Information Extraction for the Geospatial Domain

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

Vorgelegt von

André Blessing

aus Esslingen am Neckar

Hauptberichter: Prof. Dr. Hinrich Schütze
Mitberichter: Prof. Dr. Ulrich Heid
Mitberichter: Prof. Dr. Jonas Kuhn

Tag der mündlichen Prüfung: 08. Juli 2014

Institut für Maschinelle Sprachverarbeitung (IMS)
Universität Stuttgart

2014

Deutsche Zusammenfassung

Geographisches Wissen ist ein zunehmend wichtiger Bestandteil in Anwendungssoftware. Gründe dafür sind zum Beispiel die weite Verbreitung von mobilen Endgeräten und die Wichtigkeit ortsbasierter Suche im World Wide Web. Umgebungsmodelle helfen, geographisches Wissen digital abzubilden und verfügbar zu machen. Georäumliche Objekte lassen sich nicht in beliebiger Skalierbarkeit mit physikalischen Sensoren erfassen, daher ist das Erfassen und Nachführen von Daten eines Umgebungsmodells kostspielig. Textuelle Daten sind eine alternative Quelle für georäumliches Wissen, da sehr viele ortsbasierte Informationen sprachlich beschrieben sind. Diese Dissertation beschäftigt sich mit der Fragestellung, wie georäumliches Wissen aus Texten mittels computerlinguistischer Verfahren extrahiert werden kann. Dazu wurden Ansätze entwickelt, die als Prototypen implementiert und evaluiert wurden.

Eine erste Herausforderung ergibt sich bereits bei der Erkennung von georäumlichen Entitäten (sog. Toponymen) in natürlichsprachigen Texten. Im ersten Schritt werden mögliche Kandidaten im Text markiert. Ortslexika (Gazetteers) bieten eine Grundlage für diesen Prozess, müssen aber durch Methoden aus der maschinellen Sprachverarbeitung erweitert werden, damit flektierte Vorkommen ebenfalls gefunden werden (z.B. Stuttgarts zum Lexikoneintrag Stuttgart). Diese Markierung ist nicht hinreichend, da viele Einträge in solchen Lexika mehrdeutig sind: London kann zum Beispiel eine Stadt oder ein Nachname sein. Diese Mehrdeutigkeit wir als geo/non-geo bezeichnet. Darüber hinaus gibt es ebenfalls geo/geo-Mehrdeutigkeiten wie zum Beispiel Fulda (Stadt) im Gegensatz zu Fulda (Fluss). In den vorgestellten Experimenten haben wir das Problem von Mehrdeutigkeiten mit Hilfe eines eigens erstellten Datensatzes adressiert. Dazu wurden Nennungen von Straßennamen manuell annotiert. Der Datensatz wurde aufgeteilt in Trainings- und Testdaten, um überwachte Lernverfahren zur Toponym-Erkennung zu entwickeln und zu evaluieren. Die Ergebnisse unserer Methoden zeigen, dass geo/non-geo-Mehrdeutigkeiten

erfolgreich aufgelöst werden. Für die Auflösung von Mehrdeutigkeiten im geo/geo-Bereich wird eine Erweiterung des Ansatzes benötigt, der die Toponyme mit den zugehörigen Realwelt-Objekten verlinkt. In dieser Arbeit werden die hierfür nötigen Erweiterungen sowohl formal als auch in prototypischen Anwendungen, z.B. beim Erkennen von umgangssprachlich benannten Regionen (z.B. Altstadt, Bankenviertel), präsentiert.

Da Trainingsdaten eine notwendige Voraussetzung für die verwendeten überwachten Ansätze sind, erschwert das Fehlen annotierter Trainingsdaten die Entwicklung erheblich. Im zweiten Teil der Arbeit wird gezeigt, wie das zeitintensive und teure Verfahren der manuellen Annotation durch ein automatisches Verfahren ersetzt werden kann. Dazu werden bereits vorhandene maschinenlesbare Datensätze (z.B. aus Wikipedia-Infoboxen) verwendet, um Text zu annotieren. Dieses Verfahren wird Distant Supervision (DS) genannt und ist bereits für englische Daten etabliert. In dieser Arbeit werden in erster Linie DS-Ansätze an der deutschen Sprache erforscht, welche im Bezug auf Distant Supervision noch Neuland ist. Das Deutsche verfügt im Gegensatz zum Englischen über eine reichhaltigere Morphologie und freiere Wortstellung, was Distant Supervision erschwert. Unser angepasster DS-Ansatz wurde implementiert und in verschiedenen Szenarien evaluiert. Diese fokusieren sich auf die Extraktion von binären Relationen zwischen georäumlichen Entitäten. Beispiele hierfür sind Relationen zwischen Orstteilen und den Gemeinden, zu denen sie gehören (Bad Cannstatt - Stuttgart) oder zwischen Städten und Flüssen, die durch sie hindurchfließen (Stuttgart - Neckar). Zur Evaluation wurden jeweils zwei unterschiedliche Relationsextraktionssysteme gegenüber gestellt. Ein System wurde mit DS-annotierten Trainingsdaten trainiert und das andere mit manuell annotierten Trainingsdaten. Die Ergebnisse zeigen, dass durch den DS-Ansatz eine gleichwertige Qualität bei der Extraktion mit deutlich weniger manuellem Aufwand erreicht werden kann.

Eine Einschränkung für Distant Supervision ist in bisherigen Ansätzen, dass

sowohl die strukturierten Daten (Wissensdatenbank) als auch die textlichen Daten in der gleichen Sprache vorliegen müssen. Dies ist für Sprachen mit geringerer Verbreitung als dem Englischen nicht gewährleistet. Daher haben wir einen Ansatz entwickelt, den wir Crosslingual Distant Supervision nennen, und der ohne diese Einschränkung auskommt. In unseren Experimenten konnten wir belegen, dass strukturierte deutsche Daten auf englische, französische und chinesische Texte erfolgreich angewendet werden können. Ein weiterer Vergleich zu einem üblichen Standardverfahren, das auf maschineller Übersetzung aufbaut, zeigt, dass das neue Verfahren deutlich bessere Ergebnisse liefert.

Abstract

Geospatial knowledge is increasingly becoming an essential part of software applications. This is primarily due to the importance of mobile devices and of location-based queries on the World Wide Web. Context models are one way to disseminate geospatial data in a digital and machine-readable representation. One key challenge involves acquiring and updating such data, since physical sensors cannot be used to collect such data on a large scale. Doing the required manual work is very time-consuming and expensive. Alternatively, a lot of geospatial data already exists in a textual representation, and this can instead be used. The question is how to extract such information from texts in order to integrate it into context models. In this thesis we tackle this issue and provide new approaches which were implemented as prototypes and evaluated.

The first challenge in analyzing geospatial data in texts is identifying geospatial entities, which are also called toponyms. Such an approach can be divided into several steps. The first step marks possible candidates in the text, which is called spotting. Gazetteers are the key component for that task but they have to be augmented by linguistically motivated methods to enable the spotting of inflected names. A second step is needed, since the spotting process cannot resolve ambiguous entities. For instance, London can be a city or a surname; we call this a geo/non-geo ambiguity. There are also geo/geo ambiguities, e.g. Fulda (city) vs. Fulda (river). For our experiments, we prepared a new dataset that contains mentions of street names. Each mention was manually annotated and one part of the data was used to develop methods for toponym recognition and the remaining part was used to evaluate performance. The results showed that machine learning based classifiers perform well for resolving the geo/non-geo ambiguity. To tackle the geo/geo ambiguity we have to ground toponyms by finding the corresponding real world objects. In this work we present such approaches in a formal description and in a (partial) prototypical implementation, e.g., the recognition of vernacular named regions (like old town or financial district).

The lack of annotated data in the geospatial domain is a major obstacle for the development of supervised extraction approaches. The second part of this thesis thus focuses on approaches that enable the automatic annotation of textual data, which we call unstructured data, by using machine-readable data from a knowledge base, which we call structured data. This approach is an instance of distant supervision (DS). It is well established for the English language. We apply this approach to German data which is more challenging than English, since German provides a richer morphology and its word order is more variable than that of English. Our approach takes these requirements into account. We evaluated our approach in several scenarios, which involve of the extraction of relations between geospatial entities (e.g., between cities and their suburbs or between towns and their corresponding rivers). For our evaluation, we developed two different relation extraction systems. One is a DS-based system, which uses the automatically annotated training set and the other one is a standard system, which uses the manually annotated training set. The comparison of the systems showed that both reach the same quality, which is evidence that DS can replace manual annotations.

One drawback of current DS approaches is that both structured data and unstructured data must be represented in the same language. However, most knowledge bases are represented in the English language, which prevents the development of DS for other languages. We developed an approach called Crosslingual Distant Supervision (CDS) that eliminates this restriction. Our experiments showed that structured data from a German knowledge base can successfully be transferred by CDS into other languages (English, French, and Chinese).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis methods are introduced to extract information from textual resources, meant for assisting context-aware systems. The goal is to transfer textual knowledge into machine-readable knowledge. For this we have to identify geospatial entities in textual content and to extract information which relates to these entities. However, the field of processing textually represented geospatial data is not very well explored yet, as can be seen from the lack of annotated geospatial data and the absence of shared tasks in this domain. Even so, the growing popularity of Web 2.0 and mash-up services based on maps shows how important geospatial data are and how important it is to make such data machine-readable.

The task of transferring knowledge from a textual representation to a machine-readable representation can be divided into several subtasks. In the first part of this thesis, we successively introduce these tasks, starting by *recognizing geospatial terms* in textual content. Therefore, we will present several methods that make it possible to identify, resolve and process such data correctly. All these methods require very specific annotated data which are not available in large scale or for the requested granularities. To compensate this lack of annotations we present in the second part of this thesis a paradigm which is called *distant supervision* (DS). DS allows the annotation of data without the need of human annotators. Resources

which are annotated by DS are helpful to evaluate our extraction methods and and they are essential if supervised methods have to be trained. We will show that DS can be applied to the geospatial domain and that it is also possible to create DS-annotations on different granularity levels. Finally, we contribute an approach that enables DS for a crosslingual setting which was evaluated on German, French, English and Chinese.

1.1 Motivation

We present our motivation from two contrary perspectives. On the one hand, the use-case perspective motivates the extraction of geospatial data by Natural Language Processing (NLP) methods and defines the requirements. On the other hand, the NLP perspective motivates the development of new methods and shows the limitations of state of the art approaches.

1.1.1 Use-case Perspective

We chose context-aware systems[1] as a use-case to find out what is required for NLP extraction methods. Context-aware systems have become a normal part of our lives. Applications (Apps) on cell phones and services like Google Maps are well-known examples of context-aware systems. They assist us to find restaurants in the surrounding area or they advise us which route to take in case of traffic jams. Such systems require a model, the context model, in particular the environment of the users, in a digital form. Context models are intended to mirror the state of the environment as closely as possible. This poses the challenge of how to update the context model when the environment changes. Many changes are amenable to detection by sensors, e.g., temperature and lighting conditions. Many informa-

[1]Context-aware systems are also named location-based services. We use the NEXUS [Dürr et al., 2004] context-aware system as our reference system. NEXUS was developed in a large collaborative research project at the University of Stuttgart over a period of more than 10 years.

tion which are needed in a context model can not be acquired by such physical sensors. Such data have to be manually processed which is expensive. As further consequence such data can not be have short-term updates.

In addition to that, natural language can be a good source for context relevant information, since language is the main medium of communication between people. The success story of the World Wide Web boosts this since electronic text is easily accessible. For example, online newspapers alert readers to planned road constructions that slow down traffic. Such information can be used to improve the quality of routing systems.

We assume that information derived from natural language using NLP can close the gap between immediate sensorial updates and long-term manual updates of context models via manually prepared data. The challenge is to efficiently extract such information in a form that allows it be used by context-aware applications for updating context models. In this thesis, we show how context models can be supported by extracting linguistically derived information.

1.1.2 NLP Perspective

Extracting information from text by a machine is one long-term goal of NLP. In recent years there has been a paradigm shift from rule-based approaches in information extraction (IE) to approaches that use statistical methods. [Etzioni et al., 2011] presented two statements that will be considered in this thesis:

- paradigm shift: from retrieval to reading
- micro (sentence) vs macro (100 million sentences)

Following these statements, we divided our motivation into three parts:

Information Extraction instead of Information Retrieval

The paradigm shift from retrieval to reading will have as a consequence that the current interfaces of search engines (e.g. Google[2]) will change to more language based interfaces (e.g. Powerset[3]). In the future, the user wants to get specific answers to his requests instead of large sets of documents, especially, when mobile devices without space for large result lists are used. The challenge is to extract only the information needed for each given request which matches our goal in supporting context aware systems. The only difference is that in our context-aware scenario the extraction has to satisfy the requirements of machines rather than humans.

Linguistic Features instead of bag of words

The second statement by Etzioni focuses on how textual corpora should be taken into account. Typical retrieval-based systems only consider complete documents and collect the words of each document (which is called "bag of words"). Therefore, the impact of a single sentence is small, since similar information is often repeated in a redundant way. Such redundancy can be well exploited in unsupervised methods by using statistical measurements that are based on word frequency and cooccurences. But not all kinds of information are distributed in the same high density. If we take very specific domains into account, then it is less likely that one piece of information is presented in several ways. In this context, the semantics of a single sentence has more impact than if the statement is redundantly represented in several sentences. That is the reason why a deep analysis of single sentences becomes more important. In these cases linguistically motivated measurements or

[2]http:/www.google.de
[3]http://en.wikipedia.org/wiki/Powerset_(company)

features are needed to increase the performance. The methods developed in this thesis will work on sentence level to satisfy this requirement.

Distant supervision instead of supervised learning

Today, machine learning is the central method to achieve high-quality information extraction. In the early 1980s, rule-based approaches were used for this task. However, rule writing needs a lot of domain knowledge from experts and a given rule-based system can only be applied to a given domain. That is the reason why now mostly machine learning approaches are used. But many approaches that are currently available highly depend on already existing data sets that include a manually annotated part for training. If no such annotated resources are available, however, supervised approaches are not possible. The manual creation of such resources to fit various domains and different aspects on a large scale is impracticable. In this thesis we will discuss how existing knowledge bases can be used to create annotated resources with minimal manual effort.

