the sentences that the model ranks on top are consistently correct or false with respect to their voice. Table 7.10 illustrates an example of a "correct item" (i.e. top-3 sentences generated in the original voice) and a "false item".

Therefore, this human evaluation is designed to answer a very specific question about the voice decisions made by the generator. We are mostly interested in those cases where the model ranked sentences with variable voice on top ("mixed" items). We will investigate whether these cases are also more difficult to judge for humans. As a measure of difficulty, we use agreement between human judges on ranks that they assign to generated sentences.

Hypothesis Previous studies in generation mainly used human evaluation to compare different systems, or to correlate human and automatic evalua-

tions. Our primary goal is to assess how the voice decisions made by the surface realization ranker in Experiment 7 impact on the naturalness of the generation output. The n-best evaluation in Section 7.4.1 showed that there is a considerable number of cases where the top-ranked sentence did not realize the same voice as the original corpus sentence, but sentences with rank two or three realize the original voice. Our analysis aims to establish whether variable voice predictions made by our model (in the n-best list) are reflected in the agreement or correlation between human rankings. In particular, we want to establish whether the agreement between humans is higher in certain contexts than in others. In order to select these contexts, we use the predictions made by our ranking model. The rationale is as follows: if humans tend to disagree in cases where the ranking model selects different voice realizations in its n-best list, the model can be claimed to account for the variable nature of certain choices.

Method As previous studies have been more successful for relative comparisons between sentences than for absolute scores assigned to single sentences, we design the our experimental items such that they contain several surface realization candidates for a certain generation input. We used a questionnaire that shows the preceding corpus sentence and presents several surface realization outputs for the following corpus sentence.

Experimental Material The questionnaire for our experiment comprised 24 items falling into 3 classes

- ("Correct"): items where the 3 best sentences predicted by the model have the same voice as the original sentence (+ 2 sentences with the other voice), see Item A in Table 7.10
- ("Mixed"): items where the 3 top-ranked sentences realize different voices (+ 2 other random sentences), see Item C and D in Table 7.10
- ("False"): items where the model predicted the incorrect voice in all 3 top sentences (+ 2 sentences with the other voice), see Item B in Table 7.10

Each item in the questionnaire is composed of the original sentence, the 3 top-ranked sentences (if not identical to the corpus sentence) and 2 further

Item A	
Gold: active, predicted: active ("correct")	
gold	Die Banken hätten allerdings noch nicht entschieden, ob sie diesem The banks had however yet not decided, whether they this.DAT Plan zustimmten. plan agreed. (*However, the banks had not yet decided whether they would agree to this plan.*)
top-3	Noch nicht hätten allerdings die Banken entschieden, ob sie diesem Plan zustimmten.
top-1	Allerdings hätten die Banken noch nicht entschieden, ob sie diesem Plan zustimmten.
> 3	Noch nicht hätten allerdings die Banken entschieden, ob diesem Plan zugestimmt werde.
> 3	Allerdings hätten die Banken noch nicht entschieden, ob diesem Plan zugestimmt werde.
> 3	Die Banken hätten noch nicht allerdings entschieden, ob diesem Plan zugestimmt werde.
Item B	
Gold: active, predicted: passive ("false")	
top-2	Seine Suite wurde von mindestens 20 bewaffneten Männern im 14. Stock His suite was by at least 20 armed men in the 14th floor geschützt. protected. (*His suite in the 14th floor was protected by at least 20 armed men.*)
top-1	Im 14. Stock wurde seine Suite von mindestens 20 bewaffneten Männern geschützt.
gold	Mindestens 20 bewaffnete Männer schützten seine Suite im 14. Stock.
> 3	Mindestens bewaffnete 20 Männer schützten seine Suite im 14. Stock.
top-3	Seine Suite wurde im 14. Stock von mindestens 20 bewaffneten Männern geschützt.
> 3	Im 14. Stock schützten mindestens bewaffnete 20 Männer seine Suite.

Table 7.9: Example items used in the human evaluation for Experiment 7

Item C	
Gold: passive, predicted: active/passive ("mixed")	
top-2	In dem Entwurf wird der Einsatz von Streitkräften nicht direkt erwähnt. In the draft was the mission of armed forces not directly mentioned. (*The mission of the armed forces was not directly mentioned in the draft.*)
> 3	Der Einsatz von Streitkräften wird nicht direkt in dem Entwurf erwähnt.
> 3	Den Einsatz von Streitkräften erwähnt der Entwurf nicht direkt.
top-1	Der Entwurf erwähnt den Einsatz von Streitkräften nicht direkt.
gold	Der Einsatz von Streitkräften wird in dem Entwurf nicht direkt erwähnt.
top-3	Der Entwurf erwähnt nicht direkt den Einsatz von Streitkräften.
Item D	
Gold: passive, predicted: passive/active (mixed)	
top-1/ gold	Dieses Territorium müsse vor Übergriffen Zagrebs geschützt werden, This territory has against attacks Zagreb.GEN protected be, meinen trotz aller Kriegsmüdigkeit nach wie vor viele Serben. think despite all war-weariness still many Serbs. (*Despite all war-weariness, many Serbs still think that this territory has to be protected against attacks by Zagreb.*)
top-2	Dieses Territorium müsse man vor Übergriffen Zagrebs schützen, meinen viele Serben nach wie vor trotz aller Kriegsmüdigkeit.
> 3	Viele Serben meinen nach wie vor trotz aller Kriegsmüdigkeit, vor Übergriffen Zagrebs müsse dieses Territorium geschützt werden.
> 3	Man müsse dieses Territorium vor Übergriffen Zagrebs schützen, meinen viele Serben nach wie vor trotz aller Kriegsmüdigkeit.
top-3	Dieses Territorium müsse vor Übergriffen Zagrebs geschützt werden, meinen nach wie vor viele Serben trotz aller Kriegsmüdigkeit.
> 3	Trotz aller Kriegsmüdigkeit meinen nach wie vor viele Serben, vor Übergriffen Zagrebs müsse dieses Territorium geschützt werden.

Table 7.10: Example items used in the human evaluation for Experiment 7

	Items			
	All	Correct	Mixed	False
Top-ranked original sent.	84%	78%	83%	91%

Table 7.11: Human judgements for Experiment 7: How often did participants assign the top rank to the original corpus sentence?

sentences such that each item contains different voices. For each item, we presented the previous context sentence.

The experiment was completed by 8 participants, all native speakers of German, 5 had a linguistic background. The participants were asked to rank each sentence on a scale from 1-6 according to its naturalness and plausibility in the given context. The participants were explicitly allowed to use the same rank for sentences they find equally natural. The participants made heavy use of this option: out of the 192 annotated items, only 8 are ranked such that no two sentences have the same rank.

Results In Table 7.11, we show the proportions of annotated items where the participants assigned the best rank to the original corpus sentence. Thus, there is a considerable amount of cases (16%) where humans preferred a generated sentence to an original corpus sentence. This proportion is even higher when we look at items where the system predicts the correct voice (22%), but considerably lower in cases where the system predicted the "false" voice in the top three sentences (9%). This seems to suggest that the sentences are generally of good quality, and account for certain aspects of contextual naturalness. On the other hand, there seems to be some systematic errors made by the system. In the false items, humans consistenly prefer the corpus sentence with the original voice.

In a more detailed analysis, we compare the human judgements by correlating them with Spearman's ρ. This measure is considered appropriate for graded annotation tasks in general (Erk and McCarthy, 2009), and has also been used for analysing human realization rankings (Velldal, 2008; Cahill, 2009). We normalize the ranks according to the procedure in Velldal (Velldal, 2008), applying the z-transformation.

In Table 7.12, we report the correlations obtained from averaging over all pairwise correlations between the participants and the correlations restricted to the item and sentence classes. For instance, the correlations on "correct

	Items			
	All	Correct	Mixed	False
All sent.	0.58	0.6	0.54	0.62
"Correct" voice	0.64	0.63	0.56	0.72
"False" voice	0.47	0.57	0.48	0.44

Table 7.12: Human judgements for Experiment 7: averaged pairwise correlation between participants

voice" sentences in the "false items" are computed on the subset of sentences in these items that where not on the 3-best list predicted by the ranker. We used bootstrap re-sampling on the pairwise correlations to test that the correlations on the different item classes significantly differ from each other.

The correlations in Table 7.12 show that the agreement between annotators is highest on the false items, and lowest on the mixed items. This complies with the previous observation that humans tended to give the best rank to the original sentence more often on the false items (91%) than on the others. Moreover, the agreement is generally higher on the sentences realizing the correct voice.

These results seem to confirm our hypothesis that the general level of agreement between humans differs depending on the context. However, one has to be careful in relating the effects in our data solely to voice preferences. Since the sentences were chosen automatically, some examples contain very unnatural word orders that probably guided the annotators' decisions more than the voice. This is illustrated by Example (15) showing two passive sentences from our questionnaire which differ only in the position of the adverb *besser "better"*. Sentence (15-a) is completely implausible for a native speaker of German, whereas Sentence (15-b) sounds very natural.

(15) a. Durch das neue Gesetz sollen **besser** Eigenheimbesitzer
By the new law should better house owners
geschützt werden.
protected be.

b. Durch das neue Gesetz sollen Eigenheimbesitzer **besser**
By the new law should house owners better
geschützt werden.
protected be.

This observation brings us back to our initial point stating that the surface realization task is especially challenging due to the interaction of a range of discourse phenomena. Obviously, this interaction makes it difficult to single out preferences for a specific alternation type. Future work will have to establish how this problem should be dealt with in the design of human evaluation experiments.

7.4.3 Discussion

From a methodological perspective, the human evaluation carried out on our surface realization data indicates that not only automatic evaluation measures have their shortcomings, but also experimental settings for human judgement collection should be designed with care. Thus, even in a grammar-based generation scenario where all surface string are grammatical, we have the intuition that there are different context effects that one would, ideally, like to be able to distinguish. On the one hand, there are naturalness effects that concern the fluency of certain sentence-internal, idiosyncratic patterns. In our data, we typically observe them with the placement of adverbs (also see Chapter 2.2.2 for an Example) which is more interesting from a syntactic, sentence-internal perspective than from a global discourse and coherence-oriented point of view. On the other hand, there are coherence effects that concern the apropiateness of the sentence in the given context. We expect that the overlap between these dimensions would be very problematic for the human evaluation of our purely statistical system presented in Experiment 7 that also generates ungrammatical output.

Recently, Siddharthan and Katsos (2012) have presented an evaluation study that uses psycholinguistic offline readability measures to tease apart aspects of acceptability from breakdowns in comprehension. They propose a sentence recall method where participants in the experiments have to reproduce a sentence that they saw before on the screen, and a method for magnitude estimation where participants rate sentences. They find that the recall method is successful in teasing apart comprehension breakdown and acceptability for long sentences. For short sentences, they find that participants recall them although the give bad judgements in terms of grammaticality. Their results also show that magnitude estimation does not have the undesired interaction with sentence length and is useful for testing particular hypotheses formulated for the data. However, it is not useful for teasing apart comprehension and acceptability.

7.5 Conclusion

The experiments presented in this Chapter generally support approaches and arguments from early rule-based works on NLG architectures that have shown various problems with sequential generation pipelines that do not allow for interaction between discourse-level and sentence-level decisions. First, in Experiment 7, we have presented a data-driven approach for investigating generation architectures in a modular setting for surface realization and referring expression generation. The data set we created for our experiments basically integrates standards from previous research on REG and surface realization and extends the annotations to further types of implicit referents. Our results show that interactions between the different generation levels are best captured in a sequential, revision-based pipeline where the REG component has access to predictions from the syntax and the linearization module. These empirical findings obtained from experiments with generation architectures have clear connections to theoretical accounts of textual coherence like Centering that model close dependencies between word order and referring expression realization.

Second, in Experiment 8, we have analyzed interaction effects between voice and word order in our grammar-based surface realization architecture. Due to the limited increase in surface realization candidates, it is possible to address these two phenomena in a completely integrated way. We showed that is beneficial for the ranking model if no conceptual difference between these choices is encoded in the labelling or the feature model.

Looking at the results from Experiment 8, it would be an obvious direction for future work on our robbery data set to implement a more integrated version of the combined surface realization and REG. As the full combination of RE candidates, alternations and linear order leads to a considerable expansion of the search space, the set-up of such a formalism is more involved than for the extension of the grammar-based surface realization. For instance, it would not be possible to use (Bohnet et al., 2012) off-the-shelf, state-of-the-art linearization tool in such an integrated setting.

Finally, we have seen that our multi-level generation scenarios raise some interesting issues for automatic and human evaluation of a generated system output. Thus, the fact that choice in natural language use is often variable, soft and gradient entails that certain deviations from a reference (corpus) sentences should be allowed in a generation output that, still, has to obey certain fluency and appropriateness constraints. An ideal evaluation method

would tease apart the various dimensions of output quality: grammaticality, fluency, coherence, etc. Existing evaluation methods tend to conflate them.

Chapter 8

Conclusions

8.1 Summary

In this thesis, we have investigated a range of corpus-based generation settings and their implications for the underlying statistical models of linguistic choice. According to the three dimensions of a generation framework defined in the Introduction - the source, the architecture and the context model - we have investigated the following topics: In terms of input representations, we have looked at sources for modeling word order, syntactic alternations, and referring expressions, either separately or in a combined way. Our generation architectures included a standard grammar-based generate-and-rank architecture for surface realization that we extended to produce candidates for voice alternations, a dependency-based setting where generation is modeled as a sequence of steps transforming an unordered, deep dependency structure into an ordered surface dependency tree, and a referring expression realization component that selects surface forms for referent slots in an annotated text. The context models we implemented represented context in terms of sentence-internal and sentence-external factors, factors based on high-quality, fine-grained information from morpho-syntactic analysis, and less accurate factors reflecting error propagation effects in a generation history. What are the insights that we gained from this panoply of experiments?

The overarching theme that emerged from the studies we reported in this work is the pervasiveness of interacting choices that affects all dimensions of the set-up of an NLG system. This theme has always been one of the most important threads underlying NLG research, and has been addressed

in a number of rule-based NLG applications designed as being complex, but effective and conceptually accurate models of choice processes in language generation. Recently, with the rise of statistical methods in NLG and the efforts oriented towards applying comparable, flexible and trainable methods on broad-coverage data sets with annotations being available from complementary approaches in NLU, the problem of interacting choice has been often - seemingly - set on the side.

The two dominating tasks in corpus-based NLG, surface realization and referring expression generation, are typically set-up in way that the system has to generate from a source that isolates a particular choice phenomenon, and pre-specifies all the remaining information necessary to produce a surface sentence. Thus, this set-up seems to control interactions by cutting down the potential complexity on the level of the source, and the required architecture as only single components are needed to predict these isolated choice phenomena. However, as our detailed analysis of context models for word order prediction in surface realization and referring expression generation in Chapter 4 has shown, the effect of interaction is strongly reflected in the contextual factors exploited by state-of-the-art surface realizers and REG system. These systems implement context models that are typically based on the detailed information about morpho-syntactic sentence-internal context given in the generation input. While this approach provides insights into the multitude of factors potentially accounting for the appropriateness of a particular choice in a given context, these results also suggests that, from a broader perspective, the accurate performance of state-of-the-art realizers builds on some artificial assumptions.

In order to develop a generation framework for extended candidate generation where a wider range of choice can be captured, we implemented surface realization from a more abstract source, namely meaning representations derived from F-structures, in Chapter 5. Using a grammar-based generator, we found that the desired generation candidates could not be produced for a substantial proportion of the input data, due to contextual cues that blocked a well-formed realization of the alternation. This suggests that interactions between choices as they were realized in the original corpus sentence are still reflected in the representation derived in the analysis process, similar to reflections on the level of contextual factors found in Chapter 4. In order to be able to study voice alternations in a variety of contexts, we designed heuristic rules for deriving semantic representations suited for broad-coverage alternation generation in a generate-and-rank scenario. As we have shown

in Chapter 6, this strategy produces a better source such that the context model can be trained on more extensive candidate sets and pick up some underlying preferences for realizing voice alternations.

Since we also encountered some rather technical issues with our grammar-based generator which exhibits idiosyncratic requirements on its input (this problem has also been reported for similar systems), we suspected that an extensive treatment of other alternation like e.g. nominalizations and a large-scale account of REG in a grammar-based generation framework would be hard to achieve. Therefore, we used a dependency-based generation architecture for combined syntactic realization, REG and linearization in a more flexible set-up. Instead of relying on a grammar as a "black box" that produces surface realization candidates, we decomposed the generation problem into several modules where certain decisions and choices can be modeled locally. Moreover, the dependency-based components are purely statistical, meaning that they can be trained on different inputs. This has been a central technical prerequisite for carrying out our experiment that compared several generation architectures instantiating different organization modes for interacting modules, see Chapter 7. We showed that such a flexible modular architecture is able to capture certain interactions, and alleviate certain error propagation effects found in the standard NLG pipeline, by implementing some simple revision mechanisms where linearization is applied before and after the realization of referring expressions.

All these general considerations relating to the source, architecture and context model implemented in an NLG system, come together in a particular, concrete phenomenon, that has been a recurrent issue throughout the different studies in this thesis, namely the treatment of implicit arguments or implicit mentions of referents. This phenomenon has not received much attention in state-of-the-art NLG, but we found that it is decisive for being able to model certain syntactic alternations. We designed heuristic rules which establish, for instance, the alternation relation between an active with a generic agent and a passive that does not overtly realize the agent. This strategy leads to a better balanced distribution of the alternations in the training data, such that our linguistically informed generation ranking model distinguishes active and passive in a range of contexts. In the dependency-based setting, we adopt an entity-centric annotation approach inspired from other GREC data sets, where all implicit mentions of the central entities are marked throughout the text. Based on this data set, we found that the decision to include implicit referents in a generation source has far-reaching

consequences for the contextual modeling in an REG component, as it leads to increased uncertainty in the previous generation context.

8.2 Directions

There are a number of immediate possible extensions that would further increase the value and expressiveness of the generation settings we developed in this thesis. Thus, by moving from F-structures to dependencies, we payed a certain price for the flexibility that we gained from the more shallow dependency-based generation framework. In this setting, we cannot really leverage the huge amount of syntactic knowledge encoded in a grammar-based generator and the generation process does not distinguish between hard and soft constraints that hold for a particular choice. Therefore, the effect of using a carefully designed representation including implicit referents against a representation obtained in a more naive way is less pronounced. We showed that this is related to the noisy representation of argument structure in the shallow automatic dependency analysis. Related to the lack of a model of hard syntactic constraints, we would also have to integrate a proper treatment of morpho-syntactic constraints into our generator. We experimented with the purely statistical morphological generation component developed by Bohnet et al. (2010) but could not get satisfactory results for our German texts. In order to work around the morphology problem, our dependency-based generator simply produced lemmatized sentences. This lemmatized output is not well-suited for human evaluation such that we have to restrict ourselves to automatic evaluation measures. It would be interesting to explore a hybrid set-up for a generation architecture where the respective strengths, an accurate grammatical model of hard constraints on the one hand, and a flexible trainable component for processing arbitrary inputs, can be combined. Similar ideas have been pursued in hybrid approaches to parsing.

In terms of our architectural set-up for capturing interactions between choices, we see our revision-based technique as a first, straightforward and transparent approximation of some potentially more sophisticated techniques that could model some interaction between the generation components, or further decompose the generation process in the style of Angeli et al. (2010)'s approach. Recently, joint optimization techniques have also gained interest for probabilistic approaches to parsing, e.g. proposed in Bohnet et al. (2013),

Moreover, we believe that our robbery data set would be suited to create even more abstract inputs appropriate for a systematic and feasible approach to microplanning, as most of the texts in our data set are still relatively short and the texts describe very similar events. Thus, by removing information about sentence boundaries, and decomposing complex sentences, an interesting source for generation tasks going beyond the sentence-level could be constructed.

Another open question that suggests itself, but goes beyond the core domain of NLG, relates to the analysis of contextual factors accounting for a set of combined choices. Compared to the theoretically motivated approaches in the style of Bresnan et al. (2007), discussed in Chapter 3, our results have established rather coarse-grained tendencies for broad features classes, demonstrating e.g. the impact of linearization features on REG. In future work, a more detailed analysis of contextual factors on a, possibly manually revised, data set could provide deeper insights into the role and effect of a particular contextual factor and its relation to some combined choice. For instance, our integrated account of passive and word order variation in Chapter 7.3 has not quite settled the question whether there is a certain division of labour between these two alternative choices for moving constituents in a free word order language.

Finally, we think that our work points to some important issues, equally relevant in certain fields of NLU: Our studies highlight that a treatment of implicit information on the level of semantic analysis is central for obtaining corpus-based meaning representations that are suitable for applications. Similar observations have been made in e.g. the field of textual entailment where some first resources start being created, e.g. Gerber and Chai (2010).

At the very end, we want to mention an extensive quote from McDonald (1993), as it nicely puts the issues addressed in this thesis in a larger perspective:

> The lack of a consistent answer to the question of the generator's source has been at the heart of the problem of how to make research on generation intelligible and engaging for the rest of the computational linguistics community, and has complicated efforts to evaluate alternative treatments even for people in the field. Nevertheless, a source cannot be imposed by fiat. Differences in what information is assumed to be available, its relative decomposition when compared to the "packaging" available in

> the words or syntactic constructions of the language (linguistic resources), what amount and kinds of information are contained in the atomic units of the source, and what sorts of compositions and other larger scale organizations are possible–all these have an impact on what architectures are plausible for generation and what efficiencies they can achieve. Advances in the field often come precisely through insights into the representation of the source.
> ...
> It is somewhat puzzling that this question of where the comprehension process ends has apparently never been debated in the literature. Instead it seems largely taken for granted that the parsing process ends with the assembly of an expression in a suitable logic that captures the text's information content, perhaps with some functional annotations, and that a "reasoning" process then starts with that expression and draws inferences in order to resolve anaphors and establish the speaker's intent.

Essentially, the experimental set-ups discussed in this thesis are an attempt to get a deeper understanding for models of choice in generation in relation to the underlying source of the generation process. We exploited data-driven-techniques for assessing the effect of interactions between choices at the level of generation source, architecture and context modeling. As the subtle interactions always cut across the architectural set-up, it remains a methodologcially challenging path that may require some unconventional decisions. But we think that with the advances in data-driven techniques, natural language generation, and maybe computational linguistics in general, has an increasingly powerful tool box for addressing the source question.