1.2 Contributions

Our contributions can be split into two branches. First, we discuss and develop methods for recognizing geospatial information in natural language texts. Our main contribution is the consideration of granularity. Our methods should allow us to distinguish between different geospatial types (e.g. streets vs. towns vs. rivers). Second, we provide methods and results in the field of relation extraction. Again, we consider fine-grained relations which have not been considered in recent extraction tasks. To overcome the problem of having to manually generate expensive training sets, we provide methods that are based on distant supervision to create annotated resources automatically. This leads to the main contributions of this dissertation: Distant supervision for German data and for crosslingual data. Figure 1.1 depicts the discussed topics of this thesis. The main part focuses on dif-

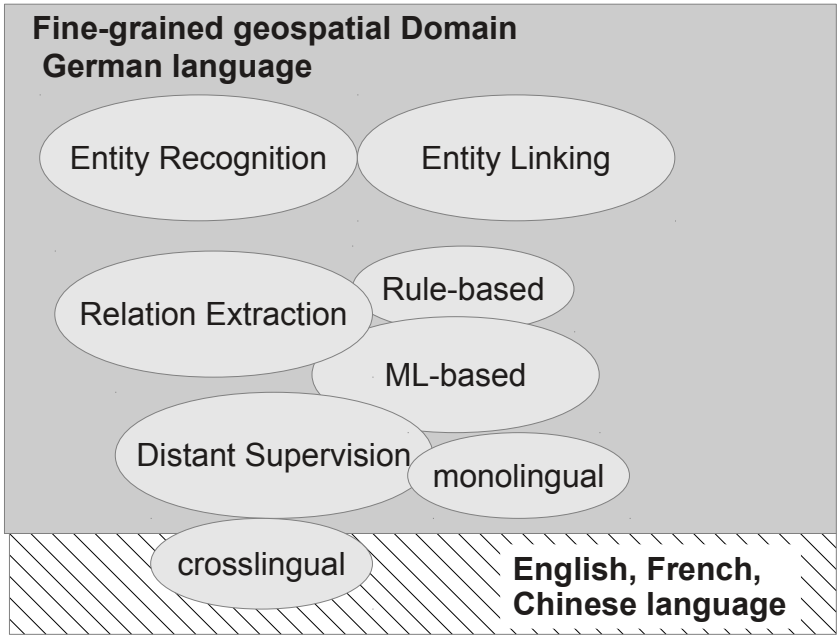

Figure 1.1: Overview of the topics treated in this thesis.

ferent methods to extract information from German texts in the geospatial domain. In the last part we extend our methods to support different languages.

1.2.1 Fine-grained geospatial entity processing

Recognizing named entities has a long tradition in NLP research. Our contribution is to investigate novel methods which allow the processing of entities on a finer granularity. Furthermore we present methods to classify the entities and to link the them to their real world references that are given by gazetteers. In chapter 4 we provide an application to detect and visualize vernacular regions which are based on this approach. We present a study that shows how attributes from different research fields that use different metrics can be combined to improve performance. Record linkage for geospatial data is mainly done from only one direction. We introduce a method that also considers linguistically motivated attributes like mor-

pheme similarity and string-based similarity metrics. Finally, we look at additional constraints to improve the quality of the mapping.

1.2.2 Distant Supervision for fine-grained relation extraction and crosslingual settings

We will describe a new approach to develop and implement an end-to-end system to extract fine-grained geospatial relations by using a supervised machine learning approach without expensive manual labeling. Using DS-annotation, systems can be rapidly developed and adapted for new relations without expensive manual annotation. In previous studies DS was only used on English data. Our result will show that DS can be successfully transferred to German. This includes the use of DS in a fine-grained setting. Finally, our relation extraction approach will show that linguistically motivated features improve the quality of the extraction results.

In chapter 5 we extend the idea of DS to crosslingual distant supervision (CDS). In this scenario it is not necessary that the knowledge base (KB) which is needed for the DS annotation is represented in the same language as the text that has to be annotated. We will show that CDS has two benefits: We can reach (i) higher accuracy of automatic annotation due to crosslingual filtering and (ii) better coverage because automatic labels can be transferred from the KB-language to a target language.

1.3 Outline

In chapter 2 we give an overview of basic techniques that are relevant in this thesis by introducing concepts of machine learning and defining terms in the field of geospatial information extraction.

Chapter 3 has two parts. First, we introduce essential resources that are used in this thesis. The chapter introduces the resources by explaining why they are

important. Wikipedia has had an important impact on our research and we explain why. The second part of this chapter depicts our implementation framework that is based on the UIMA framework. The goal of our implementation is to provide a flexible processing pipeline that allows a fast adaptation during the development of new methods for information extraction systems.

In chapter 4 we present several tasks that are required to process textual data including geospatial content. We show how the basic techniques which were introduced in chapter 2 can be adapted to new domains and to texts in German.

Chapter 5 will show how we modify distant supervision such that it can be applied to geospatial data on a fine-grained level. Additionally, we present our approach for German data.

Chapter 6 extends the distant supervision approach to a crosslingual setting. The experiments show that this approach can be applied to English, French, German and Chinese.

In chapter 7 we give a summary of the results and insights of this thesis.

Chapter 2

Background

In this chapter, we give an overview of basic techniques that are relevant in this thesis. First, we introduce concepts of machine learning. Later on, we define essential terms, especially the term *toponym*, in the field of information extraction and give a brief overview of methods used in this field.

2.1 Background in Machine Learning

Machines are brilliant if they are used for tasks that involve a lot of calculation or that involve many repetitions, like in a production chain of a factory. Programs define the behavior of machines. They are developed for deterministic situations, but often it is not feasible that a program covers all possible situations. Assume, for example, that in a factory chain some conditions have changed (e.g., extreme heat) and the sensors provide measurements that are out of range with respect to the measurements expected by the program. In such situations additional information is needed to adapt the program to the new conditions. Machine learning (ML) is an approach to close this gap by adapting programs to new conditions.

A more sophisticated definition of ML compares it to learning processes of children. Experience plays a central role for children. Every moment they

gain more experience either by observing the world or by interacting with the world. If we transfer this to ML, a machine has to gather experience to find better solutions for its tasks. Mitchell [Mitchell, 1997] uses similar terms to define ML:

Learning = Improving (I) with experience (E) at some task (T)

In theory, we can try to memorize all experience and use this knowledge to improve our program. But in practice this will lead to huge search spaces (e.g., all possible permutations of the facts) which cannot be computed in an adequate manner. One challenge for ML techniques is to find a good trade-off between gathering knowledge and pruning to reach manageable search spaces.

We give a more specific example by considering a common task in NLP: tagging. The goal of tagging is to predict the word category[1] corresponding to a given word. It is obvious that we cannot model all possible input sequences. However, a rule-based system performs very well for this task, but nevertheless there are many cases where such systems make wrong decisions. Adding more rules to the system can be seen as a kind of a never-ending Sisyphean task. ML approaches provide an elegant way to overcome this, but they are also far from providing the perfect solution. The challenge is to find the approach that fits best. This involves the task to find the best representation and modeling of the problem. The term "feature" describes characteristic measurements that can be used by the ML approach to make the right decisions. The dependencies between features and the distribution of the feature space are important aspects for choosing the best ML methods. Feature design and engineering do not play a key role in this thesis. That means that we integrate established features into our approaches if they are existing. We are more interested in how the *experience step* can be done. The question is how to gain experience. In the next sections we will explore several paradigms that consider the experience step in different ways. We start by presenting the supervised and

[1] This word category is often called "part of speech".

unsupervised paradigms which are quite complementary in the manner of gaining experience. Afterwards, we will present the semi-supervised paradigm which is a mixture of the previously introduced paradigms. Finally, we present a novel way to induce new experience that has reached good quality without any need for human advice. It is called *distant supervision*.

2.1.1 Supervised vs. Unsupervised Learning

One of the main challenging aspects of ML is how to gain experience. The most obvious way is used by the supervised learning paradigm where each observation is labeled[2]. A supervised system [Russell and Norvig, 1995, chapter 18.2] needs observations (data) and a teacher (annotations or feedback), while an unsupervised system [Russell and Norvig, 1995, chapter 20.3] needs only observations (data) to learn. Teacher in this context is an abstract term that can point to many different realizations. Annotated data can be seen as a resource that was set up by a teacher. The choice of the data which have to be labeled is mostly done randomly. But there is also a large research field that considers how to select the most informative instances for the labeling process in order to minimize the labeling costs.

Each of these two paradigms have their fields of application. Supervised systems lead to better models in most cases, but on the other hand the costs for annotations are very high. Large-scale manual annotation are indeed not feasible[3].

Unsupervised machine learning avoids annotated data by searching patterns in the data. Supervised methods deliver well-defined output data, since the annotated data provides all class labels. In the unsupervised setting the interpretation of output is not straightforward, since the found patterns that are learned by the unsupervised methods may not relate to labels which are used by humans. This fact is a main drawback of unsupervised learning since the semantics of the results

[2]This is also called “annotation” or “rating”.

[3]Crowd sourcing is one approach to solve this issue but it has also limitations with respect to its scalability.

may not fit the requirements of the task.

2.1.2 Semi-supervised Learning

Semi-supervised learning [Abney, 2007] is a paradigm that combines advantages from the supervised and unsupervised learning paradigms. It has the advantage of well defined output[4], which is given by manual annotations, and the advantage of considering large amounts of data. Semi-supervised learning works iteratively by annotating data and gathering new knowledge from the context of these annotations to update the model. This approach is also called "bootstrapping". A small set of good examples, patterns or rules is given to the system, which tries to find other such examples, patterns or rules.

The iterative processing can lead to non-controllable behavior which is often called schema-drift. It means that the newly integrated data in the learning set may not perfectly match the originally defined class criteria. This uncertainty can lead to a drift in each iteration: the probability increases that the learned classifier is heading in an unwanted direction which will reinforce this misbehavior in each following iteration.

2.1.3 Distant Supervision

We introduce distant supervision (DS) [Snow et al., 2004, Mintz et al., 2009] in more detail in comparison to the other three paradigms. This has two reasons: first, it is a relatively novel paradigm and not as well-known as the other paradigms, and second, a main part of this thesis is the further development of DS. At first sight, semi-supervised learning seems to be the paradigm most similar to distant supervision, since it also tries to annotate large data sets without handcrafted work. However, DS, in contrast to semi-supervised learning, does not use instances in the training set that are derived from previous classifications. Instead, only instances

[4]The labels of the results have a clear semantics.

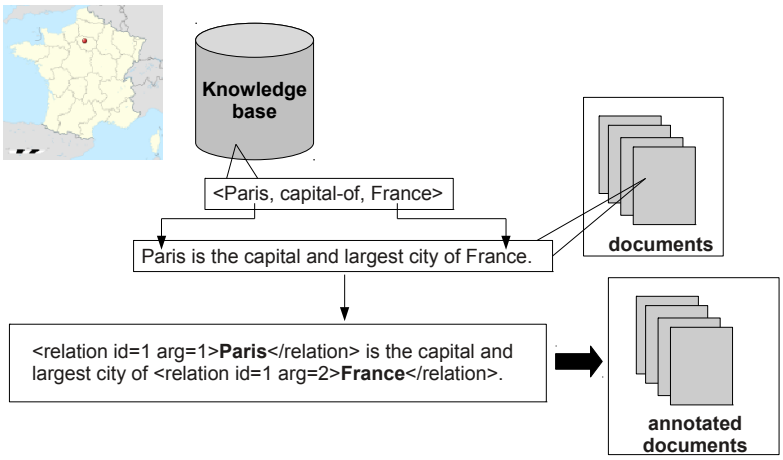

Figure 2.1: Automatic annotation in distant supervision

of high quality are used (no uncertainty, established or proven knowledge) that are derived from knowledge bases. The basic idea behind DS is more to emulate the traditional annotation process which is used in supervised learning by substituting human annotators by machine annotators. Since machines do not have the same experience as human annotators and since they cannot access the same knowledge sources, machine-readable knowledge bases (KBs) are the key resources needed for this approach. Finally, distant supervision uses these existing KBs to annotate data which can be used to train a supervised model on that annotated data. In section 3.1.3 we will further introduce such machine-readable KBs.

Figure 2.1 shows an example of annotating the binary relation between countries and their capital cities by using the distant supervision approach. An instance of the relation – <*Paris*, CAPITAL-OF, *France*> – is retrieved from the KB. Next, unstructured data referring to the arguments of the relation are identified in raw text and the text is annotated (shown as markup) with the instance from the KB. We call this annotation an automatically generated label or *automatic label*. A corpus annotated in this way is then used to train a classifier.

2.1.4 Summary of Machine Learning Approaches

Table 2.1 summarizes the four ML paradigms previously discussed.

paradigm	advantages	disadvantages
supervised	high accuracy	expensive expert/domain knowledge needed adaptation requires re-annotation
unsupervised	no manual annotation large data sets easy adaptation	lower accuracy unknown semantics
semi-supervised	manually defined seeds easy adaptation	schema-drift
distant supervision	high accuracy no manual annotation easy adaptation	integration of KB

Table 2.1: Summary of the different machine learning paradigms.

The paradigm that involves annotated data either by manual or automatic annotating reaches the best accuracies. In this thesis we first (chapter 4) focus on using the supervised paradigm. However, the effort to create the required labeled training data is high. As a solution we will show (chapters 5 and 6) that the use of the distant supervision paradigm can reach similar performance as the supervised paradigm.

2.1.5 Classification

In the previous section we introduced some foundations of machine learning. In this section we provide an overview of one application of supervised machine learning: classification. For this thesis, classification is a basic ML-method since many natural language processing (NLP) questions can be formulated as a classification problem, e.g., to decide to which word category a given word in a sentence belongs; or to decide if two given entities belong to a relation or not.

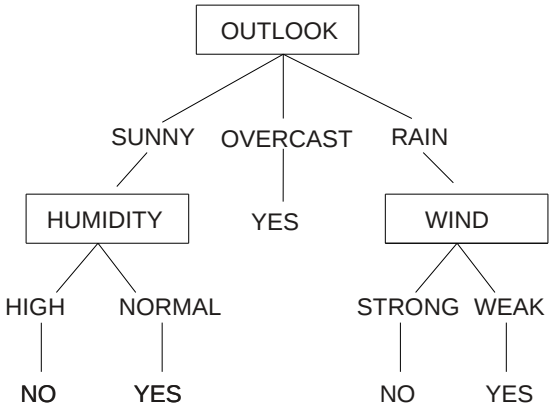

Figure 2.2: A decision tree for the playTennis decision. This example is taken from [Mitchell, 1997].

Decision Trees

The decision tree algorithm belongs to the first generation of supervised ML algorithms. Figure 2.2 presents a simple decision tree to decide if the conditions are good to play tennis. The most valuable attribute for this decision is the outlook attribute. E.g., if the outlook variable is assigned to OVERCAST, then the decision is already made by YES. All other cases need more attributes to find the best decision. Quinlan introduced the ID3 and C4.5 algorithms [Quinlan, 1986] for inducing such trees as classification models. One challenge in modeling decision trees is to avoid over-fitting, which means that the decision tree only memorizes the training data without any generalization. One solution is to include pruning that helps to restrict the size and depth of the tree. Pruning is also a way to increase the robustness against noisy training data. An advantage of decision trees is that they can be easily visualized. This helps to understand and to interpret how the used attributes are integrated into the learned model. We also exploit this advantage in Section 4.3.4.