Appendix A

Transfer grammar for the active/passive alternation

A.1 Marking agents

```
" PRS (1.0) "

grammar = 'mark-agents'.

:- set_transfer_timeout_limit(1000000, 1000000, 2000000).
:- set_transfer_option(include_cstr, 1).
:- set_transfer_option(normalize, 1).


sem_arg(%Arg) :=
 ( ( +PRED(%Arg,%%), -PRON-TYPE(%Arg,null) )
  | ( +COORD(%Arg,%%) )
  | ( +PRED(%Arg,%%), +PRON-TYPE(%Arg,null), +PRED-RESTR(%Arg,%%))
 ).

generic_arg(%Arg) :=
  +PRED(%Arg,pro),
  ( +PRON-FORM(%Arg,man)
  | +PRON-FORM(%Arg,jemand)
  | +PRON-FORM(%Arg,irgendjemand)
  | +PRON-FORM(%Arg,niemand)
  | +PRON-FORM(%Arg,etwas)
  | +PRON-FORM(%Arg,irgendetwas)
  | +PRON-FORM(%Arg,nichts)
  | +PERS(%Arg,1)
```

```
  ).

agent_prep(%Arg) :=
  ( +PRED(%Arg,durch)
  | +PRED(%Arg,mit)
  "| +PRED(%Arg,bei)"
  | +PRED(%Arg,in)
  | +PRED(%Arg,so)
  | +PRED(%Arg,dort)
  | +PRED(%Arg,da), -CLAUSE-TYPE(%Arg,%%)
  | +PRED(%Arg,hier)
  ).

mark_verb(%VerbPred) ::
  +VTYPE(%N,main), +PRED(%N,%VerbPred), +PASSIVE(%N,+),
  -VCOMP(%%,%N), -GEN-CAT(%%,%%), +arg(%N,1,%Arg1),
  @sem_arg(%Arg1), +arg(%N,2,%Arg2), @sem_arg(%Arg2)
  ==>
  GEN-CAT(%N,PASSIVE-2-ARG);

  +VTYPE(%N,main), +PRED(%N,%VerbPred), +PASSIVE(%N,-),
  -VCOMP(%%,%N), -GEN-CAT(%%,%%), +arg(%N,1,%Arg1),
  @sem_arg(%Arg1), +arg(%N,2,%Arg2), @sem_arg(%Arg2)
  ==>
  GEN-CAT(%N,ACTIVE-2-ARG);

  GEN-CAT(%N,ACTIVE-2-ARG), +arg(%N,1,%Arg1),@generic_arg(%Arg1)
  ==>
  GEN-CAT(%N,ACTIVE-2-ARG-GENERIC);

  -GEN-CAT(%%,%%), +VTYPE(%N,main), +PRED(%N,%VerbPred),
  +PASSIVE(%N,+), -VCOMP(%%,%N), -ATYPE(%N,%%),
  +arg(%N,1,NULL), +arg(%N,2,%Arg1), @sem_arg(%Arg1)
  ==>
  GEN-CAT(%N,PASSIVE-1-ARG);

  GEN-CAT(%N,ACTIVE-2-ARG), +arg(%N,1,%Arg1),@generic_arg(%Arg1)
  ==>
  GEN-CAT(%N,ACTIVE-2-ARG-GENERIC);

  GEN-CAT(%N,ACTIVE-2-ARG), +arg(%N,1,%Arg1), -PRON-TYPE(%Arg1,rel),
  +in_set(%N,%CoordN),+COORD(%CoordN,+), COORD-LEVEL(%CoordN,%%),
  +in_set(%N2,%CoordN),+arg(%N2,%%,%Arg1), {\+ %N = %N2}
  ==>
  GEN-CAT(%N,ACTIVE-2-AGENTCOORD);
```

```
GEN-CAT(%N,ACTIVE-2-ARG), +arg(%N,1,%Arg1), +PRED(%Arg1,pro),
+PRON-FORM(%Arg1,%%), +GEND(%Arg1,neut), -PRON-TYPE(%Arg1,int),
-PRON-TYPE(%Arg1,rel)
==>
GEN-CAT(%N,ACTIVE-2-NEUTPRONAGENT);

GEN-CAT(%N,ACTIVE-2-ARG), +arg(%N,1,%Arg1), +PRED(%Arg1,pro),
+PRON-FORM(%Arg1,%%), (+COMP-FORM(%N,%%) | +ADJ-REL(%%,%N)),
-PRON-TYPE(%Arg1,int), -PRON-TYPE(%Arg1,rel)
==>
GEN-CAT(%N,ACTIVE-2-EMBPRONAGENT);

GEN-CAT(%N,ACTIVE-2-ARG), +arg(%N,1,%Arg1), +PRED(%Arg1,pro),
+PRON-FORM(%Arg1,%%), -PRON-TYPE(%Arg1,int), -PRON-TYPE(%Arg1,rel),
+OBJ(%C,%N), +CLAUSE-TYPE(%C,%%)
==>
GEN-CAT(%N,ACTIVE-2-EMBPRONAGENT);

GEN-CAT(%N,PASSIVE-1-ARG), +ADJUNCT(%N,%Adjuncts),
+in_set(%Adj,%Adjuncts), @agent_prep(%Adj)
==>
GEN-CAT(%N,PASSIVE-ADJ-AGENT);

GEN-CAT(%N,PASSIVE-1-ARG), +XCOMP(%X,%N),
+ADJUNCT(%X,%Adjuncts),  +in_set(%Adj,%Adjuncts), @agent_prep(%Adj)
==>
GEN-CAT(%N,PASSIVE-ADJ-AGENT);

GEN-CAT(%N,PASSIVE-1-ARG), -ATYPE(%N,%%), +in_set(%N,%CoordN),
+COORD(%CoordN,+), COORD-LEVEL(%CoordN,%%), +in_set(%N2,%CoordN),
+PASSIVE(%N2,-),+PRED(%N2,%%), +SUBJ(%N2,%%), +arg(%N2,2,%%),
+lex_id(%N,%Id1), +lex_id(%N2,%Id2), {%Id1 > %Id2}
==>
GEN-CAT(%N,PASSIVE-AGENTCOORD);

GEN-CAT(%N,PASSIVE-1-ARG), +OBJ(%C,%N), +CLAUSE-TYPE(%C,%%)
==>
GEN-CAT(%N,PASSIVE-1-EMBEDDED);

GEN-CAT(%N,PASSIVE-1-ARG), (+COMP-FORM(%N,%%)  |  +ADJ-REL(%%,%N))
==>
GEN-CAT(%N,PASSIVE-1-EMBEDDED);

-GEN-CAT(%%,%%),+VTYPE(%N,main), +PRED(%N,%VerbPred)
```

```
  ==>
  GEN-CAT(%N,FILTER-VCOMP),
  ALT-CAT(%N,FILTER-VCOMP).


+FIRST(%F,%%), -GEN-CAT(%%,%%)
==>
GEN-CAT(%F,FILTER).

@mark_verb(schützen).

+GEN-CAT(%N,%%),CHECK(%N,%CHECK),
_VLEX(%CHECK,%V),_AUX-SELECT(%V,%%)
==> 0.

+GEN-CAT(%N,%%),CHECK(%N,%CHECK),
_VLEX(%CHECK,%V),_AUX-SELECT(%V,%%)
==> 0.

+GEN-CAT(%N,%%),+TNS-ASP(%N,%CHECK),
PASS-ASP(%CHECK,%%)
==> 0.

+GEN-CAT(%N,%%), ( +SUBJ(%N,%C) | +OBJ(%N,%C) ),
CASE(%C,%%)
==> 0.

+GEN-CAT(%N,%%), ( +SUBJ(%N,%C) | +OBJ(%N,%C) ),
CHECK(%C,%%)
==> 0.

+GEN-CAT(%N,%%), ( +SUBJ(%N,%C) | +OBJ(%N,%C) ),
+COORD(%C,+), +in_set(%CONJ,%C), CHECK(%CONJ,%%)
==> 0.

+GEN-CAT(%N,%%), ( +SUBJ(%N,%C) | +OBJ(%N,%C) ),
+COORD(%C,+), +in_set(%CONJ,%C), CASE(%CONJ,%%)
==> 0.
```