Maximum Entropy

If we formalize the classification task as a probability distribution that takes two variables: x for the outcome class and y for the context, then we can use the principle of maximum entropy to estimate the correct distribution of p(x,y) by using some supervised observations. Later on, the estimated distribution can be used to classify new instances. Dependencies between features have no negative impact on maximum entropy models, which is an important advantage with respect to other classification algorithms (e.g., Naive Bayes). So the framework allows to integrate information from many heterogeneous information sources for classification. The data for a classification problem is described as a (potentially large) number of features. These features can be quite complex and allow the experimenter to make use of prior knowledge about what types of information are expected to be important for classification. This algorithm is mostly used in this thesis; in addition to already mentioned advantages we use it for a pragmatic reason: because it easily fits into our pipeline[5].

Support Vector Machines

A support vector machine (SVM) is another supervised learning technique from the field of machine learning. The support vector network implements the following idea [Cortes and Vapnik, 1995, p. 274]: "Map the input vectors into a very high dimensional feature space Z through some non-linear mapping chosen a priori. In this space a linear decision surface is constructed with special properties that ensure high generalization ability of the network." Figure 2.3 shows an example of two classes which are separated by SVM (solid line). It is obvious that there are many ways to separate the two classes (e.g., consider the dashed lines) but the solid line is the only separation that maximizes the margins to both classes.

One important difference to the other algorithms is that SVM-based classification

[5]The pipeline will be explained in section 3.2.1.

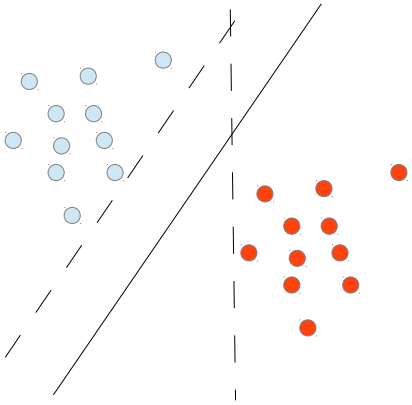

Figure 2.3: SVM calculates the best separation (solid line) of two classes.

cannot deliver confidence as probability values. This means that the distance to the hyperplane cannot be transferred to a probabilistic value.

2.2 Background in Natural Language Processing

Information Extraction (IE) has been quite popular since the Message Understanding Conferences (MUC) [Grishman and Sundheim, 1996] in the late 1980s and 1990s. Before we go into more details we define some terminology.

2.2.1 Terminology

Named Entity

A named entity is an instance of a certain entity class such as *people* or *companies*. For example, *Angela Merkel* is an instance of the entity class *person*. Named Entity Recognition (NER) is the task to identify named entities in texts and to assign them to entity classes. Typical classes are *person (PER)*, *organization (ORG)* and *location*

(LOC) or *geo-political entity (GPE)*.

Toponym

A *toponym* is a special named entity which denotes a geospatial object. The class toponym is a subset of the class of *geo-political entities*. Two classes of toponyms can be distinguished: The first are natural ones like mountains (*Mount Everest*) or rivers (*Rhine*). The second are artificial ones such as settlements (*Berlin*) or countries (*France*). For our goal to support context-aware systems we have a special interest in recognizing toponyms in texts. Chapter 4 presents different methods for toponym recognition, classification and mapping.

Relation

Entities can build relations with other entities. For example, the entity pair between *Stuttgart* and *Germany* is an instance of the binary relation CITY_IN_COUNTRY. Relations are not restricted to two arguments. Each n-ary combination is possible like the following 3-ary relation example: WINS_NOBELPRICE_FOR_INYEAR (*Barack Obama*, *Peace*, *2009*).

2.2.2 Introduction to Information Extraction

The message understanding conferences (MUC) had the most impact at the beginning of information extraction research. We give a short overview of the history of these conferences. MUC-1 was the first step towards IE without having any common data formats and evaluation. This changed in MUC-2 where the task was to extract information for a given template that contained 10 slots. Templates are used to describe the required information that has to be extracted. Each slot in a template defines one fact, e.g.: birth place of a person. Figure 2.4 shows a template for organizations which has 5 slots. The counts of correctly and wrongly filled slots

are used to calculate precision and recall. In the following conferences, the task became more complex by including more slots (18 in MUC-3 and 24 in MUC-24). In MUC-5, additional domains were considered and Japanese was used as further language besides English. But the task got too complex, especially by using hierarchical templates. Therefore, for MUC-6 several the task were split in several subtasks. The Named entity recognition (NER) task was one of these.

```
We are striving to have a strong renewed creative partnership with
Coca-Cola,‘‘ Mr. Dooner says. However, odds, of that happening are
slim since word from Coke headquarters in Atlanta is that...
```

$<ORGANIZATION - 9402240133 - 5>$:=
ORG_NAME : Coca-Cola
ORG_ALIAS : Coke
ORG_TYPE : COMPANY
ORG_LOCALE : Atlanta CITY
$ORG\_COUNTRY$: United States

Figure 2.4: Example from MUC-7 for the slot filling task. The given text snippet from document 9402240133 is used to fill the five slots in the organization template.

At MUC-7 in 1997, most systems for NER were rule-based and led to good results, but setting these systems up was expensive. Moreover they work only on special domains, adaptation to other domains involves a lot of work. To solve this problem a number of methods from machine learning [Florian et al., 2003] have been applied in the field of NER in the last ten years. Nadeau and Sekine [Nadeau and Sekine, 2007] give a good overview of the approaches and systems for NER.

2.2.3 Named Entity Recognition beyond English

Research in Named Entity Recognition up to now has mainly been concerned with English. But it is necessary to take into consideration the special features of languages other than English.

English:	I	like	London	.
	SUBJ	V	OBJ	
German:	London	finde	ich	toll .
	OBJ	V	SUBJ	

Table 2.2: Example for free constituent order in German.

We propose a design for a system which satisfies the harder requirements of German. Many aspects of current models for English NER do not apply to German because German has a different linguistic structure. In German, not only proper names, but all nouns start with capital letters and therefore the amount of potential proper names increases dramatically and so does the number of cases which show ambiguity between geospatial proper names and common nouns. In addition to that, the local contexts have more variance than the English ones (rigid syntax: subject-verb-object) because German is a language with relatively free constituent order. If, in English, a capitalized word in subject position occurs before a communication verb, the word must be a proper name (person or organization, see figure 2.4). This is not true for German, since the finite verb can take three different positions in the sentence (first, second, last) and the subject does not have to be in front position.

Faruqui and Padó developed a NER model [Faruqui and Padó, 2010] for German that is based on the Stanford Named Entity Recognizer. A weakness of this system is that it cannot differentiate between fine-grained categories. If we consider the recognition of geospatial entities, then the coverage of this approach is low.

2.2.4 Named Entity Recognition for Toponyms

One challenge in processing geospatial data is that there are many different ways to represent these data. Communication by means of natural language generally uses symbolic coordinates (e.g., Stuttgart, Germany) to determine spatial information. In contrast, spatial information in computational modeling is represented by means

```
Paris
Paris, France
capital of France
city of light
48.8567, 2.3508 (WGS84)
31U 452380 5411729 (UTM)
```

Figure 2.5: Different representations of the toponym Paris. The first four lines represent symbolic coordinates that point to the entity Paris. The last two lines represent geometric coordinates to point to Paris.

of geometric coordinates (e.g., WGS84[6]). Figure 2.5 enumerates the possible coordinate codings that point to Paris. The mapping between the different geometric representations can simply be done by defined conversion matrices. But if we go back to the symbolically-based representation then we see that in this case no defined conversion exists. But to enable geospatial data processing that is informed by data extracted from textual data, an interpretation and mapping is mandatory. This task can be divided into several steps. The first one is very similar to named entity recognition which we described before. We call this step toponym recognition.

We explain the difference between NER and toponym recognition by an example: If you are supposed to meet someone in a foreign city, it will not be enough to mention the name of the city, but you will need further details on the location, like names and numbers of the streets and the places. Current NER systems do not fully meet these requirements because they are rarely able to recognize geospatial entities below the granularity level of cities. But the recognition of toponyms is only the first step for extracting geospatial information. In the second step, it is necessary to find the real-word object which corresponds to a toponym. This is called toponym resolution or grounding [Leidner et al., 2003]. The main challenge for toponym resolution is to identify the appropriate item for a given toponym, in case there are more than one items of the same name.

In chapter 4, we go into more detail and present different approaches to process

[6]World Geodetic System 1984 [National Imagery and Mapping Agency, 2000]

toponyms that are represented in textual context.

2.2.5 Relation Extraction

The previously introduced tasks for recognizing entities are not sufficient if we aim to extract real information from textual content. We need more information about the entities, which can also be toponyms. Relations to other entities provide such information. The extraction of relations was established by the ACE shared tasks in 2004 [Doddington et al., 2004]. In theory, a relation can be established between an arbitrary amount of entities. In practice, relations are mostly defined between two entities, which is called a binary relation. Most prominent examples are: PART_OF, BORN_IN, MEMBER_OF, ACQUIRED_BY. In the beginning of relation extraction, rule-based approaches were adequate to solve this task. In the last years, more general approaches which are based on machine learning (ML) have been established. They are applicable to a wider range of relations and text sources. In the next paragraphs, we give a short introduction to the different paradigms that are used for relation extraction.

Rule-based approach

Rule-based approaches need a high amount of work for rule engineering by domain experts. Such systems reach high quality by high precision but they are restricted to small domains. We will present such a system which can be used to extract relations from a traffic news corpora. The adaptation of rules to new domains leads to writing new rules. Unsupervised learning systems also use rule or pattern-based approaches. The idea is that such patterns can be extracted from large corpora in an iterative way. At the beginning a small set of rules are defined and the system looks for similar patterns in the corpus to successively increase the number of applicable patterns.

Pattern-based approach

Pattern-based approaches are similar to rule-based approaches. The goal is to find structured parts which can be used as discriminative features. Pattern can be built on several levels. The simplest way is to use string-based pattern. In section 5.1, we present several systems that are pattern-based.

Feature-based approach

Machine learning techniques are an alternative to rule-based approaches which need no hand-crafted rule set. The disadvantage is that such systems are often not easy to understand, which makes it harder to change the model. Feature engineering is also a very time-consuming task but some basic features which are investigated in several works [Zhou et al., 2005] can be used.

2.3 Summary and Discussion

This chapter gave an overview of basic techniques that are relevant for this thesis. Therefore, we introduced machine learning methods and gave a short introduction to information extraction. The brief survey of relation extraction systems showed the paradigm shift from rule-based over pattern-based systems which are semi-supervisedly trained to mostly unsupervised systems that act on large-scale data. One challenge is to add more quality to the systems and to enable them to extract specific relations on a fine-grained level in various languages. We will face these challenges in the following chapters.

Chapter 3

Resources and Implementation

This thesis is about the exploitation of different resources to extract information from textual sources which can support applications like context-aware systems. For this we need a processing architecture which can be seen as a large pipeline of sequential processing tasks. Our architecture is based on modules which can be combined easily to fulfill the requirements of the experiments which will be presented in the following chapters.

3.1 Resources

The number of resources which are freely available has dramatically increased over the last years. This opens many new opportunities to exploit them in a scientific context. One of our goals is to show how resources can be integrated into the process of creating new methods in the field of information extraction. The resources can be used as new domain data or as knowledge bases which can be an integral part of the new methods. Even more, we will introduce methods that help to aggregate different kinds of resources to create augmented new resources that can in turn be used either to train supervised methods or for evaluation tasks.

In this thesis, we make a difference between two types of resources: *structured*

structured	unstructured
Relational Databases	Text
Digital maps	Images
Freebase	Audio Data
DBpedia	Video Data

Table 3.1: Comparison of structured and unstructured resources.

resources and *unstructured* resources. Structured resources are resources that are directly machine readable like knowledge bases that contain fields which have defined semantics. Unstructured resources, on the other hand, are not directly readable for machines, e.g. textual resources. For machines, textual resources only consist of a sequence of meaningless characters (see table 3.1 for some examples of both categories). However, some resources cannot be assigned exactly to either one of these categories, since they consist of parts which are directly machine readable and parts which are not. Wikipedia is a good example for this. As Wikipedia is important for the work reported in this thesis it will be described in a separate subsection (3.1.2).

The acquisition and maintenance of structured data is labor intensive. This often leads to such structured data being incomplete. infrequent updates or incompleteness of such structured data. That raises the challenge to provide methods to extract information from unstructured resources to close this gap; especially, if we consider context-aware systems that need a rich knowledge about the world in their context model. Methods to keep the context model up-to-date are essential for context-aware systems and cannot be reached by only using physical sensors (Section 1.1.1). In chapter 4 we will provide appropriate methods to reach this goal.

The next section introduces some resources that are used in this thesis.

3.1.1 Textual Resources

We consider textual data as a data resource which is *unstructured*. The number of digital texts has dramatically increased of the last years. One reason for this is the World Wide Web which provides a large amount of textual data in various forms. At the same time, this data also presents a lot of new challenges: it seems that there are no limits on formats or writing styles (including colloquial style). Our goal is not to focus on this aspect: as we are only interested in well-written texts, we concentrate on Wikipedia as a resource from the World Wide Web. In the next section, we will give more reasons for the use of Wikipedia.

From the perspective of context-aware systems not only national and international newspapers are relevant but also regional and local ones, such as, for example, the official journals of municipalities. This is why we also include a German newswire corpus[1] in our studies.

3.1.2 Wikipedia

Wikipedia is a large and still growing collaborative encyclopedia that fulfills all corpus requirements for our tasks by providing *unstructured* as well as *structured* information. The unstructured part (the raw text of Wikipedia articles) is central in Wikipedia, since humans are the target audience and for them, texts are easier to produce and to interpret. The structured data are used to augment the article by additional concepts. So, for example, important concepts or entities are hyperlinked to the corresponding articles which the reader can use to find more information. Also, each article in Wikipedia is assigned to several categories which are shown at the end of each article. The most important additional structured information in Wikipedia are *infoboxes*.[2] They are displayed on the right panel at the bottom of an article and they are used to give a short list of the facts of the given article.

[1] Stuttgart newspaper corpus which contains textual content of the Stuttgarter Zeitung (years 1993/1994).

[2] The term *infobox* is short of *infobox template*.