A.2 Rules for mapping passive to active F-structures

```
+GEN-CAT(%N,PASSIVE-2-ARG),
+OBL-AG(%N,%AG), +OBJ(%AG,%LoSUBJ),
ADJUNCT(%AG,%Adjunct)
==>
ADJUNCT(%LoSUBJ,%Adjunct).


+GEN-CAT(%N,PASSIVE-ADJ-AGENT),
+ADJUNCT(%N,%Adjuncts),
+in_set(%Adj,%Adjuncts), @agent_prep(%Adj), +OBJ(%Adj,%LoSUBJ),
ADJUNCT(%Adj,%OAdjunct)
==>
ADJUNCT(%LoSUBJ,%OAdjunct).

( +GEN-CAT(%N,PASSIVE-1-ARG)
 |+GEN-CAT(%N,PASSIVE-2-ARG)
 |+GEN-CAT(%N,PASSIVE-ADJ-AGENT)
),
+in_set(%N,%CoordN),+COORD(%CoordN,+),
+in_set(%N2,%CoordN), {\+ %N = %N2},
+lex_id(%N,%Id1), +lex_id(%N2,%Id2), {\+ %Id1 = %Id2},
+PRED(%N2,%%), +SUBJ(%N2,%Subj),
+NUM(%Subj,%Num), +PERS(%Subj,%Pers),
SUBJ(%N,%Subj),arg(%N,2,%Subj), COORD-LEVEL(%CoordN,%%)
==>
SUBJ(%N,%NewSubj), arg(%N,2,%NewSubj),
PRED(%NewSubj,pro), PRON-FORM(%NewSubj,sie),
PRON-TYPE(%NewSubj,pers), NUM(%NewSubj,%Num), PERS(%NewSubj,%Pers).

( +GEN-CAT(%N,PASSIVE-1-ARG)
 |+GEN-CAT(%N,PASSIVE-2-ARG)
 |+GEN-CAT(%N,PASSIVE-ADJ-AGENT)
),
+XCOMP(%X,%N),
+in_set(%X,%CoordN),+COORD(%CoordN,+),
+in_set(%N2,%CoordN), {\+ %X = %N2},
+lex_id(%X,%Id1), +lex_id(%N2,%Id2), {\+ %Id1 = %Id2},
+PRED(%N2,%%), +SUBJ(%N2,%Subj),
+NUM(%Subj,%Num), +PERS(%Subj,%Pers),
SUBJ(%N,%Subj),arg(%N,2,%Subj), COORD-LEVEL(%CoordN,%%)
==>
```

```
SUBJ(%N,%NewSubj), arg(%N,2,%NewSubj),
PRED(%NewSubj,pro), PRON-FORM(%NewSubj,sie),
PRON-TYPE(%NewSubj,pers), NUM(%NewSubj,%Num), PERS(%NewSubj,%Pers).


( +GEN-CAT(%N,PASSIVE-1-ARG)
 |+GEN-CAT(%N,PASSIVE-2-ARG)
 |+GEN-CAT(%N,PASSIVE-ADJ-AGENT)
),
+in_set(%N,%CoordN),+COORD(%CoordN,+),
+in_set(%N2,%CoordN), {\+ %N = %N2},
+lex_id(%N,%Id1), +lex_id(%N2,%Id2), {\+ %Id1 = %Id2},
+PRED(%N2,%%), +SUBJ(%N2,%%),
+TNS-ASP(%N2,%Tns2), +TENSE(%Tns2,%Tense),+MOOD(%Tns2,%Mood),
TNS-ASP(%N,%Tns2)
==>
TNS-ASP(%N,%NewTns), TENSE(%NewTns,%Tense),MOOD(%NewTns,%Mood).

( +GEN-CAT(%N,PASSIVE-1-ARG)
 |+GEN-CAT(%N,PASSIVE-2-ARG)
 |+GEN-CAT(%N,PASSIVE-ADJ-AGENT)
),
+in_set(%N,%CoordN),+COORD(%CoordN,+),
+in_set(%N2,%CoordN), {\+ %N = %N2},
+lex_id(%N,%Id1), +lex_id(%N2,%Id2), {\+ %Id1 = %Id2},
+PRED(%N2,%%), +SUBJ(%N2,%%),
ADJUNCT(%CoordN,%Adjuncts)
==>
ADJUNCT(%N,%Adjuncts).

+GEN-CAT(%N,PASSIVE-2-ARG), OBL-AG(%N,%AG), arg(%N,%%,%AG),
PRED(%AG,%%), OBJ(%AG,%LoSUBJ), PSEM(%AG,%%),
PTYPE(%AG,%%), SUBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-ARG),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
OBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,-).

+GEN-CAT(%N,PASSIVE-ADJ-AGENT),
+ADJUNCT(%N,%Adjuncts), in_set(%Adj,%Adjuncts), @agent_prep(%Adj),
PRED(%Adj,%%),PTYPE(%Adj,%%), arg(%Adj,%%,%LoSUBJ),
OBJ(%Adj,%LoSUBJ), SUBJ(%N,%SUBJ),
arg(%N,1,NULL), arg(%N,2,%SUBJ),
```

```
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-ARG),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
OBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,-).

+GEN-CAT(%N,PASSIVE-ADJ-AGENT),
+XCOMP(%X,%N),
+ADJUNCT(%X,%Adjuncts), in_set(%Adj,%Adjuncts), @agent_prep(%Adj),
PRED(%Adj,%%),PTYPE(%Adj,%%), arg(%Adj,%%,%LoSUBJ),
OBJ(%Adj,%LoSUBJ),SUBJ(%N,%SUBJ),
arg(%N,1,NULL), arg(%N,2,%SUBJ),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-ARG),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
OBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,-).


+GEN-CAT(%N,PASSIVE-ADJ-AGENT),
+ADJUNCT(%N,%Adjuncts),
in_set(%Adj,%Adjuncts), @agent_prep(%Adj),
-PTYPE(%Adj,%%), -OBJ(%Adj,%%), PRED(%Adj,%%),
SUBJ(%N,%SUBJ),
arg(%N,1,NULL), arg(%N,2,%SUBJ),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-ARG),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
PRED(%LoSUBJ,pro), PRON-FORM(%LoSUBJ,sie), NUM(%LoSUBJ,sg),
PERS(%LoSUBJ,3), GEND(%LoSUBJ,neut),
OBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,-).

+GEN-CAT(%N,PASSIVE-ADJ-AGENT),
+ADJUNCT(%N,%Adjuncts),
in_set(%Adj,%Adjuncts), @agent_prep(%Adj),
PRED(%Adj,%%),PTYPE(%Adj,%%), arg(%Adj,%%,%LoSUBJ),
OBJ(%Adj,%LoSUBJ), +OBJ-TH(%N,%OBJTh), arg(%N,2,%OBJTh),
SUBJ(%N,%Subj),nonarg(%N,1,%Subj),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-ARG),
```

```
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
PASSIVE(%N,-).

+GEN-CAT(%N,PASSIVE-1-ARG),
SUBJ(%N,%SUBJ),
arg(%N,1,NULL), arg(%N,2,%SUBJ),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-ARG-GENERIC),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
PRED(%LoSUBJ,pro), PRON-FORM(%LoSUBJ,man), NUM(%LoSUBJ,sg),
PERS(%LoSUBJ,3), OBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,-).

+GEN-CAT(%N,PASSIVE-1-ARG),
SUBJ(%N,%SUBJ), nonarg(%N,1,%SUBJ),
arg(%N,1,NULL),
+OBL(%N,%OBL), +OBJ(%OBL,%OblObj), +arg(%N,2,%OblObj),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-ARG-GENERIC),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
PRED(%LoSUBJ,pro), PRON-FORM(%LoSUBJ,man), NUM(%LoSUBJ,sg),
PERS(%LoSUBJ,3), PASSIVE(%N,-).


+GEN-CAT(%N,PASSIVE-1-ARG),
SUBJ(%N,%SUBJ), nonarg(%N,1,%SUBJ),
arg(%N,1,NULL),
+OBJ-TH(%N,%OBJTH), +arg(%N,2,%OBJTH),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-ARG-GENERIC),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
PRED(%LoSUBJ,pro), PRON-FORM(%LoSUBJ,man), NUM(%LoSUBJ,sg),
PERS(%LoSUBJ,3),  PASSIVE(%N,-).

+GEN-CAT(%N,PASSIVE-1-EMBEDDED),
SUBJ(%N,%SUBJ),
arg(%N,1,NULL), arg(%N,2,%SUBJ),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-EMBPRONAGENT),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
PRED(%LoSUBJ,pro), PRON-FORM(%LoSUBJ,sie), NUM(%LoSUBJ,sg),
```

```
PERS(%LoSUBJ,3), GEND(%LoSUBJ,masc),
OBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,-).

+GEN-CAT(%N,PASSIVE-1-EMBEDDED),
+OBJ-TH(%N,%OBJTh), +arg(%N,2,%OBJTh),
arg(%N,1,NULL),
SUBJ(%N,%Subj),nonarg(%N,1,%Subj),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-EMBPRONAGENT),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
PRED(%LoSUBJ,pro), PRON-FORM(%LoSUBJ,sie), NUM(%LoSUBJ,sg),
PERS(%LoSUBJ,3), GEND(%LoSUBJ,masc), PASSIVE(%N,-).

+GEN-CAT(%N,PASSIVE-AGENTCOORD),
SUBJ(%N,%SUBJ),
arg(%N,1,NULL), arg(%N,2,%SUBJ),
+in_set(%N,%CoordN),+COORD(%CoordN,+),
+in_set(%N2,%CoordN), +PASSIVE(%N2,-),
+SUBJ(%N2,%CoordSUBJ),
PASSIVE(%N,+)
==>
ALT-CAT(%N,ACTIVE-2-AGENTCOORD),
SUBJ(%N,%CoordSUBJ), arg(%N,1,%CoordSUBJ),
OBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,-).
```