Figure 3.1 shows the Wikipedia article about *Esslingen am Neckar* from the English edition. The infobox shows some attributes of the city, for example, location, size, administration, etc.

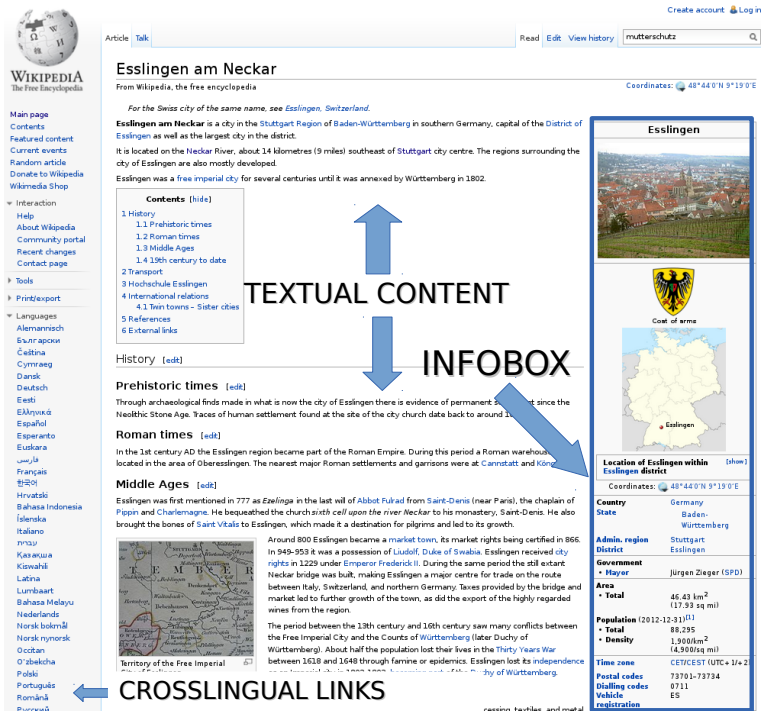

Figure 3.1: A Wikipedia article consists of unstructured information (textual data) and structured information (infobox).

For our research, both, structured and unstructured data from Wikipedia are a substantial resource because both contain a lot of knowledge and cover a broad range of domains. However, structured data is not as well covered as unstructured (textual) data since not all concepts and entities provide infobox data. On the other hand, it is also a good resource for many domains. Furthermore Wikipedia provides language editions in more than 50 languages which makes it possible to carry out research that is not restricted to the English language only.

See Medelyan et al. [Medelyan et al., 2009] for a survey about the impacts and challenges of Wikipedia to NLP, IR and IE research. For our thesis, the IE perspective is the most relevant one. We follow the idea of Wu [Wu et al., 2008] who combines infobox data and textual data to generate missing infobox content. We

will show (chapter 5) how this can be used to automatically generate annotated training corpora, which enables us to train extraction systems with less manual effort.

3.1.3 Wiki-derived Knowledge Bases

Wiki-derived KBs are a convenient way to use *structured* data of Wikipedia. Most structured data of Wikipedia are represented in templates. For our purposes, the infobox templates are the most important resource. The facts and relations of the article's headword are represented as attribute-value pairs in infoboxes. One problem is how to process template data. The Wikipedia syntax has primarily been for an easy visualization of the content and not to represent formally content. Furthermore, Wikipedia has the goal being a collaborative resource. That means that the interface and the syntax must be simple to enable an easy access. On the other hand, this can be an issue since the same content can be provided in various representations. For that reason we declare infobox data often as semi-structured data since they have no common semantics. Therefore, some processing steps are necessary that include some heuristics to extract certain structured data.

To avoid this step, Wiki-derived KBs like DBpedia [Auer et al., 2007], Yago [Suchanek et al., 2007] and Freebase [Bollacker et al., 2008] can be used. They developed their own methods to extract structured knowledge from Wikipedia. In this work we will use these Wiki-derived KBs as a baseline. But we will show that for fine-grained data this solution is often not sufficient.

We will show in 5.4.1 that more extensive mining of the contents of infoboxes yields good results – especially for complex relations: results comparable in quality cannot be achieved using facts from Wiki-derived KBs only.

3.1.4 Geonames

For geospatial data, Gazetteers are the most important resource. Like a telephone book for people, gazetteers list objects in the geospatial domain. The community-based Geonames project[3] has produced a huge freely available gazetteer. It contains more than 7 million geospatial entities. Each entity is represented by its names[4], its type (e.g.: city, country, mountain, building) and its coordinates. The project started by importing data from different freely available sources. We conducted experiments which showed that the quality of the data depends on the source used. Some of the coordinates are represented in coarse-grain grid based manner which is not an appropriate representation. In chapter 4.3 we will deal with the question of how to map these data to another geospatial data set.

The simplified representation that uses only one point in the WGS84 coordinate system as a representation of each geospatial entity leads to the problem that no spatial extensions can be modeled. Consequently, there is no simple and reliable way of computing a part-of relationship between geospatial objects. For example, if we look at two adjacent cities like *Stuttgart* and *Esslingen* in *Germany*, we do not get any information about their spatial extension. This becomes an even bigger problem if we take a look at *Untertürkheim* which is a suburb of *Stuttgart*. The assumption that *Untertürkheim* belongs to the closest town is in this case wrong since the distance to *Esslingen* is shorter than that to *Stuttgart*. However, Geonames used this metric to calculate the part-of relationship between towns and suburbs which led to noisy results in the data set. In Section 5.5 we contribute a method to correct such false SUBURB-TOWN relations in the Geonames data set by extracting the correct information from textual content of Wikipedia.

[3] http://www.geonames.org

[4] Most of the alternative names of a geospatial object are translations into different languages and transliterations.

3.1.5 NAVTEQ Data

We use the NAVTEQ data set[5] as an example of data from a commercial source. NAVTEQ is one of the main providers of maps for navigation systems. Services like Google Maps are also using this data set. We use this structured data as a source of street names (chapter 4.1) and for investigations on mapping between different data sets (chapter 4.3). OpenStreetMap (OSM) is a similar resource that has been established in the last years. Our approaches that require NAVTEQ data can now also be applied by using the freely available OSM data sets.

3.1.6 Summary

Table 3.2 gives a summary of the discussed resources in this section. In the next chapters we will to provide methods that integrate these resources into new information extraction methods. One key aspect of structured resources is that they are hardly ever complete and that most of them have slow update cycles, which is why IE methods are needed. On the other hand, textual resources can provide the missing information, since such data are often published before the information is integrated into the corresponding structured resource. Furthermore, such structured resources can be exploited to annotate unstructured resources. The World Wide Web and Web 2.0 provide a huge amount of unstructured textual resources. We focus our research on a German newswire corpus and the textual parts of Wikipedia.

Electronic text offers a wealth of information about geospatial data and can be used to improve the completeness and accuracy of geospatial resources (e.g., gazetteers).

[5] http://www.navteq.com

name	structured	unstructured	semi-structured
Wikipedia	X	X	X
DBpedia	X		
Yago	X		
Freebase	X		
Newswire		X	
Geonames	X		
NAVTEQ	X		
OSM	X		X

Table 3.2: Summary of the resources used in this thesis.

3.2 System Implementation in UIMA

The goal of our implementation is to develop a modular framework that can be easily divided into independent components. Component-based implementations have the advantage that different solutions for subtasks in the Natural Language Processing chains can be exchanged. E.g., there are different solutions for part-of-speech tagging. A component-based system allows the exchange of these solutions and thus helps to compare the impact of different NLP methods.

We aim to provide end-to-end solutions; our framework has to provide several automatic annotation components (sentence detection, tokenizing, etc.). There are many software solutions for linguistic annotation on different levels, but the combination is often not trivial, since the tools use different encodings or different data formats, or they run on different operating systems.

We decided to use the Unstructured Information Management Architecture (UIMA) framework [Ferrucci and Lally, 2004][Müller et al., 2008] as a common framework for our tasks. In the next section, we will give a short introduction to UIMA, before we explain our modules in more detail.

3.2.1 UIMA Pipeline

To a machine there is no difference between natural language data and unstructured data. IBM developed a middleware system called UIMA [Ferrucci and Lally, 2004] to process such data. The core of UIMA provides a data-driven framework for the development and application of NLP processing systems. UIMA provides a customized annotation scheme which is called *type system* in UIMA world. This type system is flexible and makes it possible to integrate own annotation on different layers (e.g., part of speech tags, of named entities) in the UIMA framework. Annotation components are called analysis engines in the UIMA systems. Each of these engines has to be defined by a description language. This includes the enumeration of all input and output types. This allows us to chain different engines including validation checks. One special case are *Readers* in the chain. Each chain needs exactly one reader to import data into the UIMA data format. *Readers* can be used to extract data from all kinds of sources, like plain text, databases, semi-structured data or web pages. The replaceability of the readers allows us to use the same chain of analysis engines for different sources. The most popular application in relation to UIMA is Watson [Ferrucci et al., 2010], an open domain question answering system. It gained its popularity by winning against human competitors in the US show "Jeopardy". The Watson system shows how scalable a UIMA system can be by answering questions in the range of 3-5 seconds. Next, we introduce some of our core UIMA modules which are used in this thesis.

3.2.2 Workflow

Figure 3.2 gives an overview of our implementation framework. On the left the resources described above are depicted which are used by our system. The Collection reader framework has to provide an interface for each of these resources. The type system defines the vocabulary of the representation to which the data is transferred during the reading process. Afterwards different automatic annotators

are used to augment the data on various linguistic annotation layers. We use an external relational database to store the results of each annotation step, in order to be able to reuse them later in processing.

To exploit linguistic annotations, we integrated several existing tools, which automatically annotate textual data, into our pipeline. Table 3.3 gives an overview of these tools. One challenge of using several tools is to unify the different representation formats of these tools. The integration into the UIMA framework solves this issue which will be described in Section 3.2.5. Furthermore, the annotation layers of the used tools are overlapping. Therefore, we can exchange tools and compare their impact. For instance, in Section 5.5.4, we will show that the usage of FSPar instead of OpenNLP leads to a better result.

name	description	annotation layers
IMS Tokenizer [Schmid, 2000]	Tokenizer and sentence boundary detector	sentence, token
TreeTagger [Schmid, 1995]	Part-Of-Speech tagging	token:pos, token:lemma, chunks
RFTagger [Schmid and Laws, 2008]	Part-Of-Speech tagging using a fine-grained POS tagset	token:pos
German NER [Faruqui and Padó, 2010]	German Named Entity Recognizer based on Stanford CoreNLP	named entities
Stuttgart Dependency Parser [Bohnet, 2010]	Bohnet dependency parser	token:pos, token:lemma token:morphology, dependency structure
FSPar [Schiehlen, 2003]	Cascaded finite-state parser	sentence, token, token:pos, token:lemma, token:morphology, dependency structure
OpenNLP framework	machine learning based toolkit for the processing of natural language text	sentence, token, token:pos, token:lemma

Table 3.3: External tools which are used in our processing pipeline as annotation engines.

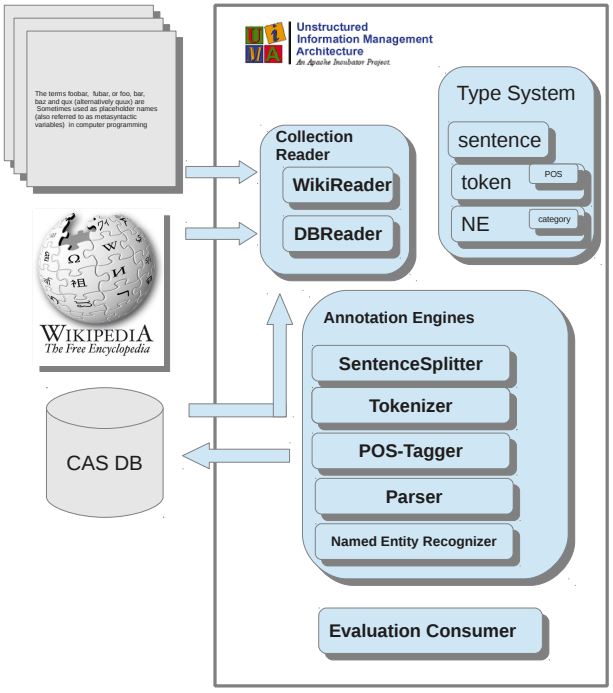

Figure 3.2: Combination of different resources and our UIMA pipeline

3.2.3 Type System

A UIMA type system specifies the annotation scheme of the data that can be manipulated by the annotator components. For linguistic analysis the types in table 3.4 are used. Documents are segmented into sentences and sentences are segmented into tokens. Each token has several attributes, like a part-of-speech tag or lemma information. Additional layers allow to annotate syntactic information like NPs or PPs. Existing structured data can also be included (such as hyperlinks from Wikipedia).

Annotation element	Attributes	Description
Sentence		complete sentence
Token		basic element in a sentence
	pos	part-of-speech Tag
	lemma	lemma
	head	reference to the head token
	deprel	edge label to the head token
	morphology	morphological analysis
NE		named entity
	class	entity type
NP		noun phrase
PP		preposition phrase
Link		Wikipedia hyperlink
	page	Wikipedia article which is referred

Table 3.4: The IMS annotation scheme (represented in the UIMA type system) includes classes on different annotation levels. Attributes are used to represent additional features in the classes.

3.2.4 Collection Reader

Plain-text Reader

The goal of our approach is to process plain text to extract information. The Plain-text reader only converts text to the internal UIMA data format without additional annotation.

FSPar Reader

FSPar [Schiehlen, 2003] is a standalone tool for linguistic analysis and annotation, up to dependency parsing. The tool includes its own tokenization which is the reason why FSPar cannot be included in chains that use a different tokenization. Results of the FSPar pipeline can be included by the FSPar reader. The reader needs two result files from FSPar as input. One provides the constituency parsing layer and the second the dependency parsing layer. The complete reader provides a linguistic analysis on different levels (tokenizer, morphology, part of speech (POS), chunking and partial dependency analysis). For this work only a few annotations

are wrapped as UIMA types (see Table 3.4): token (including lemma, POS), multi-word, sentence, NP, PP and dependency relations (labeled edges between tokens).

Wikipedia Reader

Each month, Wikipedia provides one or two data dumps for each language edition. These dumps are XML encoded and include several files which represent not only the content of the Wikipedia articles, but also metadata and structured data. The JWPL (Java Wikipedia Library) framework [Zesch et al., 2008] allows us to convert the dumps into a database format which can easily be accessed by the JWPL API. The API provides convenient methods to process a collection of Wikipedia articles. This includes plain text extraction and getting the offset addresses of additional structured data, like hyperlinks within the texts. We adapted some of the methods of the API to enable a deep analysis of template data like infoboxes, since the standard API does not accept arbitrarily nested templates. Our Wikipedia reader allows the integration of Wikipedia articles into our UIMA type system.

DB Reader

To avoid large preprocessing costs, the processed documents are stored in a relational database. Each document is represented as CAS (Common Analysis System) which is the internally used format of UIMA. One advantage is that no file system issues are involved, if a database system is used. The pipeline only needs a connection to the database and no direct file system access. That allows a distributed read and write access which is automatically managed by the database system. To integrate the data base feature into UIMA a reader function is needed which aggregates the requested documents from a database.

Gold Standard Reader

Evaluation is important for measuring the quality during the development of NLP systems. Data which includes a gold standard annotation is a good resource to evaluate methods which annotate the same data in an automatic way. The creation of gold standard data is expensive and therefore we designed and implemented a Gold Standard Reader which allows us to import existing data sets like annotated relations between named entities (cf. chapter 6).

3.2.5 Annotation Engines

Sentence Detection and Tokenization

Plain text is not segmented, i.e. there is no information about which parts of the text are sentences and which parts in the sentences are tokens. The standard OpenNLP framework provides models for English and German sentence detection and tokenization. The integration into UIMA can be done by a wrapper package which is part of the UIMA examples package. First results on German showed that the available models are not sufficient for German. For that reason we integrated a wrapper that allows us to use the prepossessing tools by Helmut Schmid [Schmid, 1995] (which are only usable at IMS). For French we used a freely available UIMA annotation engine for the segmentation. For Chinese we integrated a part of the Stanford NLP pipeline into an analysis engine to enable tokenization.

Tagging and Lemmatizing

We built a wrapper class to integrate the existing TreeTagger [Schmid, 1995] into UIMA. TreeTagger provides models for German, French and English. In parallel, we did some experiments with the OpenNLP Tagger, for which the components UIMA analysis engine is already available.

We created a wrapper around the Bohnet's parser tools[6] which provide also a functionality for lemmatization and part-of-speech tagging.

Parsing

We are using two different dependency parsers in our pipeline. The integration of the Bohnet parser is straightforward by implementing a wrapper. The advantage of the Bohnet parser is its performance in terms of speed which allows to analyze large data sets (e.g. the complete Wikipedia corpus) in short time.

FSPar is a complex parser which includes several knowledge-driven components that allows lexical distinctions on different levels. We mentioned FSPar already in the Reader section, because it is not possible to integrate this tool directly into the pipeline, as its results cannot be stored in the same document as those of the other tools. Later on, we will present how we can also compare the results. For many word classes FSPar uses its own dictionaries for morphological analysis which can be adapted and modified. In our case we did experiments to adjust the dictionaries for towns and geo-political locations.

Named Entity Recognition

Named entities play an important role in this thesis, since geospatial entities are a subset of named entities. We did some initial experiments by using a freely available NER implementation [Faruqui and Padó, 2010] but the results were not satisfying since this system is not optimized for the domain of geospatial data. In chapter 4, we will provide alternative methods to enable fine-grained toponym detection in German texts.

A more general approach to avoid error-prone NER approaches is to exploit structured data if they are available. Hyperlinks in Wikipedia texts are good candidates to automatically annotate named entities. The corresponding article behind

[6] http://code.google.com/p/mate-tools/

a hyperlink can be used to classify the type of the named entity. Categories of Wikipedia articles can be used to classify the correct type. Templates (e.g. infoboxes) of Wikipedia articles are more reliable source to classify the types, but their coverage is not as high as the coverage of categories. For instance, in the German and English Wikipedia editions each article about a person contains the template *persondata*. WE will use this kind of NER annotation in chapters 5 and 6.

Coreference Resolution

If we focus on Wikipedia, some simple heuristics can be used to solve coreference resolution. The most effective strategy is to use an approach which is similar to "one sense fits it all" that is often used in word sense disambiguation. The pronoun which has the highest frequency in the article refers to the main entity of the article topic. In Section 5.4.2 we present a small study which gives some evidence for this method.

Semantic Annotators

One advantage of UIMA is the opportunity to develop annotation modules fast and simple. Therefore we implemented several small annotators that are used to annotate parts of the unstructured text to highlight the specific types of knowledge they contain. E.g. one annotator is used to annotate subject phrases in sentences. Another one uses the Geonames data set to annotate mentions of geospatial named entities.

Machine Learning Consumer

We introduced in section 2.1.5 some theory of classification algorithms. To use such algorithms, the ClearTK [Ogren et al., 2008] framework provides interfaces to several algorithms (MaxEnt, SVM, NaiveBayes, etc.) that allow us to use them in an UIMA pipeline. Additionally, this framework provides also several methods to

extract and preprocess features. Our implementation is based on this framework, but as in some cases the provided feature extractor is not sufficient, we had to extend the framework.

One important function of the ClearTK-ML-consumer is that it operates in two modes, a learning mode and a classification mode. Both modes share the same feature processing; the only difference is that in the learning mode all instances are collected and finally transferred into the native format of the used ML algorithm, which allows to train a model. After that the classification model is persistently serialized. The classifying mode of the consumer loads such a trained model and transforms the features for each instance immediately to the native format of the ML algorithm. This procedure allows also a sequential processing by including previous decisions in the feature space.

The bottleneck of this approach is the feature extraction step. Therefore we implemented some caching strategies to store the extracted features, in order to improve the processing performance.

Evaluation Consumer

Statistical NLP involves a lot of feature engineering. In order to do that fast evaluation is important. The UIMA architecture supports evaluation, since complete modules instead of features in the pipeline can easily be exchanged which allows for a fast comparison between different setups. To enable a convenient development workflow, we introduced a special result annotation into our type system which allows us to protocol each run of the pipeline. These results are persistently stored in the UIMA representation. The evaluation component considers the results and calculates the corresponding scores. In most cases we count true positives (TP), false negatives (FN) and false positives (FP). These values are used to calculate precision, recall and the F_1-score [Manning et al., 2008].

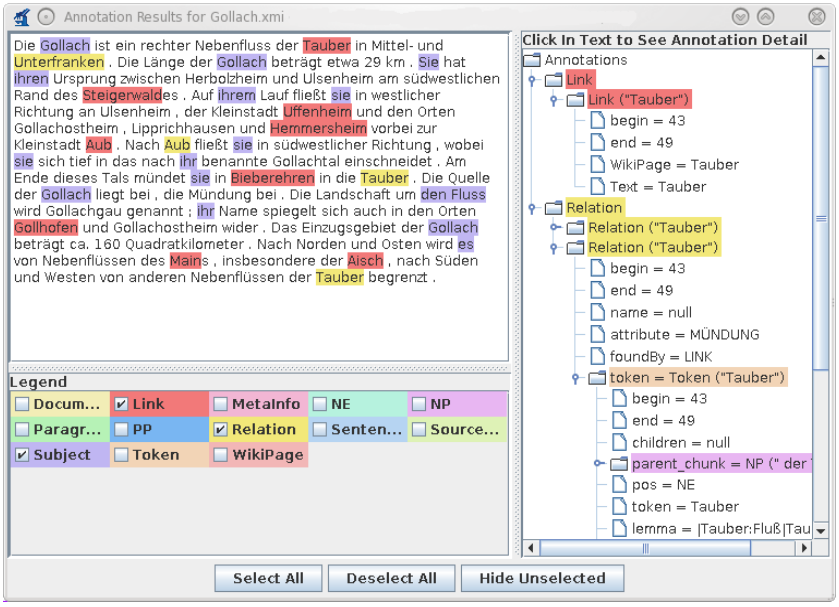

Figure 3.3: Screenshot of the UIMA AnnotationViewer.

3.2.6 Annotation Viewer

The described quantitative performance is not the only way to judge the quality of a pipeline. Inspecting individual instances is another important step in evaluating the results as well as to find new features to increase the performance. UIMA provides an integrated tool to visualize the result of an annotation chain.

Figure 3.3 shows a screenshot of the UIMA AnnotationViewer. In this example, the Wikipedia article about the river *Gollach* was linguistically annotated by our pipeline. In the legend all used annotation types are listed. In this case, three types are highlighted: (red) the type *Link* marks all hyperlinks to other Wikipedia articles; (yellow) the type *Relation* marks all entities which stand in a specific relation[7] to Gollach; (purple) the type *Subject* marks all entities, pronouns, and mentions which refer to the main entity (Gollach).

[7]In Section 5.5.4, we will present more details about these relations.

3.3 Summary and Discussion

This chapter presented essential resources and implementation concepts on which this thesis is based. Wikipedia is an ideal representative for our needs of two different kinds of resources, structured and unstructured data. In addition we presented other knowledge bases in the geospatial domain and knowledge bases which are derived from Wikipedia.

The implementation section presented the basic components of our processing pipeline which is using UIMA as basic framework. One important fact about our implementation is that our components can easily be exchanged, which helps to do different experiments with different linguistic methods.

Chapter 4

Geospatial Toponym Processing

This chapter focuses on information extraction methods for the geospatial domain. We divide geospatial toponym processing into several steps. In the first step, we restrict our research to the recognition of toponyms (4.1) in texts. One essential obstacle in geospatial NLP research is that annotated resources that allow machine-learning and evaluation are rarely available. Due to this fact, we have to create our own annotated resources for the purpose of this thesis. For the recognition of toponyms[1] in German texts, we give a detailed presentation of all necessary steps to create such an annotated resource: task definition, annotation guidelines, annotation evaluation, classifier development and evaluation.

The results of the evaluation of the toponym recognition task show that there are some limitations when each toponym is treated independently by using the local context without considering a global context. Knowledge about previously mentioned toponyms can help increase recognition performance. To reach this goal, we had to not only recognize and type the toponyms in the text, but also to resolve them to their corresponding real world toponyms. In the second step of our toponym processing (4.2) we present a solution for this task: *toponym resolution*.

In the processing of toponyms, another important aspect is the fact that differ-

[1] In this case we restrict toponyms to streets.

ent data resources or data providers often represent their data in different formats. This leads to the question of how to merge such data sets. In section 4.3 we consider this issue as an entity linkage problem. Therefore, we provide an approach that combines geometrically-motivated and linguistically-motivated features. To simplify the feature engineering process we are using visual analytics methods.

Section 4.4 of this chapter presents a small application that combines the introduced methods of this chapter for geospatial data processing.

Parts of this chapter were published in [Blessing et al., 2007, Blessing and Schütze, 2008, Sanftmann et al., 2009].

4.1 Toponym Recognition

Toponym recognition is the first step in acquiring geospatial information from textual resources. The task is twofold: (i) the textual boundaries of the toponym must be identified in its textual context (this task is also known as *toponym spotting*), and (ii) the type of the toponym must be identified (e.g. city, mountain or non-geo) [Amitay et al., 2004].

For the *toponym spotting* task, we use a dictionary, which is extracted from a gazetteer, to mark all possible toponyms in the text. These spotted entities are called *candidates*. We focus on the second task which can also be divided into two separate tasks. The first is to decide if the candidate is a toponym or not. We name this *geo* vs. *non-geo* reading. Then in the second step, if the decision is *geo*, we have to decide to which type the toponym belongs. This is in contrast with conventional named entity recognizer systems that just extract geo-political entities (GPEs) like *Germany*, *Berlin*. We are interested, instead, in more sophisticated distinctions (e.g. city vs. state).

4.1.1 Aspects of Granularity

Granularity is an important factor to defining geospatial data. The standard NER task does not distinguish different granularities when representing named entities which belong to the geo-political (GPE) domain. For real world applications, this is not sufficient and further types distinguished on the basis of finer granularity are needed. Many names represent multiple toponyms like: *Berlin* (state/city), *Dachstein* (town/mountain) and *Congo* (country/river). Sometimes, the location of two toponyms is identical (*Berlin*(city) and *Berlin*(state)), at other times it is in a subset relation (*Congo*(river) is part of *Congo*(country)); or the two items have nothing in common (*Dachstein*(town) and *Dachstein*(mountain)). In the next section we investigate street name recognition as an example of fine-grained toponym recognition.

4.1.2 Street Name Recognition

We chose street names as the primary geospatial type for our research because street names play an essential role in representing locations in textual contexts. Addresses usually contain several geospatial parts: i) a street name, ii) numbers to define the location on the street, iii) a city name and postal code to disambiguate between equally named streets in different cities and iv) the country.

Handwritten address recognition is a well researched topic [Srihari, 1993, Brakensiek and Rigoll, 2004] mainly due to its importance in minimizing human effort for postal distribution services. One important aspect in recognition is the rigid format of an address, which simplifies the recognition process significantly. But street names are not only used in the context of complete addresses. Streets are also mentioned in texts, such as newspaper articles. When street names are not represented in the rigid form of an address, then the recognition process is much more difficult. For example, events are often described without mentioning the complete address in a textual context (s. figure 4.1).

```
The parade is scheduled to start at noon at the corner of
Figueroa and 5th St. It will head south on Figueroa for one
mile and end in front of Staples Center. The parade is
scheduled to last approximately 30-45 minutes. Fans are invited to
line the sidewalks of Figueroa to watch.
```

Figure 4.1: A small text snippet that demonstrates the use of street names in a textual context. Two streets of Los Angeles, CA are mentioned in this text: Figueroa Street and 5th Street.

To find such occurrences of street names, we need recognition methods that are able to process natural language texts. We start our investigation by considering the typical morphological constituents of street names. At first glance, it seems that there are well-defined conventions for the constitution of street names. Each street name consists of two parts. The first can be seen as a designator and the second as the descriptor of the type of the street. For example in "Figueroa Street" the designator is "Figueroa" and the type of the street is "Street". Sometimes ordinal numbers are used a designators (e.g. *5th* Avenue in Manhattan) which simplify the orientation by counting the blocks. But if we focus on the naming schema outside of the US[2], we find more diversity. In languages like German this distinction is not as clear because many names are compounded to one term. Additionally, streets can be named without a street type descriptor.

We investigated the different types of street name descriptors in Baden-Württemberg (BW, a state in the south-western part of Germany) by analyzing either the last token of the street name or the head morpheme of the street name if it is a compound. Surprisingly, we found that only 50% of all different name types[3] use a descriptor (see figure 4.1). A closer look at the distribution shows that names without standard descriptors occur less frequently than names with standard descriptors. But the less frequent names have also an impact on applications, as they are often used to represent information which has not previously been mentioned.

[2]In the US most cities have a grid-based street layout.

[3]On token level each occurrence of an entity counts. On type level all entities that occur at least once have the same weight.