A.3 Rules for mapping active to passive F-structures

```
(
  +GEN-CAT(%N,ACTIVE-2-ARG)
| +GEN-CAT(%N,ACTIVE-2-AGENTCOORD)
| +GEN-CAT(%N,ACTIVE-2-EMBPRONAGENT)
| +GEN-CAT(%N,ACTIVE-2-ARG-GENERIC)
| +GEN-CAT(%N,ACTIVE-2-NEUTPRONAGENT)
),
+in_set(%N,%CoordN),+COORD(%CoordN,+), -ADJ-REL(%%,%CoordN),
+in_set(%N2,%CoordN), {\+ %N = %N2},
+lex_id(%N,%Id1), +lex_id(%N2,%Id2), {\+ %Id1 = %Id2},
+PRED(%N2,%%), +SUBJ(%N2,%Subj),
```

```
+NUM(%Subj,%Num), +PERS(%Subj,%Pers),
SUBJ(%N,%Subj),arg(%N,1,%Subj), COORD-LEVEL(%CoordN,%%)
==>
SUBJ(%N,%NewSubj), arg(%N,1,%NewSubj),
PRED(%NewSubj,pro), PRON-FORM(%NewSubj,sie),
PRON-TYPE(%NewSubj,pers), NUM(%NewSubj,%Num), PERS(%NewSubj,%Pers).

(
  +GEN-CAT(%N,ACTIVE-2-ARG)
| +GEN-CAT(%N,ACTIVE-2-AGENTCOORD)
| +GEN-CAT(%N,ACTIVE-2-EMBPRONAGENT)
| +GEN-CAT(%N,ACTIVE-2-ARG-GENERIC)
| +GEN-CAT(%N,ACTIVE-2-NEUTPRONAGENT)
),
+in_set(%N,%CoordN),+COORD(%CoordN,+), +ADJ-REL(%%,%CoordN),
+in_set(%N2,%CoordN), {\+ %N = %N2},
+lex_id(%N,%Id1), +lex_id(%N2,%Id2), {\+ %Id1 = %Id2},
+PRED(%N2,%%), +SUBJ(%N2,%Subj), +PRON-TYPE(%Subj,rel),
+NUM(%Subj,%Num), +PERS(%Subj,%Pers), SUBJ(%N,%Subj),
arg(%N,1,%Subj), PRON-REL(%N,%Subj), COORD-LEVEL(%CoordN,%%)
==>
SUBJ(%N,%NewSubj), arg(%N,1,%NewSubj),
PRED(%NewSubj,pro), PRON-FORM(%NewSubj,die), PRON-REL(%N,%NewSubj),
PRON-TYPE(%NewSubj,rel), NUM(%NewSubj,%Num), PERS(%NewSubj,%Pers).

(
  +GEN-CAT(%N,ACTIVE-2-ARG)
| +GEN-CAT(%N,ACTIVE-2-AGENTCOORD)
| +GEN-CAT(%N,ACTIVE-2-EMBPRONAGENT)
| +GEN-CAT(%N,ACTIVE-2-ARG-GENERIC)
| +GEN-CAT(%N,ACTIVE-2-NEUTPRONAGENT)
),
+in_set(%N,%CoordN),+COORD(%CoordN,+), +ADJ-REL(%%,%CoordN),
+in_set(%N2,%CoordN), {\+ %N = %N2},
+lex_id(%N,%Id1), +lex_id(%N2,%Id2), {\+ %Id1 = %Id2},
ADJUNCT(%CoordN,%Adjunct)
==>
ADJUNCT(%N,%Adjunct).

(  +GEN-CAT(%N,ACTIVE-2-ARG-GENERIC)
 | +GEN-CAT(%N,ACTIVE-2-NEUTPRONAGENT)
),
OBJ(%N,%Obj), PRED(%Obj,pro), PRON-TYPE(%Obj,refl), arg(%N,2,%Obj),
+OBL(%N,%Obl), +OBJ(%Obl,%OblObj), arg(%N,3,%OblObj)
==> arg(%N,2,%OblObj).
```

```
(
  GEN-CAT(%N,ACTIVE-2-ARG)
| GEN-CAT(%N,ACTIVE-2-AGENTCOORD)
| GEN-CAT(%N,ACTIVE-2-EMBPRONAGENT)
),
OBJ(%N,%Obj), PRED(%Obj,pro), PRON-TYPE(%Obj,refl),
arg(%N,2,%Obj), SUBJ(%N,%Subj), arg(%N,1,%Subj)
==>
OBJ(%N,%Subj), arg(%N,2,%Subj),
SUBJ(%N,%NewSUBJ), PRED(%NewSUBJ,pro), PRON-FORM(%NewSUBJ,man),
NUM(%NewSUBJ,sg), PERS(%NewSUBJ,3), arg(%N,1,%NewSUBJ),
GEN-CAT(%N,ACTIVE-2-ARG-GENERIC) .


(
   +GEN-CAT(%N,ACTIVE-2-ARG-GENERIC)
 | +GEN-CAT(%N,ACTIVE-2-NEUTPRONAGENT)
),
OBJ(%N,%Obj), PRON-TYPE(%Obj,inh-refl_), nonarg(%N,1,%Obj),
SUBJ(%N,%Subj), arg(%N,1,%Subj),
+OBL(%N,%Obl), +OBJ(%Obl,%OblObj), arg(%N,2,%OblObj)
==>
OBJ(%N,%Subj), arg(%N,2,%Subj),
SUBJ(%N,%NewSUBJ), PRED(%NewSUBJ,pro), PRON-FORM(%NewSUBJ,man),
NUM(%NewSUBJ,sg), PERS(%NewSUBJ,3),
arg(%N,1,%NewSUBJ), arg(%N,3,%OblObj),
GEN-CAT(%N,ACTIVE-2-ARG-GENERIC) .


(
  GEN-CAT(%N,ACTIVE-2-ARG)
| GEN-CAT(%N,ACTIVE-2-AGENTCOORD)
| GEN-CAT(%N,ACTIVE-2-EMBPRONAGENT)
),
OBJ(%N,%Obj), PRON-TYPE(%Obj,inh-refl_), nonarg(%N,1,%Obj),
SUBJ(%N,%Subj), arg(%N,1,%Subj),
+OBL(%N,%Obl), +OBJ(%Obl,%OblObj), arg(%N,2,%OblObj)
==>
OBJ(%N,%Subj), arg(%N,2,%Subj),
SUBJ(%N,%NewSUBJ), PRED(%NewSUBJ,pro), PRON-FORM(%NewSUBJ,man),
NUM(%NewSUBJ,sg), PERS(%NewSUBJ,3),
arg(%N,1,%NewSUBJ), arg(%N,3,%OblObj),
GEN-CAT(%N,ACTIVE-2-ARG-GENERIC) .
```

```
+GEN-CAT(%N,ACTIVE-2-ARG),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
( OBJ(%N,%SUBJ) | COMP(%N,%SUBJ) ), arg(%N,2,%SUBJ),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-2-ARG),
OBL-AG(%N,%AG), OBJ(%AG,%LoSUBJ), PRED(%AG,von),arg(%AG,1,%LoSUBJ),
arg(%N,1,%LoSUBJ), SUBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,+).

+GEN-CAT(%N,ACTIVE-2-ARG),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
+OBL(%N,%OBL), +OBJ(%OBL,%OblObj), arg(%N,2,%OblObj),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-2-ARG),
OBL-AG(%N,%AG), OBJ(%AG,%LoSUBJ), PRED(%AG,von),
arg(%AG,1,%LoSUBJ), arg(%N,1,%LoSUBJ), SUBJ(%N,%SUBJ),
arg(%N,2,%SUBJ), PRED(%SUBJ,pro), PRON-FORM(%SUBJ,man),
NUM(%SUBJ,sg), PERS(%SUBJ,3), arg(%N,3,%OblObj),
PASSIVE(%N,+).


+GEN-CAT(%N,ACTIVE-2-ARG),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
+OBJ-TH(%N,%ObjTh), +arg(%N,2,%ObjTh),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-2-ARG),
OBL-AG(%N,%AG), OBJ(%AG,%LoSUBJ), PRED(%AG,von),
arg(%AG,1,%LoSUBJ), arg(%N,1,%LoSUBJ),
SUBJ(%N,%SUBJ), nonarg(%N,1,%SUBJ),
NUM(%SUBJ,sg), PERS(%SUBJ,3),
PASSIVE(%N,+).


+GEN-CAT(%N,ACTIVE-2-ARG-GENERIC),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
( OBJ(%N,%SUBJ) | COMP(%N,%SUBJ) ), arg(%N,2,%SUBJ),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
```

```
SUBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,+).

+GEN-CAT(%N,ACTIVE-2-ARG-GENERIC),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
+OBL(%N,%OBL), +OBJ(%OBL,%OblObj), arg(%N,2,%OblObj),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PRED(%SUBJ,pro), PRON-FORM(%SUBJ,man), NUM(%SUBJ,sg),
PERS(%SUBJ,3), arg(%N,3,%OblObj),
PASSIVE(%N,+).

+GEN-CAT(%N,ACTIVE-2-ARG-GENERIC),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
+OBJ-TH(%N,%ObjTh), +arg(%N,2,%ObjTh),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), nonarg(%N,1,%SUBJ),
NUM(%SUBJ,sg), PERS(%SUBJ,3),
PASSIVE(%N,+).


+GEN-CAT(%N,ACTIVE-2-EMBPRONAGENT),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
( OBJ(%N,%SUBJ) | COMP(%N,%SUBJ) ), arg(%N,2,%SUBJ),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,+).

+GEN-CAT(%N,ACTIVE-2-EMBPRONAGENT),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
+OBL(%N,%OBL), +OBJ(%OBL,%OblObj), arg(%N,2,%OblObj),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
```

```
PRED(%SUBJ,pro), PRON-FORM(%SUBJ,man), NUM(%SUBJ,sg),
PERS(%SUBJ,3), arg(%N,3,%OblObj),
PASSIVE(%N,+).

+GEN-CAT(%N,ACTIVE-2-EMBPRONAGENT),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
+OBJ-TH(%N,%ObjTh), +arg(%N,2,%ObjTh),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), nonarg(%N,1,%SUBJ),
NUM(%SUBJ,sg), PERS(%SUBJ,3),
PASSIVE(%N,+).

+GEN-CAT(%N,ACTIVE-2-AGENTCOORD),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
( OBJ(%N,%SUBJ) | COMP(%N,%SUBJ) ), arg(%N,2,%SUBJ),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,+).

+GEN-CAT(%N,ACTIVE-2-AGENTCOORD),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
+OBL(%N,%OBL), +OBJ(%OBL,%OblObj), arg(%N,2,%OblObj),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PRED(%SUBJ,pro), PRON-FORM(%SUBJ,man), NUM(%SUBJ,sg),
PERS(%SUBJ,3), arg(%N,3,%OblObj),
PASSIVE(%N,+).

+GEN-CAT(%N,ACTIVE-2-AGENTCOORD),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
+OBJ-TH(%N,%ObjTh), +arg(%N,2,%ObjTh),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), nonarg(%N,1,%SUBJ),
```

```
NUM(%SUBJ,sg), PERS(%SUBJ,3),
PASSIVE(%N,+).

+GEN-CAT(%N,ACTIVE-2-NEUTPRONAGENT),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
( OBJ(%N,%SUBJ) | COMP(%N,%SUBJ) ), arg(%N,2,%SUBJ),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PASSIVE(%N,+).

+GEN-CAT(%N,ACTIVE-2-NEUTPRONAGENT),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
+OBL(%N,%OBL), +OBJ(%OBL,%OblObj), arg(%N,2,%OblObj),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), arg(%N,2,%SUBJ),
PRED(%SUBJ,pro), PRON-FORM(%SUBJ,man), NUM(%SUBJ,sg),
PERS(%SUBJ,3), arg(%N,3,%OblObj),
PASSIVE(%N,+).

+GEN-CAT(%N,ACTIVE-2-NEUTPRONAGENT),
SUBJ(%N,%LoSUBJ), arg(%N,1,%LoSUBJ),
+OBJ-TH(%N,%ObjTh), +arg(%N,2,%ObjTh),
PASSIVE(%N,-)
==>
ALT-CAT(%N,PASSIVE-1-ARG),
arg(%N,1,NULL),
SUBJ(%N,%SUBJ), nonarg(%N,1,%SUBJ),
NUM(%SUBJ,sg), PERS(%SUBJ,3),
PASSIVE(%N,+).

+GEN-CAT(%N,%%), ( +SUBJ(%N,%C) | +OBJ(%N,%C) ),
CASE(%C,%%)
==> 0.

+GEN-CAT(%N,%%), ( +SUBJ(%N,%C) | +OBJ(%N,%C) ),
CHECK(%C,%%)
==> 0.

+GEN-CAT(%N,%%), ( +SUBJ(%N,%C) | +OBJ(%N,%C) ),
```

```
+COORD(%C,+), +in_set(%CONJ,%C), CHECK(%CONJ,%%)
==> 0.

+GEN-CAT(%N,%%), ( +SUBJ(%N,%C) | +OBJ(%N,%C) ),
+COORD(%C,+), +in_set(%CONJ,%C), CASE(%CONJ,%%)
==> 0.

+GEN-CAT(%N,%%), TOPIC-REL(%N,%%)
==> 0.

+GEN-CAT(%N,%%), +XCOMP(%X,%N), TOPIC-REL(%X,%%)
==> 0.

+GEN-CAT(%N,%%), OBJ(%N,%C), NUM(%C,%%), PERS(%C,%%),
+CLAUSE-TYPE(%C,decl), +COMP-FORM(%C,%%)
==>
COMP(%N,%C).

+GEN-CAT(%N,%%), OBJ(%N,%C), NUM(%C,%%), PERS(%C,%%),
+CLAUSE-TYPE(%C,int)
==>
COMP(%N,%C).

+GEN-CAT(%N,%%), +SUBJ(%N,%C),-NUM(%C,%%), -PERS(%C,%%),
+CLAUSE-TYPE(%C,decl)", +COMP-FORM(%C,%%)"
==>
NUM(%C,sg), PERS(%C,3).

+GEN-CAT(%N,%%), +XCOMP(%X,%N),
+SUBJ(%N,%S), +COMP(%N,%C), +CLAUSE-TYPE(%C,decl),
+COMP-FORM(%C,%%), SUBJ(%X,%C), nonarg(%X,1,%C)
==>
SUBJ(%X,%S), nonarg(%X,1,%S).

+GEN-CAT(%N,%%), +XCOMP(%X,%N),
+OBL-AG(%N,%O), +OBJ(%O,%LoSubj), +SUBJ(%N,%PasSubj),
SUBJ(%X,%LoSubj), nonarg(%X,1,%LoSubj)
==>
SUBJ(%X,%PasSubj), nonarg(%X,1,%PasSubj).

+GEN-CAT(%N,%%), +XCOMP(%X,%N),
+SUBJ(%N,%LoSubj),
SUBJ(%X,%PasSubj), nonarg(%X,1,%PasSubj),
{\+ %PasSubj = %LoSubj }
==>
```

```
SUBJ(%X,%LoSubj), nonarg(%X,1,%LoSubj).

+GEN-CAT(%N,PASSIVE-1-ARG), +XCOMP(%X,%N),
PRED(%X,sein), VTYPE(%X,raising), CHECK(%X,%%)
==>
PRED(%X,müssen).


+GEN-CAT(%N,%%), +OBL-AG(%N,%OBL), +OBJ(%OBL,%Ag),
+ADJUNCT(%Ag,%Adjuncts), in_set(%Adj,%Adjuncts),
+ADJUNCT-TYPE(%Adj,focus)
==>
ADJUNCT(%OBL,%NewA), in_set(%Adj,%NewA).

+GEN-CAT(%N,%%), +OBL-AG(%N,%OBL), +OBJ(%OBL,%Ag),
+NTYPE(%Ag,%T), +NSYN(%T,proper),
NUM(%Ag,%%)
?=> 0.

+GEN-CAT(%N,%%), +OBL-AG(%N,%OBL), +OBJ(%OBL,%Ag),
+COORD(%Ag,+), +in_set(%Ag1,%Ag),
+NTYPE(%Ag1,%T), +NSYN(%T,proper),
NUM(%Ag1,%%)
?=> 0.

+GEN-CAT(%N,%%), +OBL-AG(%N,%OBL), +OBJ(%OBL,%Ag),
+COORD(%Ag,+), "+in_set(%Ag1,%Ag),
+NTYPE(%Ag1,%T), +NSYN(%T,proper),"
NUM(%Ag,%%)
?=> 0.

+GEN-CAT(%N,%%), +OBL-AG(%N,%OBL), +OBJ(%OBL,%Ag),
( +NTYPE(%Ag,%T), +NSYN(%T,proper) | +PRED-RESTR(%Ag,%%)),
GEND(%Ag,%%)
?=> 0.

+GEN-CAT(%N,%%), +SUBJ(%N,%Subj),
-NUM(%Subj,%%)
?=>
NUM(%Subj,sg).

+GEN-CAT(%N,%%), +SUBJ(%N,%Subj),
PRON-FORM(%Subj,da), +PRON-TYPE(%Subj,demon)
==>
PRON-FORM(%Subj,dies).
```

```
+GEN-CAT(%N,%%), +OBJ(%N,%Subj),
NUM(%Subj,%%)
?=> 0.

+GEN-CAT(%N,%%), +TNS-ASP(%N,%TA),
PAST(%TA,%%)
?=> 0.

+GEN-CAT(%N,%%), -ALT-CAT(%N,%%)
==>
NO-ALT-CAT(%N,+).

-GEN-CAT(%%,%%)
==>
GEN-CAT(%%,FILTER).
```

Bibliography

Aissen, J. (1999). Markedness and subject choice in optimality theory. *Natural Language and Linguistic Theory*, 17(4):673–711.