16,974	strasse	(street, avenue)
11,375	weg	(road)
1,750	gasse	(alley)
643	platz	(plaza)
364	gässle	(alley)
321	halde	(grade)
261	steige	(terrace)
256	allee	(alleé)
168	brücke	(bridge)

Table 4.1: Most common types in street names in German street names. We investigated all street names of Baden-Württemberg. On type level we had 65,767 different names. The common morphemes cover around 50% of the names on type level.

So, the recognition of such entities is very important to finding rare events. In contrast, frequently used names with a descriptor like (*Haupstraße*), which occur in almost every small town in BW, are frequently represented in corpora. But here the issue of ambiguity needs much more attention. The ambiguity issue will be tackled in section 4.2.

4.1.3 Task

The goal of this investigation is to recognize street names in texts. We restrict our detection to street names that occur in BW. Therefore, we create a lexicon that contains all street names from the data set of BW that we obtained from the NAVTEQ data set (see chapter 3.1.5). All street names that are missing in this data set are not relevant for our task and have no impact on the evaluation[4]. We also do not evaluate the spotting process. We have marked all names literally in the same way as found in the dictionary[5].

The next section defines the different categories that our classifier has to deter-

[4]If we use recognized toponyms in geospatial applications we need more knowledge about each mentioned toponym. For that reason we have only to deal with toponyms that are modeled in knowledge bases.

[5]Sometimes street names are abbreviated. To find such names we inserted all commonly used abbreviations into our lexicon.

mine in more detail.

4.1.4 Data and Annotation

We developed a guideline [Blessing, 2006] for the annotation procedure. In this section, we present the main parts of that guideline. The goal is to annotate possible street names on sentence level. The annotation tool includes a GUI[6] and displays exactly one instance. We define an instance as a pair of (i) a spotted name that we call *candidate* and (ii) the local context which in this case is the complete sentence around the candidate. The annotator has to decide to which category this instance belongs. There are five categories: *definite*, *general*, *functional*, *named object*, and *none*.

Before we go into more detail about the five categories, we have the following definition for what a street is in our context: **A road/way that serves mainly for the movement of vehicles (except trains, ships) or pedestrians.** Therefore railways and ship-ways are no streets in our context. Additionally, entities like the Milky Way are also no streets. Next, we present short definitions of the five categories, as well as some example sentences.

A **definite** street name refers to a specific street. Such names can be recognized through their use in addresses.

- In Rahlstedt finden sich die **Lilienc**ronstraße und der kleine, kuschelige Liliencronpark, mit einschlägigem Denkmal.
- The **Liliencron Street** and the small, cozy Liliencron park with a relevant monument are located in Rahlstedt.
- Sehenswert ist auch der **Jubiläumsplatz** im Stadtzentrum von Mettmann, der im Wandel der Zeit schon vielen Veränderungen unterlegen war.

[6]GUI = Graphical User Interface

- Worth seeing is also the **Jubilee Square** in the center of Mettmann that has seen many changes through the ages.

General street names have a larger extension and are not typically used in addresses. This includes three types of streets: (1) national roads (interstates) that are named by a letter and a number: motorway (A8), federal highways (B10), state road (L1121) and district roads (K1230); (2) Named bridges and tunnels; (3) Notified national routes whose partial sections can have other names. For example: The *Schwarzwaldhochstraße* (literally: black forest high street) is a name for such a route. It runs through several cities and can have in the respective city an additional different name (such as *Main Street*).

- Die Landesstraße **L113** verbindet Breitenwisch mit Himmelpforten und somit mit der Bundesstraße **B73** Hamburg-Stade-Cuxhaven.
- The state road **L113** relates Breitenwisch with Himmelpforten and thus with the federal highway **B73** Hamburg-Stade-Cuxhaven.
- Von der Innenstadt führen für Fußgänger die Pliensaubrücke und für Fahrzeuge die **Vogelsangbrücke** in die Pliensauvorstadt.
- From the city centre, the Pliensau bridge leads to the Pliensauvorstadt for pedestrians and the **Vogelsang Bridge** for vehicles.
- Pfullendorf liegt an der **Oberschwäbischen Barockstraße**, der Deutschen Fachwerkstraße und an der Hohenzollernstraße.
- Pfullendorf is located along the **Upper Swabian Baroque Route**, the German Fachwerkstraße and the Hohenzollern road.

Functional street names describe streets through their location, function, etc. Like **general** street names, such names are not used in an address or as a part of a normal road map. For example, a panoramic road (Panoramastrasse) describes

a street that provides a nice panoramic view. But, in contrast *Panoramastrasse* (*Panoramic Road*) can also be used as a **definite** street name, like the *Panoramastrasse* in Stuttgart, which crosses with the *Birkenwaldstraße*. This can be confusing, but a simple substitution test can be used to verify to which category a street with a functional name belongs. If you can replace the functional part with a synonym (e.g. panoramic road with scenic road) and the semantics of the sentence does not change, then the name belongs to the **functional** name category and not to the **definite** category.

- Im Winter verbindet eine 7 km lange **Eisstraße** über den zugefrorenen See den Koli mit Vuonislahti am gegenüberliegenden Ufer des Pielinen.
- In the winter, a 7 km long **ice road** over the frozen lake connects the Koli with Vuonislahti on the opposite shore of Pielinen.
- Hier führte die **Römerstrasse** zwischen Vindonissa (Windisch) und Turicum (Zürich) durch.
- Here was the **Roman road** between Vindonissa (Windisch) and Turicum (Zürich).
- Bad Teinach-Zavelstein liegt am **Ostweg**, einem Fernwanderweg des Schwarzwaldvereins, der an vielen Sehenswürdigkeiten vorbeiführt.
- Bad Teinach-Zavelstein lies on the **Eastern trail**, a walking route of the Black Forest Association, which passes many sights.
- Anbei gilt wiederum die Ausnahme, dass bestimmte, besonders aufwändige Autobahnabschnitte auch innerhalb des Großstadtbereichs Maut kosten (z.B. **Nordumgehung** von Lyon. . .).
- Here again is the exception that certain, particularly expensive motorway sections within the metropolitan area cost toll (e.g. **northern bypass** of Lyon).

A named **object** is an item that is named with a street name. These include particular names for settlements, bus/subway stops, and local branches.

- Am Rand des Limpachmooses befindet sich die Taunersiedlung **Moosgasse**, die ab dem 17. Jahrhundert aufgebaut wurde, 1836 einem Brand zum Opfer fiel und nachher wieder hergestellt wurde.
- The settlement of **Moss Street**, which was built in the 17th century, fell victim to a fire in 1836 and was subsequently restored, is located on the edge of Limpachmoos.
- In Deutschland gibt es viele größere und kleinere Wohnprojekte, z.B. die **Schellingstraße** in Tübingen, die MiKa in Karlsruhe, die SUSI auf dem Vaubangelände in Freiburg im Breisgau, der Aegidienhof in Lübeck oder die ehemalige Yorck59, jetzt am Mariannenplatz in Berlin.
- In Germany, there are many larger and smaller residential projects, for example the **Schelling Street** in Tübingen, the MiKa in Karlsruhe, the SUSI on the Vaubangelände in Freiburg im Breisgau, the Aegidienhof in Lübeck or the former Yorck59, now on the Mariannenplatz in Berlin.
- Vor der Haltestelle **Robert-Kirchhoff-Straße** trifft sie auf ein vom ehemaligen KBE-Übergabebahnhof Bonn-Bendenfeld kommendes Verbindungsgleis.
- Before the stop **Robert Kirchhoff Road** it meets a connections track which comes from the former KBE transfer station Bonn-Bendenfeld.

The last category, the **none** street name category, is assigned to all entities that do not denote streets/places in their current context. All remaining instances that are not part of one of the previous four categories are in this category. Note that all candidates are possible street names since they are spotted by using a gazetteer.

- Daher spielten die Bands in **Hinterhöfen** oder in Kellern, jedoch auch da konnten sie der Stasi nicht entgehen.

- Therefore, the bands played in **backyards** or basements, but also there, they could not escape the Stasi.
- Das Abrutschen auf **Platz** zwei war ein harter Schlag für den Bundesligisten.
- Descending to second **place** was a hard blow for the club.

Data set

Our experiments are based on a subcorpus of 15,000 annotated sentences that are extracted from two different text corpora. The first corpus is the Stuttgart Newspaper Corpus (StZ) that contains newspaper articles from the years 1993-1994. The second corpus is the German Wikipedia edition. We are the first to use the German Wikipedia edition for geospatial named entity research (the English Wikipedia was used in [Overell and Rüger, 2008]).

Due to the variance of both corpora in content, authors, wording, and grammatical styles we chose two corpora for natural language processing. The addressed challenge is to test the robustness of our methods and to achieve the highest possible coverage on different domains and styles. Moreover, the amount of available information via Wikipedia is still increasing, so it is an excellent resource for getting highly relevant information for applications that need geospatial data for context-aware systems (cf. Section 3.1.2).

The process of selecting the 15,000 sentences is carried out in several stages. First, three sets of street names are randomly chosen from the NAVTEQ street name data set. The first set contains high frequency names[7], the second set contains medium frequency names, and the last set contains low frequency names. We filter our two corpora and only keep sentences that contain at least one name of our sets.

Each sentence represents exactly one instance and contains a candidate for a street name and the textual context (the whole sentence). It is theoretically possible

[7]We had direct access to the NAVTEQ data and can count the frequency of occurrence of each street name.

that a sentence may belong to several instances if the sentence contains more than one candidate. But due to our selection method, it is unlikely that both are part of our randomly chosen instances.

Our corpus is annotated by several annotators. The inter-annotator-agreement is high: κ above 0.8 [Carletta, 1996].

4.1.5 Classification

The annotated set was split into training, development, and evaluation sets. We used an SVM-based approach (s. chapter 2.1.5 [Cortes and Vapnik, 1995]) for the classification. We trained two different models. The first one only decides if the instance is a *geo* or a *non-geo* entity. Therefore, we can use a binary classifier. In practice we aggregate all categories (definite, general, functional, and object) to one category. Our second model is more challenging because it has to distinguish between all five categories. Therefore, we used five 1-vs-all classifiers to support multi-category classification.

Table 4.2 depicts features we used for the classification and the table also includes the extracted features for the candidate *Liliencronstraße* in the sentence: “In Rahlstedt finden sich die **Liliencronstraße** und der kleine, kuschelige Liliencronpark, mit einschlägigem Denkmal. (The **Liliencron Street** and the small, cozy Liliencron park with a relevant monument are located in Rahlstedt.)”.

The features can be separated into three classes: (i) Character-based features (first two lines in tables), which are used to model the descriptors of street names and have an important role (s. table 4.1). In this approach, we avoided the usage of a morphological analyzer to get the morphological head of the names. Instead we used different character n-grams (4-7) of the word endings. (ii) Bag of words features created on word level by using different span sizes. Therefore, we used two words before and after the candidate. Additionally, we expand the bag of word

character n-grams				
	last 4 characters raße	last 5 traße	last 6 straße	last 7 nstraße
bag of words				
unigrams unigrams	next (token+1) und	token+2 der	token-1 die	token-2 sich
bigram bigram bigram bigram	token+1/token+2 und der token-2/token-1 sich die	token+2/token+3 der kleine token-3/token-2 finden sich	token+3/token+4 kleine , token-4/token-3 Rahlstedt finden	
trigram trigram trigram	token+1/token+2/token+3 und der kleine der kleine ,		token-3/token-2/token-1 finden sich die Rahlstedt finden sich	
Part of Speech Tags				
POS POS	POS+1 KON	POS+2 ART	POS-1 ART	POS-2 PRF

Table 4.2: Extracted features for the candidate *Liliencronstraße* in the sentence: In Rahlstedt finden sich die **Liliencronstraße** und der kleine, kuschelige Liliencronpark, mit einschlägigem Denkmal. (The **Liliencron Street** and the small, cozy Liliencron park with a relevant monument are located in Rahlstedt.)

features by using bigrams and trigrams. (iii) Lastly, Part of Speech tags[8] are used as features. We used a lexicon to translate between the string-based feature representation and the integer-based feature representation in SVM-light. We also encoded the window positions (relative to the token) in the feature, which distinguishes our approach from the majority of bag-of-word-based approaches.

4.1.6 Evaluation

For the evaluation we use precision, recall, and F_1 score as the metric. Table 4.3 depicts the simple binary case. Tables 4.4 - 4.7 depict the results for the multi category classification.

Tables 4.3-4.7 give a more detailed view on the results obtained when the classi-

[8]We used TreeTagger [Schmid, 1995] for tagging

fier has to decide between all five classes. Generally, the results for *definite* and *none* classes are very similar to the binary decision. When we remove the suffix feature, then the performance for *definite* drops from the high 80s to 71.4. Surprisingly, the other combination of features has no real impact on these classes. In contrast, if we consider the most challenging class, the *object* class, then we see the high impact of the part-of-speech feature (decrease from F_1=14.9% to F_1=2.0%). The *general* and *functional* categories do also not perform very well, by only reaching F_1 of 69% and 53%, respectively.

category	FP	FN	TP	FREQ	Precision	Recall	F_1 score
street	22	50	635	685	89.8	86.4	88.1
non-street	37	24	810	834	93.0	94.4	93.7

Table 4.3: classification results, with all features

category	FP	FN	TP	FREQ	Precision	Recall	F_1 score
definite	53	21	461	482	86.2	91.7	88.8
general	14	22	80	102	69.0	64.5	66.7
functional	0	20	23	43	53.5	36.5	43.4
object	2	26	7	33	20.0	11.9	14.9
none	43	23	815	838	92.5	94.7	93.6

Table 4.4: classification results, all categories, all features

category	FP	FN	TP	FREQ	Precision	Recall	F_1 score
definite	63	85	397	482	72.9	70.0	71.4
general	10	42	60	102	53.6	41.7	46.9
functional	2	28	15	43	33.3	21.1	25.9
object	2	24	9	33	25.7	15.8	19.6
none	138	36	802	838	82.1	91.7	86.7

Table 4.5: classification results, all categories, without suffixes

Looking at the results for our toponym recognition approach, it can be seen that the classifier reaches its limitations as soon as a more fine-grained distinction between classes is required. We identified two reasons why a fine-grained distinction is hard: i) The per-sentence context does not give sufficient information, and ii)

category	FP	FN	TP	FREQ	Precision	Recall	F_1 score
definite	53	27	455	482	85.0	89.4	87.2
general	23	18	84	102	67.2	70.0	68.6
functional	3	16	27	43	58.7	45.8	51.4
object	2	32	1	33	2.9	1.5	2.0
none	36	24	814	838	93.1	94.4	93.8

Table 4.6: classification results, all categories, without POS

category	FP	FN	TP	FREQ	Precision	Recall	F_1 score
definite	61	26	456	482	84.0	89.8	86.8
general	17	20	82	102	68.9	67.2	68.0
functional	5	15	28	43	58.3	48.3	52.8
object	2	32	1	33	2.9	1.5	2.0
none	33	25	813	838	93.3	94.2	93.8

Table 4.7: classification results, all categories, only suffixes

A complete toponym resolution is needed to disambiguate between equally named toponyms that belong to different classes. In the next section, a more sophisticated approach is presented, which considers more context and integrates already classified entities in the decision process. One important factor is the relation between different toponyms and therefore a simple type classification is not sufficient. Each toponym in the text must be referenced to its real world reference.