Aissen, J. (2003). Differential Object Marking: Iconicity vs. Economy. *Natural Language and Linguistic Theory*, 21:435–483.

Angeli, G., Liang, P., and Klein, D. (2010). A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512, Cambridge, MA. Association for Computational Linguistics.

Appelt, D. E. (1982). *Planning natural language utterances to satisfy multiple goals*. PhD thesis, Stanford, CA, USA.

Ariel, M. (2001). Accessibility Theory: An Overview. In Sanders, T., Schliperoord, J., and Spooren, W., editors, *Text Representation*, pages 29–87. John Benjamins, Amsterdam.

Arnold, J. E., Losongco, A., Wasow, T., and Ginstrom, R. (2000). Heaviness vs. newness: The effects of structural complexity and discourse status on constituent ordering. *Language*, pages 28–55.

Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Bangalore, S. and Rambow, O. (2000a). Corpus-based lexical choice in natural language generation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 464–471. Association for Computational Linguistics.

Bangalore, S. and Rambow, O. (2000b). Exploiting a probabilistic hierarchical model for generation. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 42–48. Association for Computational Linguistics.

Banik, E. (2009). Parenthetical constructions: an argument against modularity. In *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks*, pages 46–53. Association for Computational Linguistics.

Banik, E., Gardent, C., and Kow, E. (2013). The kbgen challenge. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 94–97, Sofia, Bulgaria. Association for Computational Linguistics.

Barzilay, R. and Lapata, M. (2005). Collective content selection for concept-to-text generation. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 331–338. Association for Computational Linguistics.

Barzilay, R. and Lapata, M. (2006). Aggregation via set partitioning for natural language generation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 359–366. Association for Computational Linguistics.

Barzilay, R. and Lapata, M. (2008). Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34:1–34.

Bateman, J. and Zock, M. (2003). Natural Language Generation. In Mitkov, R., editor, *The Oxford Handbook of Computational Linguistics*. Oxford University Press.

Behaghel, O. (1909). Beziehungen zwischen umfang und reihenfolge von satzgliedern. *Indogermanische Forschungen*, 25:110–142.

Belz, A. (2005). Statistical generation: Three methods compared and evaluated. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG'05)*, pages 15–23.

Belz, A. (2008). Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455.

Belz, A. and Kow, E. (2010). The GREC Challenges 2010: overview and evaluation results. In *Proceedings of the 6th International Natural Language Generation Conference*, INLG '10, pages 219–229, Stroudsburg, PA, USA.

Belz, A., Kow, E., Viethen, J., and Gatt, A. (2008). The GREC challenge: overview and evaluation results. In *Proc. of the 5th International Natural Language Generation Conference*, INLG '08, pages 183–193, Stroudsburg, PA, USA.

Belz, A., Kow, E., Viethen, J., and Gatt, A. (2009). The grec main subject reference generation challenge 2009: overview and evaluation results. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation*, pages 79–87. Association for Computational Linguistics.

Belz, A. and Varges, S. (2007). Generation of repeated references to discourse entities. In *Proc. of the 11th European Workshop on Natural Language Generation*, ENLG '07, pages 9–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

Belz, A., White, M., Espinosa, D., Kow, E., Hogan, D., and Stent, A. (2011). The first surface realisation shared task: Overview and evaluation results. In *Proc. of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, pages 217–226, Nancy, France. Association for Computational Linguistics.

Belz, A., White, M., van Genabith, J., Hogan, D., and Stent, A. (2010). Finding common ground: Towards a surface realisation shared task. In *Proceedings of the 6th International Natural Language Generation Conference (INLG'10)*.

Bhatt, R. and Pancheva, R. (2006). Implicit arguments. In Everaert, M. and Riemsdijk, H. C. v., editors, *The Blackwell companion to syntax*, pages 554–584. Blackwell Publishing.

Bobrow, D. G., Cheslow, B., Condoravdi, C., Karttunen, L., King, T. H., Nairn, R., de Paiva, V., Price, C., and Zaenen, A. (2007). PARC's Bridge question answering system. In King, T. H. and Bender, E. M., editors, *Proceedings of the GEAF (Grammar Engineering Across Frameworks) 2007 Workshop*, pages 13–15.

Bock, K. (1987). Exploring levels of processing in sentence production. In *Natural Language Generation*, pages 351–363. Springer.

Bohnet, B. (2010). Top accuracy and fast dependency parsing is not a contradiction. In *Proc. of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China.

Bohnet, B., Björkelund, A., Kuhn, J., Seeker, W., and Zarriess, S. (2012). Generating non-projective word order in statistical linearization. In *Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 928–939, Jeju Island, Korea.

Bohnet, B., Mille, S., Favre, B., and Wanner, L. (2011). <stumaba >: From deep representation to surface. In *Proc. of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, pages 232–235, Nancy, France.

Bohnet, B., Nivre, J., Ginter, F., Haich, J., and Farkas, R. (2013). Joint morphological and syntactic analysis for richly inflected languages. In *Transactions of the Association for Computational (TACL)*.

Bohnet, B., Wanner, L., Mill, S., and Burga, A. (2010). Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer. In *Proc. of the 23rd International Conference on Computational Linguistics*, pages 98–106, Beijing, China.

Bouma, G. (2008). *Starting a sentence in Dutch. A corpus study of subject- and object-fronting*. PhD thesis.

Bouma, G. (2010). Syntactic tree queries in prolog. In *Proceedings of the Fourth Linguistic Annotation Workshop, ACL 2010*.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*.

Bresnan, J. (2001). *Lexical-functional syntax*, volume 16. Blackwell Oxford.

Bresnan, J., Cueni, A., Nikitina, T., and Baayen., H. (2007). Predicting the Dative Alternation. In Boume, G., Kraemer, I., and Zwarts, J., editors,

Cognitive Foundations of Interpretation. Amsterdam: Royal Netherlands Academy of Science.

Bresnan, J., Dingare, S., and Manning, C. D. (2001). Soft Constraints Mirror Hard Constraints: Voice and Person in English and Lummi. In *Proceedings of the LFG '01 Conference*.

Bresnan, J. and Ford, M. (2010). Predicting syntax: Processing dative constructions in american and australian varieties of english. *Language*, 86(1):168–213.

Büring, D. (1997). *The meaning of topic and focus: the 59th Street Bridge accent*, volume 3. Psychology Press.

Büring, D. (2001). What do definites do that indefinites definitely don't? *Audiatur vox sapientiae: A Festschrift for Arnim von Stechow*, 52:70.

Butt, M., Dyvik, H., King, T. H., Masuichi, H., and Rohrer, C. (2002). The parallel grammar project. In *Proceedings of the 2002 workshop on Grammar engineering and evaluation-Volume 15*, pages 1–7. Association for Computational Linguistics.

Cahill, A. (2009). Correlating human and automatic evaluation of a german surface realiser. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 97–100, Suntec, Singapore. Association for Computational Linguistics.

Cahill, A. and Forst, M. (2009). Human evaluation of a german surface realisation ranker. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 112–120, Athens, Greece. Association for Computational Linguistics.

Cahill, A., Forst, M., and Rohrer, C. (2007a). Stochastic Realisation Ranking for a Free Word Order Language. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 17–24, Saarbrücken, Germany. DFKI GmbH.

Cahill, A., III, J. T. M., Meurer, P., Rohrer, C., and Rosén, V. (2007b). Speeding up LFG Parsing using C-Structure Pruning . In *Coling 2008: Proceedings of the workshop on Grammar Engineering Across Frameworks*, pages 33 – 40.

Cahill, A. and Riester, A. (2009). Incorporating information status into generation ranking. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 817–825, Suntec, Singapore. Association for Computational Linguistics.

Cahill, A. and Van Genabith, J. (2006). Robust pcfg-based generation using automatically acquired lfg approximations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1033–1040. Association for Computational Linguistics.

Cahill, L., Doran, C., Evans, R., Mellish, C., Paiva, D., Reape, M., Scott, D., and Tipper, N. (1999). In search of a reference architecture for nlg systems. In *Proc. of the European Workshop on Natural Language Generation (EWNLG)*, pages 77–85.

Callaway, C. B. (2003). Evaluating coverage for large symbolic nlg grammars. In *IJCAI*, pages 811–816. Citeseer.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluation the role of bleu in machine translation research. In *Proeceedings of the 11th Conference of the European Chapter of the ACL (EACL 2006)*, volume 6, pages 249–256.

Chafe, W. L. (1976). Givenness, contrastiveness, definiteness, subjects, topics, and point of view in subject and topic. In Li, C. N., editor, *Subject and topic*, pages 25–55. Academic Press, New York.

Cheung, J. C. and Penn, G. (2010). Entity-based local coherence modelling using topological fields. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*. Association for Computational Linguistics.

Clarke, J. and Lapata, M. (2010). Discourse constraints for document compression. *Computational Linguistics*, 36(3):411–441.

Condoravdi, C. and Gawron, J. M. (1996). The context-dependency of implicit arguments. In Kanazawa, M., Pinon, C., and de Swart, H., editors, *Quantifiers, deduction and context*, pages 1–32. CSLI publications, Stanford.

Corston-Oliver, S., Gamon, M., Ringger, E., and Moore, R. (2002). An overview of amalgam: A machine-learned generation module. In *Proceedings of the International Natural Language Generation Conference*, pages 33–40.

Crouch, D. and King, T. H. (2006). Semantics via F-Structure Rewriting. In Butt, M. and King, T. H., editors, *Proceedings of the LFG06 Conference*.

Crouch, R., King, T. H., III, J. T. M., Riezler, S., and Zaenen, A. (2004). Exploiting F-structure Input for Sentence Condensation. In Butt, M. and King, T. H., editors, *Proceedings of the LFG04 Conference*, University of Canterbury.

Dale, R. (1989). Cooking up referring expressions. In *Proc. of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 68–75, Vancouver, British Columbia, Canada.

Dale, R. (1992). *Generating referring expressions: Constructing descriptions in a domain of objects and processes*. The MIT Press.