4.2 Toponym Resolution

Toponym resolution can be seen as toponym recognition that has been extended by linking the toponyms to their real world entities. Leidner et al. [Leidner et al., 2003] call this step either *geo-tagging* or *grounding*. We prefer the term *grounding* since *geo-tagging* is misleading as it is often used in the context of linking photos to a geospatial location name which represents another task.

The most challenging part of toponym resolution is solving ambiguities. Overell [Overell, 2011] presented a summary of empirical studies that focused on ambiguous names in gazetteers: [Brunner and Purves, 2008] observed that depending on

the gazetteer 10-50% of all place names are ambiguous. [Smith and Crane, 2001] distinguish between continents in their analysis and come to the conclusion that while the rate of ambiguity in Africa is 17-20%, it rises to 25-57% for new world colonies. The first solution to disambiguate the place names of a given article is presented in [Smith and Crane, 2001]. Their approach makes the assumption that all places in a document are located close to each other. Therefore all combinations of the toponym resolution have to be considered and for each of them the centroid has to be calculated. The combination that has the least distance from all places to the centroid is then used for the disambiguation. In contrast to this computationally intensive approach, the approach of [Overell, 2009] uses Wikipedia to disambiguate ambiguous place names. This approach focuses on the most common location for the disambiguation and reaches 89,6%. However, the approach has two limitations: (i) if the text mentions smaller places that have a larger namesake (ii) if the place is not mentioned in Wikipedia. In both cases their approach will not find a correct solution. If we reconsider the requirements of geospatial applications like context-aware systems then we are also interested in good performance on the finer granularities, which means that we also need a good performance on the long-tail, e.g. we should be able to ground the toponym *Berlin, MD* to the small town *Berlin* in *Maryland, USA* instead of mapping it to *Berlin, Germany*.

Our approach also uses the idea that toponyms in documents must have a relation to each other. Instead of using the geometric distance between the toponyms we are using a hierarchical approach, which uses administrative entities (country, state, city, etc.). This leads to new requirements for the classification task since in the previous section 4.1 we only used the local context of the mention for the classification. As a consequence, we had to integrate features that consider the global context into our toponym resolution approach.

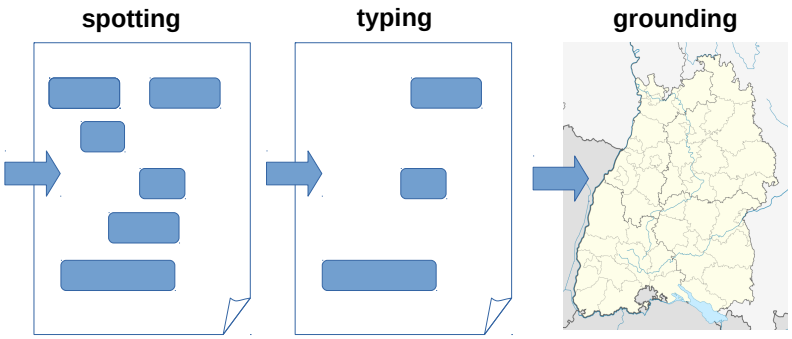

Figure 4.2: Three step mapping process: spotting, typing, and grounding.

4.2.1 Work Flow

Our toponym resolution process consists of three steps that are depicted in figure 4.2. The first two steps are analogous to the methods described above (4.1). In step one, we detect all mentions that are possible toponym candidates. We call this step *spotting*. The *spotting* task provides two challenges. Firstly, a toponym can be a multi word expression, and a simple lookup of each token in a knowledge base is not sufficient. Secondly, the tokens can be inflected which also makes the lookup difficult. The second step of our processing workflow includes typing. It is similar to the toponym recognition approach which was described in the previous section. The third step requires a novel method which relates each toponym to its real word object.

4.2.2 Knowledge Base

We use two main sources for our knowledge base: a standard gazetteer[9] is used to get global geospatial knowledge about the world. This is not fine-grained enough for our purposes. Therefore, we use the NAVTEQ data set (3.1.5) as a detailed geographic knowledge base for Germany. This data set represents all streets, cities,

[9]http://www.geonames.org

regions and states inside Germany. To use this resource as a knowledge base we had to perform a few preprocessing steps.

- Normalization: Replacement of all non-alphabetic characters with spaces;
- Capitalization: Adjustment of lower and upper case;
- Expansion: Production of abbreviations and writing variants (expand *Hauptstraße* to: *Hauptstrasse*, *Hauptstr.*);
- Extension: Inclusion of misspelled variants (*Württenberg* instead of *Württemberg*).

4.2.3 Model

We use the following notation for the specification of our model:

$\mathbb{E}$	set of linguistic entities in the knowledge base
$e \in \mathbb{E}$	linguistic entity
$E =< e_1, \ldots, e_n >$	sequence of linguistic entities (from spotter)
$\mathbb{T}$	set of typed entities (incl. non-geo)
$t \in \mathbb{T}$	typed entity
$T =< t_1, \ldots, t_n >$	sequence of typed entities
$\mathbb{G}$	set of toponyms (geospatial objects) in the database
$g \in \mathbb{G}$	toponym (geospatial object)
$G =< g_1, \ldots, g_n >$	sequence of toponyms
K_{ci}	context knowledge available at pos. “i”
K_g	set of global (world) knowledge items
L_c	linguistic context

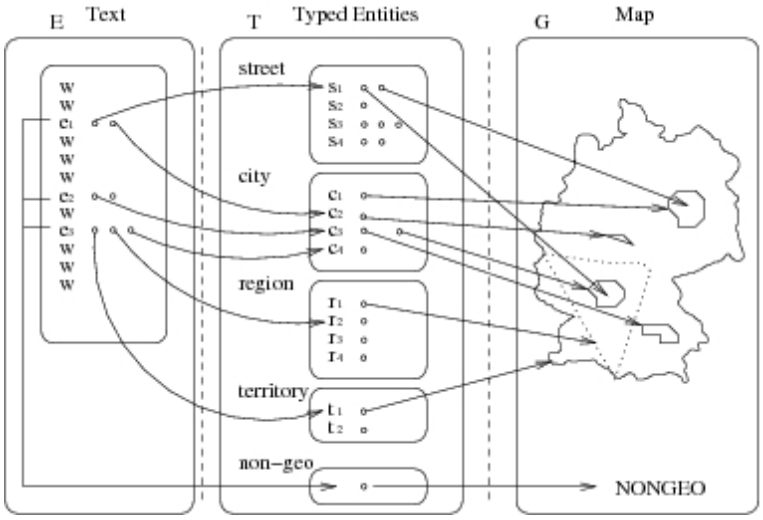

Figure 4.3: 2-Step Mapping Process: From linguistic entities to typed entities to toponyms.

We distinguish the following types of toponyms:

$\mathbb{G}_s$: street {A1, ..., Zypressenweg}

$\mathbb{G}_c$: city(town){Aachen, ...,Zwickau}

$\mathbb{G}_r$: region{Baden, ..., Württemberg}

$\mathbb{G}_t$: territory(country) {Afghanistan, ..., Zypern}

$\mathbb{G}_n$: non-geo{non-geospatial reading}

Mapping Process

The spotting process returns a sequence E of linguistic entities $e_1, ..., e_n$. As shown in Figure 4.3, we then want to type (Text $\rightarrow$ Typed entities) and reference (Typed Entities $\rightarrow$ Map) each of these entities. The leftmost box in Figure 4.3 contains the input text with marked linguistic entities e_i. Each linguistic entity can be associated to one or more of the available five toponym types (street, city, region, territory, non-geo).

Example: *Berlin*-city, *Berlin*-non-geo (the composer), *Berlin*-region (federal state in *Germany*).

A typed entity may in turn have several mapping possibilities to a toponym in the context model (*Berlin*-city $\rightarrow$ *Berlin OH*, *Berlin PA* etc.). Our goal is to find a mapping function which assigns the current linguistic entity e_i the best typed entity t_i and then assigns t_i the best toponym g_i, given the current context.

For the type non-geo, we introduce the dummy object NONGEO. All linguistic entities that were typed as non-geo are mapped to NONGEO.

Knowledge Sets

K_{ci} is the set of pairs <entity, toponym> that has been explicitly mentioned up to position $i-1$ in the text. This set is constantly increasing during grounding because each grounded entity is incrementally included.

In addition, when processing text which is associated with a particular locality, there is a set of toponyms associated with the locality that is immediately available for grounding without having been mentioned explicitly. For example, if a street name in an article in *Stuttgarter Zeitung* is noted without any city references, we can infer that the current city in this context must be *Stuttgart*. We initialize K_{c1} with the set of prominent toponyms associated with the locality of publication.

There is a second set K_g that contains a set of universally prominent toponyms such as *5th Avenue*, *Champs Elysees* or *New York City*. These entities are always capable of being referenced directly – without prior geographic context. Non-prominent entities (e.g., a small town like *Backnang* in *Germany*) always require some form of geographic context ("*Backnang* in *Germany*", "we took the train to *Stuttgart* and then continued on to *Backnang*" etc); in contrast, prominent entities can be used without such geographic context.

We distinguish K_{ci} and K_g because entities in the former knowledge base are available for grounding (the default interpretation for *Bahnhofstraße* in the

Stuttgarter Zeitung is the corresponding street in *Stuttgart*, a member of K_{ci}), but not for the latter ("*Peter Cooper Rd*" cannot be used without geographic context even though *New York City* is in K_g; *New York City* has to be mentioned explicitly and then added to K_{ci}).

Calculus

Our goal is to find the best sequence G of toponyms given an input sequence E of linguistic entities, a linguistic context L_c, and contexts K_{c1} and K_g:

$$G_{best}(E) = arg\max_{G} P(G|EL_cK_{c1}K_g)$$

We have defined an intermediate step to simplify the two different mapping steps. In the intermediate step we assign each linguistic entity to a typed entity:

$$\begin{array}{cccc} g_1 & g_2 & \dots & g_n \\ | & | & & | \\ t_1 & t_2 & \dots & t_n \\ | & | & & | \\ e_1 & e_2 & \dots & e_n \end{array}$$

For example, "Berlin" might be typed as a city and "main street" as a street.

We assume that t_i only depends on e_i and the linguistic and geographic contexts:

$$P(t_i|E, L_c, K_{c1}, K_g) = P(t_i|e_i, L_c, K_{ci}, K_g)$$

Recall that K_{ci} contains all previously made grounding decisions in the article. That is, once e_i has been assigned to g_i, we add $< g_i, t_i >$ to $K_{c(i+1)}$.

We assume that g_i only depends on t_i and the geographic contexts:

$$P(g_i|t_i, E, L_c, K_{c1}, K_g) = P(g_i|t_i, K_{ci}, K_g)$$

Assuming conditional independence of the typing and referencing decisions in the n positions of the sequence, we obtain:

$$
\begin{aligned}
G_{best}(E) &= arg\max_{G} P(G|E, L_c, K_{c1}, K_g) \\
&= arg\max_{G} \prod_{i=1...n} P(g_i, t_i|E, L_c, K_{ci}, K_g) \\
&= arg\max_{G} \prod_{i=1...n} P(g_i|t_i, E, L_c, K_{c1}, K_g) P(t_i|E, L_c, K_{c1}, K_g) \\
&= arg\max_{G} \prod_{i=1...n} P(g_i|t_i, K_{ci}, K_g) P(t_i|e_i, L_c, K_{ci}, K_g)
\end{aligned}
$$

The components $P(g_i|t_i, K_{ci}, K_g)$ and $P(t_i|e_i, L_c, K_{ci}, K_g)$ are partially set by constraints we impose on the model and partially estimated from annotated training data.

Constraint 1. One sense per article. If a linguistic entity e_i has been tagged as g_i, then all subsequent tokens of the same entity also have to be tagged as g_i. For example, if a sentence mentioning Berlin, the capital of Germany, is followed by the expression "*Berlin* is cool.", then *Berlin* cannot refer to *Irving Berlin*. We formalize this constraint as follows:

$$P(g_i|t_i, K_g, K_{ci}) = 1.0 \text{ if } < g_i, t_i > \in K_{ci}$$

We can construct sentences which violate this constraint: *Paris Hilton* stays in *Paris*. The first "Paris" refers to a person and the second to the city. In this case we argue that the spotting had to use a longest match strategy and we get different entities: "[*Paris Hilton*] stays in [*Paris*]". In this example our constraint holds. A more constructed sentence like: "[*Paris*] stays in [*Paris*]." cannot be resolved by our approach. However, we think that such wordplays are very uncommon.

Constraint 2. Completeness of knowledge base. If our geographic knowledge

base does not contain a toponym, then we ignore that toponym:

$$P(g_i|t_i, K_g, K_{ci}) = 0.0 \text{ if } < g_i, t_i > \notin \mathbb{G}$$

A special case of this constraint is that a linguistic entity that was typed as non-geo can only be assigned to the dummy object NONGEO. If we reconsider the goal of the task, which is to assign to each toponym a specific location that has coordinates, then it will be clear that we can only do that if we have a representation of the object in our knowledge base.

Constraint 3. Mandatory geographic context. A toponym can only be introduced if its geographic context is accessible. There are three types of accessibility:

- (i) the object is in K_{ci}
- (ii) the object is in K_g
- (iii) the object is a child of an object in K_{ci}

To formalize this notion, we have defined a part-of relation. It obeys the following axioms:

$$\forall g_s \in \mathbb{G}_s \exists g_c \in \mathbb{G}_c R_{partof}(g_s, g_c)$$
$$\forall g_c \in \mathbb{G}_c \exists g_r \in \mathbb{G}_r R_{partof}(g_c, g_r)$$
$$\forall g_r \in \mathbb{G}_r \exists g_t \in \mathbb{G}_t R_{partof}(g_r, g_t)$$

That is, each street is part of at least one city, each city is part of at least one region and each region is part of at least one territory.