Dale, R. and Reiter, E. (1995). Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

Dale, R., Scott, D., and Di Eugenio, B. (1998). Introduction to the special issue on natural language generation. *Computational Linguistics*, 24(3):346–353.

Dale, R. and Viethen, J. (2009). Referring expression generation through attribute-based heuristics. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 58–65. Association for Computational Linguistics.

Danlos, L. (1984). Conceptual and linguistic decisions in generation. In *Proc. of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, pages 501–504, Stanford, California, USA.

de Kok, D. (2010). Feature selection for fluency ranking. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 155–163. Association for Computational Linguistics.

de Kok, D., Plank, B., and van Noord, G. (2011). Reversible stochastic attribute-value grammars. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 194–199, Stroudsburg, PA, USA. Association for Computational Linguistics.

De Smedt, K., Horacek, H., and Zock, M. (1996). Architectures for natural language generation: Problems and perspectives. In *Trends In Natural Language Generation: An Artifical Intelligence Perspective*, pages 17–46. Springer-Verlag.

Dethlefs, N., Cuayáhuitl, H., Hastie, H., Rieser, V., and Lemon, O. (2014). Cluster-based prediction of user ratings for stylistic surface realisation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 702–711, Gothenburg, Sweden. Association for Computational Linguistics.

DeVault, D., Traum, D., and Artstein, R. (2008). Practical grammar-based nlg from examples. In *Proceedings of the Fifth International Natural Language Generation Conference*, INLG '08, pages 77–85, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dipper, S. and Zinsmeister, H. (2009). The role of the German Vorfeld for local coherence. In Chiarcos, C., de Castilho, R. E., and Stede, M., editors, *Von der Form zur Bedeutung: Texte automatisch verarbeiten/From Form to Meaning: Processing Texts Automatically*, pages 69–79. Narr, Tübingen.

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc.

Duboue, P. A. (2002). Content planner construction via evolutionary algorithms and a corpus-based fitness function. In *Proceedings of the 2nd International Natural Language Generation Conference (INLG'02)*, pages 89–96.

Elhadad, M., Robin, J., and McKeown, K. (1997). Floating constraints in lexical choice. *Computational Linguistics*, 23(2):195–239.

Erk, K. and McCarthy, D. (2009). Graded Word Sense Assignment. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 440 – 449, Singapore.

Féry, C. (2008). Information structural notions and the fallacy of invariant correlates. *Acta Linguistica Hungarica*, 55(3):361–379.

Féry, C. and Krifka, M. (2008). Information structure. notional distinctions, ways of expression. In van Sterkenburg, P., editor, *Unity and diversity of languages*, pages 123–136. John Benjamins Publishing, Amsterdam.

Filippova, K. and Strube, M. (2007a). Generating constituent order in german clauses. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 320–327, Prague, Czech Republic. Association for Computational Linguistics.

Filippova, K. and Strube, M. (2007b). The german vorfeld and local coherence. *Journal of Logic, Language and Information*, 16:465–485.

Filippova, K. and Strube, M. (2009). Tree Linearization in English: Improving Language Model Based Approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 225–228, Boulder, Colorado. Association for Computational Linguistics.

Fillmore, C. J. (1986). Pragmatically controlled zero anaphora. In *Proceedings of the 14th annual meeting of the Berkeley Linguistics Society*, pages 35–55.

Fodor, J. A. (1983). *The modularity of mind: An essay on faculty psychology.* MIT press.

Forst, M. (2007). *Disambiguation for a Linguistically Precise German Parser.* PhD thesis, University of Stuttgart.

Frank, A., King, T. H., Kuhn, J., and Maxwell, J. T. (2001). Optimality Theory Style Constraint Ranking in Large-Scale LFG Grammars . In Sells, P., editor, *Formal and Empirical Issues in Optimality Theoretic Syntax*, page 367–397. CSLI Publications.

Frey, W. (2004). A medial topic position for German. *Linguistische Berichte*, 198:153–190.

Garoufi, K. and Koller, A. (2013). Generation of effective referring expressions in situated context. *Language and Cognitive Processes*.

Gerber, M. and Chai, J. (2010). Beyond nombank: A study of implicit arguments for nominal predicates. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592, Uppsala, Sweden.

Givón, T. (1983). *Topic continuity in discourse: A quantitative cross-language study*, volume 3. John Benjamins Publishing.

Givón, T. (1994). The pragmatics of de-transitive voice: Functional and typological aspects of inversion. In *Voice and inversion*, pages 3–44. Amsterdam: John Benjamins Publishing Company.

Goldberg, E., Driedger, N., and Kittredge, R. I. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

Greenbacker, C. F. and McCoy, K. F. (2009). Feature selection for reference generation as informed by psycholinguistic research. In *Proceedings of the CogSci 2009 Workshop on the Production of Referring Expressions: Bridging the Gap between Computational and Empirical Approaches to Reference (PRE-CogSci 2009)*, Vancouver, British Columbia, Canada.

Grosz, B. J., Joshi, A., and Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Gundel, J. K. and Fretheim, T. (2004). Topic and focus. *The handbook of pragmatics*, 175:196.

Gundel, J. K., Hedberg, N., and Zacharski, R. (1993). Cognitive status and the form of referring expressions in discourse. *Language*, pages 274–307.

Guo, Y., Wang, H., and VAN GENABITH, J. (2011). Dependency-based n-gram models for general purpose sentence realisation. *Natural Language Engineering*, 17(04):455–483.

Halliday, M. A. (1967). Notes on transitivity and theme in english. *Journal of Linguistics*, 3(01):37–81.

Henschel, R., Cheng, H., and Poesio, M. (2000). Pronominalization revisisted. In *Proceedings of the 18th Conference on Computational Linguistics (COLING 2000)*, pages 306–312, Birmingham.

Höhle, T. (1982). Explikation für 'normale betonung'und 'normale wortstellung'. *Satzglieder im Deutschen–Vorschläge zur syntaktischen, semantischen und pragmatischen Fundierung, Studien zur deutschen Grammatik*, (15):75–153.

Hovy, E. H. (1990). Pragmatics and natural language generation. *Artificial Intelligence*, 43(2):153–197.

Jacobs, J. (2001). The dimensions of topic–comment1. *Linguistics*, 39(4):641–681.

Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 217–226.

Kamp, H. and Reyle, U. (1993). *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*. Number 42. Springer.

Karamanis, N., Poesio, M., Mellish, C., and Oberlander, J. (2009). Evaluating centering for information ordering using corpora. *Computational Linguistics*, 35(1).

Keenan, E. L. and Dryer, M. S. (2007). Passive in the world's languages. In Shopen, T., editor, *Clause Types*, volume 1 of *Language Typology and Syntactic Description*, pages 224–275. Cambridge University Press.

Kelly, C., Copestake, A., and Karamanis, N. (2009). Investigating content selection for language generation using machine learning. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 130–137. Association for Computational Linguistics.

Kempen, G. and Harbusch, K. (2004). How flexible is constituent order in the midfield of german subordinate clauses? a corpus study revealing

unexpected rigidity. In *Proceedings of the International Conference on Linguistic Evidence*, pages 81–85.

Kibble, R. and Power, R. (2004). Optimizing referential coherence in text generation. *Computational Linguistics*, 30(4):401–416.

Knight, K. and Hatzivassiloglou, V. (1995). Two-level, many-paths generation. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 252–260. Association for Computational Linguistics.

Koller, A. and Petrick, R. (2011). Experiences with planning for natural language generation. *Computational Intelligence*, 27(1):23–40.

Koller, A. and Stone, M. (2007). Sentence generation as planning. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 336–343, Prague.

Krahmer, E. and Van Deemter, K. (2012). Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

Krifka, M. (2008). Basic notions of information structure. *Acta Linguistica Hungarica*, 55(3):243–276.

Kuno, S. (1972). Functional sentence perspective: A case study from japanese and english. *Linguistic inquiry*, pages 269–320.

Lambrecht, K. (1994). *Information Structure and Sentence Form: Topic, Focus, and the Mental Representations of Discourse Referents*. Cambridge University Press.

Langkilde, I. and Knight, K. (1998). Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 704–710. Association for Computational Linguistics.

Langkilde-Geary, I. (2002). An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proceedings of the 12th International Natural Language Generation Workshop*, pages 17–24. Citeseer.

Lenerz, J. (1977). *Zur Abfolge nominaler Satzglieder im Deutschen*, volume 5. TBL-Verlag Narr.

Levelt, W. (1989). *Speaking: From intention to articulation*, volume 1. MIT press.

Levy, R. and Jaeger, F. (2007). Speakers optimize information density through syntactic reduction. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*.

Mairesse, F., Gasic, M., Jurcicek, F., Keizer, S., Thomson, B., Yu, K., and Young, S. (2010). Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1552–1561, Uppsala, Sweden. Association for Computational Linguistics.

Marciniak, T. and Strube, M. (2005). Beyond the pipeline: discrete optimization in nlp. In *Proc. of the 9th Conference on Computational Natural Language Learning*, CONLL '05, pages 136–143, Stroudsburg, PA, USA.

McCoy, K. F. and Strube, M. (1999). Generating Anaphoric Expressions: Pronoun or Definite Description? In *The Relation of Discourse/Dialogue Structure, Proceedings of the Workshop held in conjunction with the 38th Annual Meeting of the ACL*, pages 63 – 71, College Park, USA.

McDonald, D. D. (1993). Issues in the choice of a source for natural language generation. *Computational Linguistics*, 19(1):191–197.

Mellish, C. and Dale, R. (1998). Evaluation in the context of natural language generation. *Computer Speech & Language*, 12(4):349–373.

Mellish, C., Evans, R., Cahill, L., Doran, C., Paiva, D., Reape, M., Scott, D., and Tipper, N. (2000). A representation for complex and evolving data dependencies in generation. In *Proc. of the 6th Conference on Applied Natural Language Processing*, pages 119–126, Seattle, Washington, USA.

Merlo, P., Bunt, H., and Nivre, J. (2010). Current trends in parsing technology. In Bunt, H., Merlo, P., and Nivre, J., editors, *New Trends in Parsing Technology: Dependency Parsing, Domain Adaptation and Deep Parsing*, pages 1–17. Springer.

Meteer, M. (1991). Bridging the generation gap between text planning and linguistic realization. In *Computational Intelligence*, volume 7 (4).

Morris, J. and Hirst, G. (1991). Lexical cohesion, the thesaurus, and the structure of text. *Computational Linguistics*, 17(1):21–225.

Nakanishi, H., Miyao, Y., and Tsujii, J. (2005). Probabilistic models for disambiguation of an hpsg-based chart generator. In *Proceedings of the Ninth International Workshop on Parsing Technology*, Parsing '05, pages 93–102, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nakatsu, C. and White, M. (2010). Generating with discourse combinatory categorial grammar. *Linguistic Issues in Language Technology*, 4(1).

Naumann, K. (2006). Manual for the annotation of in-document referential relations. Technical report, Seminar für Sprachwissenschaft, Abt. Computerlinguistik, Universität Tübingen.

Nivre, J. (2005). Dependency grammar and dependency parsing. Technical Report MSI 05133, Växjö University, School of Mathematics and Systems Engineering.

Oh, A. H. and Rudnicky, A. I. (2002). Stochastic natural language generation for spoken dialog systems. *Computer Speech & Language*, 16(3):387–407.

Owczarzak, K., van Genabith, J., and Way, A. (2007). Evaluating machine translation with lfg dependencies. *Machine Translation*, 21(2):95–119.

Paiva, D. S. and Evans, R. (2005). Empirically-based control of natural language generation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 58–65. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL-02*, pages 311–318, NJ, USA.

Poesio, M., Stevenson, R., di Eugenio, B., and Hitzeman, J. (2004). Centering: A parametric theory and its instantiations. *Computational Linguistics*, 30(3):309–363.

Prince, E. F. (1981). Toward a Taxonomy of Given-New Information. In Cole, P., editor, *Radical Pragmatics*, pages 233–255. Academic Press, New York.

Rajkumar, R. and White, M. (2011). Linguistically motivated complementizer choice in surface realization. In *Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop*, pages 39–44, Edinburgh, Scotland. Association for Computational Linguistics.

Ratnaparkhi, A. (2000). Trainable methods for surface natural language generation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 194–201. Association for Computational Linguistics.

Reinhart, T. (1981). Pragmatics and linguistics: An analysis of sentence topics in pragmatics and philosophy i. *Philosophica anc Studia Philosophica Gandensia Gent*, 27(1):53–94.

Reiter, E. (1994). Has a Consensus NL Generation Architecture Appeared, and is it Psycholinguistically Plausible? pages 163–170.

Reiter, E. (1995). NLG vs. Templates. In *Proceedings of the 5th European Workshop on Natural Generation*, Leiden, The Netherlands.

Reiter, E. and Belz, A. (2009). An Investigation into the Validity of Some Metrics for Automatically Evaluating Natural Language Generation Systems. *Computational Linguistics*, 35(4).

Reiter, E. and Dale, R. (1997). Building applied natural language generation systems. *Nat. Lang. Eng.*, 3(1):57–87.

Reiter, E. and Sripada, S. (2002). Should corpora texts be gold standards for nlg? In *Proceedings of INLG*, volume 2, pages 97–104.

Reiter, E., Sripada, S., Hunter, J., Yu, J., and Davy, I. (2005). Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2):137–169.

Riester, A. (2008). A semantic explication of information status and the underspecification of the recipients' knowledge. In *Proceedings of Sinn und Bedeutung*, volume 12, pages 508–522.

Riester, A., Lorenz, D., and Seemann, N. (2010). A Recursive Annotation Scheme for Referential Information Status. In Chair), N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta. European Language Resources Association (ELRA).

Riezler, S., King, T. H., Kaplan, R. M., Crouch, R., Maxwell, J. T., and Johnson, M. (2002). Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques . In *Proceedings of ACL 2002*.

Riezler, S. and Maxwell, J. (2006). Grammatical Machine Translation . In *Proceedings of HLT-NAACL'06*, New York.

Riezler, S. and Vasserman, A. (2004). Gradient feature testing and l1 regularization for maximum entropy parsing. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP'04), Barcelona, Spain*.

Ringger, E. K., Gamon, M., Moore, R. C., Rojas, D., Smets, M., and Corston-Oliver, S. (2004). Linguistically Informed Statistical Models of Constituent Structure for Ordering in Sentence Realization. In *Proceedings of the 2004 International Conference on Computational Linguistics*, Geneva, Switzerland.

Robin, J. (1993). A revision-based generation architecture for reporting facts in their historical context. In *New Concepts in Natural Language Generation: Planning, Realization and Systems. Frances Pinter, London and*, pages 238–265. Pinter Publishers.

Rohrer, C. and Forst, M. (2006). Improving Coverage and Parsing Quality of a Large-Scale LFG for German. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, Genoa, Italy.

Rooth, M. (1992). A theory of focus interpretation. *Natural language semantics*, 1(1):75–116.

Roth, M. and Frank, A. (2012). Aligning predicate argument structures in monolingual comparable texts: A new corpus for a new task. In *Proc. of*

*the 1st Joint Conference on Lexical and Computational Semantics (*SEM)*, Montreal, Canada.

Rubinoff, R. (1992). Integrating text planning and linguistic choice by annotating linguistic structures. In Dale, R., Hovy, E. H., Rösner, D., and Stock, O., editors, *NLG*, volume 587 of *Lecture Notes in Computer Science*, pages 45–56. Springer.

Ruppenhofer, J., Sporleder, C., Morante, R., Baker, C., and Palmer, M. (2010). Semeval-2010 task 10: Linking events and their participants in discourse. In *Proc. of the 5th International Workshop on Semantic Evaluation*, pages 45–50, Uppsala, Sweden.

Schwarzschild, R. (1999). Givenness, avoidf and other constraints on the placement of accent*. *Natural language semantics*, 7(2):141–177.

Scott, D. and de Souza, C. S. (1990). Getting the message across in rst-based text generation. *Current research in natural language generation*, 4:47–73.

Seeker, W. and Kuhn, J. (2012). Making Ellipses Explicit in Dependency Conversion for a German Treebank. In *Proc. of the 8th conference on International Language Resources and Evaluation*, Istanbul, Turkey.

Sgall, P., Hajicová, E., and Panevová, J. (1986). *The meaning of the sentence in its semantic and pragmatic aspects*. Springer.

Shibatani, M. (1985). Passives and related constructions: A prototype analysis. *Language*, 61:821–848.

Siddharthan, A. and Copestake, A. (2004). Generating referring expressions in open domains. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 407–414, Barcelona, Spain.

Siddharthan, A. and Katsos, N. (2012). Offline sentence processing measures for testing readability with users. In *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, pages 17–24. Association for Computational Linguistics.

Siddharthan, A., Nenkova, A., and McKeown, K. (2011). Information status distinctions and referring expressions: An empirical study of references to people in news summaries. *Computational Linguistics*, 37(4):811–842.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

Speyer, A. (2005). Competing constraints on vorfeldbesetzung in german. In *Proceedings of the Constraints in Discourse Workshop*, pages 79–87.

Stede, M. (1996). Lexical paraphrases in multilingual sentence generation. *Machine Translation*, 11(1-3):75–107.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). brat: a web-based tool for nlp-assisted text annotation. In *Proc. of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France.

Stent, A., Marge, M., and Singhai, M. (2005). Evaluating evaluation methods for generation in the presense of variation. In *Proceedings of CICLING*, pages 341–351.

Stent, A., Prasad, R., and Walker, M. (2004). Trainable sentence planning for complex information presentation in spoken dialog systems. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 79. Association for Computational Linguistics.

Stone, M., Doran, C., Webber, B., Bleam, T., and Palmer, M. (2003). Microplanning with communicative intentions: The spud system. *Computational Intelligence*, 19(4):311–381.

Talmy, L. (1985). Lexicalization patterns: Semantic structure in lexical forms. *Language typology and syntactic description*, 3:57–149.

Telljohann, H., Hinrichs, E., Kübler, S., and Zinsmeister, H. (2006). Stylebook for the tübingen treebank of written german (tüba-d/z). revised version. Technical report, Seminar für Sprachwissenschaft, Universität Tübingen.

Thompson, S. A. (1987). The passive in english: A discourse perspective. In *In Honor of Ilse Lehiste*, pages 497–511.

Tomlin, R. (1983). On the interaction of syntactic subject, thematic information, and agent in english. *Journal of Pragmatics*, 7(4):411–432.

Uszkoreit, H. (1987). *Word order and constituent structure in German*, volume 8. Center for the Study of Language and Information.

Vallduví, E. (1993). *The informational component*. PhD thesis.

Vallduví, E. and Engdahl, E. (1996). The linguistic realization of information packaging. *Linguistics*, 34(3):459–520.

Velldal, E. (2008). *Empirical Realization Ranking*. PhD thesis, University of Oslo, Department of Informatics.

Velldal, E. and Oepen, S. (2005). Maximum entropy models for realization ranking. In *Proceedings of the 10th Machine Translation Summit*, pages 109–116, Thailand.

Velldal, E. and Oepen, S. (2006). Statistical ranking in tactical generation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia.

Viethen, J. and Dale, R. (2006). Algorithms for generating referring expressions: do they do what people do? In *Proceedings of the Fourth International Natural Language Generation Conference*, pages 63–70. Association for Computational Linguistics.

Viethen, J. and Dale, R. (2010). Speaker-dependent variation in content selection for referring expression generation. In *Proceedings of the 8th Australasian Language Technology Workshop*, pages 81–89.

Vonk, W., Hustinx, L., and Simons, W. (1992). The use of referential expressions in structuring discourse. *Language and Cognitive Processes*, 7(3-4):301–333.

Walker, M. A., Rambow, O., and Rogati, M. (2001). Spot: A trainable sentence planner. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

Walker, M. A., Stent, A., Mairesse, F., and Prasad, R. (2007). Individual and domain adaptation in sentence planning for dialogue. *J. Artif. Intell. Res.(JAIR)*, 30:413–456.

Wan, S., Dras, M., Dale, R., and Paris, C. (2009). Improving grammaticality in statistical sentence generation: Introducing a dependency spanning tree algorithm with an argument satisfaction model. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 852–860. Association for Computational Linguistics.

Wanner, L. (1994). Building another bridge over the generation gap. In *Proc. of the 7th International Workshop on Natural Language Generation*, INLG '94, pages 137–144, Stroudsburg, PA, USA.

Wanner, L., Mille, S., and Bohnet, B. (2012). Towards a surface realization-oriented corpus annotation. In *Proceedings of the Seventh International Natural Language Generation Conference*, INLG '12, pages 22–30, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ward, G. and Birner, B. (2004). Information structure and non-canonical syntax. In *The handbook of pragmatics*, pages 153–174. Blackwell, Oxford, UK.

Ward, G. L. (1988). *The semantics and pragmatics of preposing*. Garland New York.

Wasow, T. (1997). Remarks on grammatical weight. *Language variation and change*, 9(01):81–105.

Weber, A. and Müller, K. (2004). Word order variation in german main clauses: A corpus analysis. In *Proceedings of the 20th International conference on Computational Linguistics*, pages 71–77.

White, M. (2004). Reining in ccg chart realization. In *Natural Language Generation*, pages 182–191. Springer.

White, M. and Rajkumar, R. (2009). Perceptron reranking for ccg realization. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 410–419. Association for Computational Linguistics.

White, M., Rajkumar, R., and Martin, S. (2007). Towards broad coverage surface realization with ccg. In *Proceedings of the Workshop on Using Corpora for NLG: Language Generation and Machine Translation (UCNLG+ MT)*, pages 267–276.

Wong, Y. W. and Mooney, R. (2007). Generation by inverting a semantic parser that uses statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 172–179, Rochester, New York. Association for Computational Linguistics.

Zarrieß, S. (2009). Developing german semantics on the basis of parallel lfg grammars. In *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks (GEAF 2009)*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

Zarrieß, S., Cahill, A., and Kuhn, J. (2011). Underspecifying and predicting voice for surface realisation ranking. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1007–1017, Portland, Oregon, USA. Association for Computational Linguistics.

Zarrieß, S., Cahill, A., and Kuhn, J. (2012). To what extent does sentence-internal realisation reflect discourse context? A study on word order. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 767–776, Avignon, France. Association for Computational Linguistics.

Zarrieß, S. and Kuhn, J. (2010). Reversing f-structure rewriting for generation from meaning representations. In *Proceedings of the LFG10 Conference*, Ottawa, Canada.

Zarrieß, S. and Kuhn, J. (2013). Combining surface realisation and referring expression generation: A corpus-based investigation of architectures. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Sofia, Bulgaria. Association for Computational Linguistics.

Zhong, H. and Stent, A. (2005). Building surface realizers automatically from corpora. In *Proceedings of UCNLG'05*, pages 49–54.