With this part-of relation, we can formalize Constraint 3 as follows:

$$
\begin{aligned}
P(g_i|t_i, K_g, K_{ci}) \;&=\; 0.0 \\
&\text{if} \;\; \neg \;\; (\quad < g_i, t_i > \in K_g \\
&\qquad\qquad \vee < g_i, t_i > \in K_{ci} \\
&\qquad\qquad \vee \exists g_j, t_j [R_{partof}(g_i, g_j) \wedge < g_j, t_j > \in K_{ci}] \;)
\end{aligned}
$$

In preliminary experiments, we have found that imposing these constraints very rarely causes tagging errors. However, it is important to build a good linguistic preprocessing module. For example, the spotter must recognize *Jack London* and *Berlin OH* as linguistic entities otherwise phrases like "from *London* to *Jack London*'s home town" or "from *Berlin*, *Germany*, to *Berlin OH*" cannot be tagged correctly. Also, without proper linguistic preprocessing different forms of the same name cannot be recognized (e.g., the geographic database may list *Freiburg im Breisgau*, but this city is often just referred to as *Freiburg*).

The parameters $P(g_i|t_i, K_g, K_{ci})$ that are not covered by the constraints and the parameters $P(t_i|e_i, L_c, K_{ci}, K_g)$ are estimated from a labeled training set. In the training set, each instance of typing and each instance of tagging are represented as labeled feature vectors. We use external and internal evidence to build feature vectors.

External evidence includes the right and left linguistic contexts of the current linguistic entity (window size 3); and whether the linguistic entity already occurs with the same type / object or a different type / object in K_{ci} or K_g. We extract words and bigrams from the linguistic contexts and add them to the feature vectors.

As internal evidence we use the shape of the linguistic entity (e.g., does it contain a hyphen), its last morpheme, its capitalization and a shortened representation (conflating capitalized characters, conflating non-capitalized characters, conflating special characters).

We use Support Vector Machines (SVM) to build a probabilistic classifier from

the training set. SVM scores are then transformed into probabilities using logistic regression.

Enumerations

Entity enumerations (e.g. coordination) in texts have a special status in our model because all elements of an enumeration are mapped to the same toponym type. We have confirmed this hypothesis in a small empirical study (we automatically extracted all enumerations from our corpora and found no exceptions). Therefore, for grounding enumerations, all sequences of linguistic entities $e_i, ..., e_{i+n}$ which are separated by one of the tokens {, ; : - and or} (and no other tokens) are combined to a new linguistic entity e'. After e' has been assigned a typed entity t', all constituents of the enumeration are assigned t'. The grounding process is then run separately for each constituent.

Application

We integrated parts[10] of our toponym resolution methods as a plugin into the OpenJump[11] framework. OpenJump is a geographic information system (GIS) which is used to visualize and manipulate geospatial content. Our text interface allows to integrate textual content into this system and to visualize the corresponding entities in the mapping. The benefit of using a service like OpenJump instead of services like GoogleMaps is that the plugins have access to the complete data set. This enables name indexing and the application of knowledge base aided disambiguation methods. Other mash-up systems like GoogleMaps provide only black box toponym resolution systems which are mainly based on returning the most frequent toponym of an ambiguous geospatial entity.

[10]Due to the lack of annotated data we did not implement the complete system and we can not provide a sophisticated evaluation. Experiments on city toponyms showed that we reach a F_1 score of 90%.

[11]http://www.openjump.org/

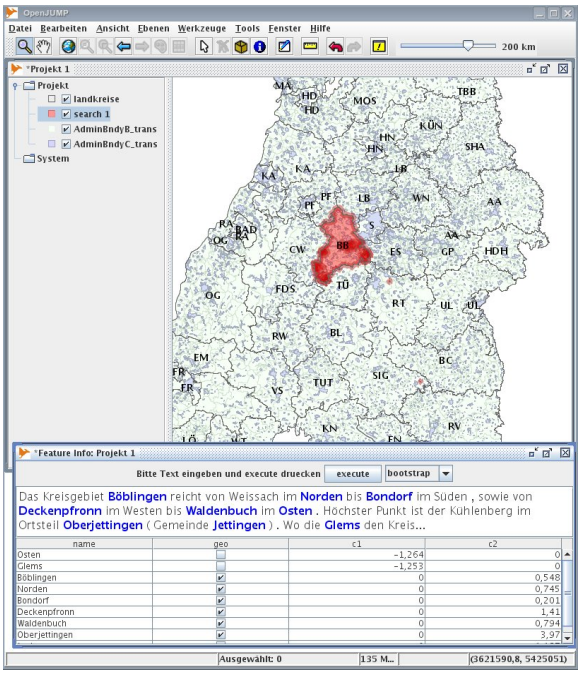

Figure 4.4: Prototype of our city name classifier. A user can insert a text and all toponyms are recognized and mapped to a map. The prototype is implemented as a plugin for OpenJump.

4.3 Entity Linkage for Toponyms

In this section we consider how to aggregate data from different resources that contain overlapping information. The linkage of different data sets is not trivial since they can use different attributes or slightly different values for the attributes. But if an application needs all available information about a particular toponym, a method is needed to find corresponding toponyms and merge the existing attributes. Different research communities are working on this topic. In the field of geographic science only the geometric parts of the data are used to calculate matching pairs. The more computer science orientated field of information retrieval uses the similarity of string values to match the data sets. Our approach can be seen as a hybrid approach that combines these methods and adds some linguistically motivated similarity features. In a collaboration with the visualization group of the

University of Stuttgart we integrated methods from visual analytics to simplify the feature engineering task.

4.3.1 Related Work

Previous work on matching of toponyms includes [Sehgal et al., 2006]. They match data sets for Afghanistan from two different providers. They use three different string similarity measures as well as physical distance. First they consider each feature independently and then compare different threshold values for the corresponding feature. In the last step they combine the features and learn a function to weight them. The main difference to our work is that they do not use any visual analysis. Furthermore our feature set is richer. In a follow-up publication, a graphical tool for entity resolution was presented [Kang et al., 2007]. The tool provides many configuration options for automatic matching, but is focused on user-driven interactive matching.

Our iterative matching approach can be seen as the continuation of two different approaches. First, the semantic matching approach. [Giunchiglia et al., 2008] show how to use semantic features for matching entities. Secondly, the collective classification approach [Besag, 1986, Lu and Getoor, 2003, Jensen et al., 2004] emphasizes the relevance of relational aspects in general classification tasks.

4.3.2 Data and Annotation

We simulate the data integration with two data sets from different providers. We use again data sets from Geonames (Section 3.1.4) and NAVTEQ (Section 3.1.5). Our scenario requires the linkage of cities of Germany. This is a challenging task since it is not clear how the quality and coverage of collaborative resource like Geonames is. Furthermore, the two providers use different data representations, whereas our experiments need a common feature representation. Consequently, a few preprocessing steps are obligatory. In the NAVTEQ data set each location is a

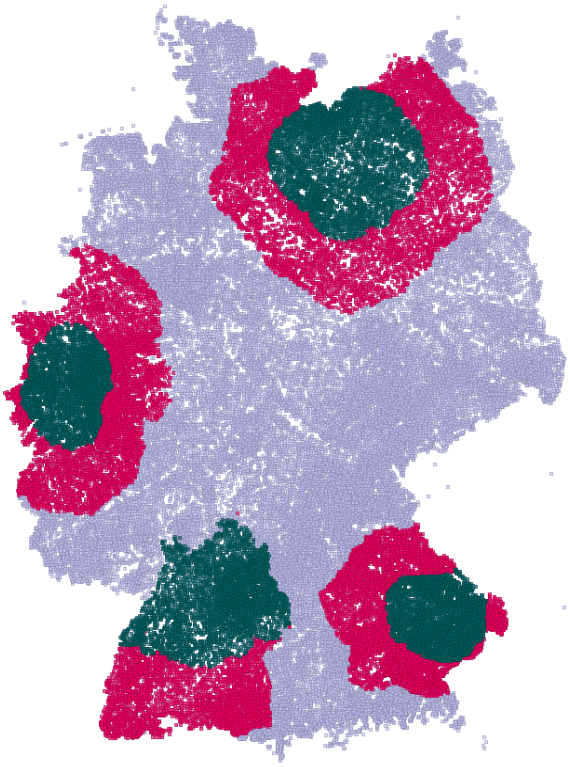

Figure 4.5: The area on the lower left side (corresponding to the German state Baden-Württemberg) is used as development set. It is further split into a training (north BW) and a test set (south BW). The other three areas (WEST, NORTH-EAST and SOUTH-EAST) are only used for the final evaluation. Each evaluation area contains about 10,000 objects from the Geonames data set.

complex object with a geometric attribute and several symbolic attributes. The geometric attribute contains exactly one polygon, which draws the geospatial region of the location. The symbolic attributes include of several string attributes. The NAVTEQ string attributes only contain letters in upper case (ISO-8859-1 char-set). This is a drawback for German because the German letter *ß* cannot be represented correctly which makes string matching more difficult. Geonames data items consist of a geo-position and a name attribute which are in the correct case and also contain the German letter *ß*. We only consider the geo-position and the name attribute in this work.

For training and evaluation we need manually annotated data. Our data set

is split into four regions that are far away from each others to compensate for differences in their regional properties. Figure 4.5 shows these regions (WEST, NORTH-EAST, SOUTH-EAST and SOUTH-WEST). Each region is split into two parts: the red one is used as training set and the green one is used as evaluation set. Furthermore, we used the SOUTH-WEST region, which is equal to the German state Baden-Württemberg, as development set. The snowball selection approach [Biernacki and Waldorf, 1981] is used to determine the instances of these data sets. One property of this approach is that all drawn instances are connected. This fact is important since in our later approach we exploit knowledge about the local surrounding of each instance. To simplify the annotation process, we have developed a simple tool to assist the human annotator in suggesting possible matches. The tool suggests possible matching pairs by ordering all pairs with a simple heuristic which computes the similarity of the names.

Our data sets are annotated by two annotators. The inter-annotator agreement is good [Eugenio and Glass, 2004] (see table 4.8, (κ >0.8). A matching candidate is defined as a match if both annotators annotated it as a match. The annotation

	SOUTH-WEST	SOUTH-EAST	WEST	NORTH-EAST
κ	0.92	0.89	0.77	0.84

Table 4.8: Kappa values for annotated regions

process results in 5682 corresponding items.

4.3.3 Feature Selection

Our feature set was optimized on the development set of Baden-Württemberg in several iterations. In each iteration, we defined additional features that can cover candidate pairs that were not handled correctly in the previous iteration. Each matching candidate is represented by a set of features.

Table 4.9 lists the used features. The first feature models the spatial distance

name	explanation	examples
logDist	spatial distance	log(Esslingen,Stuttgart) = 1.04
sim	trigram similarity	Essen-Esslingen = 3/10
levenshtein	levenshtein distance	Essen-Esslingen = 4
partof	part-of relation	Essen-Esslingen = 0 ; Urach-Bad Urach = 1
Bad	starts with Bad	Bad Urach = 1 ; Stuttgart = 0
fw	forward match	Essen-Esslingen = 5
bw	backward match	Essen-Esslingen = 0 ; Suhlingen-Esslingen = 6
hyph	contains hypen	Stuttgart-Untertürkheim = 1
sim_05	other candidate has sim>0.5	

Table 4.9: Basic features developed for the integration task.

between source and destination object. It is denoted by *logDist*: $\log_{10}(distance)$. Besides the logDist distance feature, a set of features is designed to measure the similarity between names. Extensive experiments showed that it is not possible to represent all necessary information about name similarity in a single feature. We separated the different properties of string similarity into 7 features (cf. middle part of table 4.9). The usage of several morpheme-based features as a measurement of string similarity is also enforced by the evaluation results of our street name identifier (cf. Section 4.1.6). Finally, we add a feature that counts other possible candidates in the vicinity that have a sim value higher than 0.5.

4.3.4 Classification

Each instance of an object from Geonames and NAVTEQ is transformed into a feature vector including a set of the above described features. Classification (see algorithm 1) is done using a J48 decision tree with pruning. The reason for this choice is that the classification decisions can be analyzed and understood more easily than those of many other classifiers.

The first results showed that the straightforward classifier is not sufficient, because nearby classifications have an impact on each other. Thus an iterative algorithm is applied instead. In each iteration the classification result of the previous

Algorithm 1 Classifier Algorithm

1: compute $\vec{a}_i =< f_1(v_i), \ldots, f_n(v_i) >$ for each training set node
2: train J48 decision tree g on training set $\{(\vec{a}_i, class)\}$
3: compute $\vec{a}_i =< f_1(v_i), \ldots, f_n(v_i) >$ for each test set node
4: apply g to test set $\{\vec{a}_i\}$

iteration is taken as input. In the *bootstrap step* an initial classifier is trained on the training set. This classifier is then applied to the training set. The modified set is the input for the second classifier which can be applied iteratively. Two additional features model the previous assignments. The feature *preScore* models the score of the previous iteration. The more important new feature *rank* is the rank value. The ranking is built over candidate pairs that include the same Geonames object.

4.3.5 Evaluation

We used three different feature sets. The simplest combination of distance and similarity (SIMPLE in table 4.10) is used as baseline because it contains the most basic information for making a decision. The second set includes all features developed in our work. Since the features are not independent, this does not result in the best possible outcome. In the third set the BEST feature set for the development set was determined by trying all permutations ($2^{15} - 1 = 32767$) of all features. The combination of *logDist*, *levenshtein*, *substLev*, *partof*, and *hyph* reached the highest performance. In table 4.10 the f_Scores of the different classifiers are listed. The main result is that the iterative approach performs very robustly on the different regions.

4.4 Acquisition of Vernacular Names

The previous section presented methods which are necessary for processing toponyms of textual resources. In this section we consider an application that com-

		SIMPLE	ALL	BEST	ITERATIVE
development		52.3%	91.0%	93.0%	91.8%
SOUTH-EAST	D	51.1%	86.0%	86.4%	86.4%
SOUTH-EAST	S	72.1%	88.3%	86.7%	90.9%
WEST	D	34.6%	77.8%	81.3%	82.4%
WEST	S	62.0%	80.0%	85.4%	88.4%
NORTH-EAST	D	55.0%	69.6%	66.7%	82.6%
NORTH-EAST	S	48.0%	82.9%	82.0%	89.1%

Table 4.10: F_1-scores. D: classifier trained on development set. S: classifier trained on similar region.

bines these methods. The goal is to acquire new knowledge from text that is not represented in geospatial resources. In preliminary experiments, we found that regions that are named by vernacular names are rarely represented in geospatial data. This approach aims to learn these region names from textual resources and relates them to a geometric representation that can be visualized in a map.

4.4.1 Motivation

Again, we use a context-aware system (Section 1.1.1) as our scenario. Such systems need comprehensive data in their context models to represent the real world. Geospatial data which are provided by surveyors build the basic datasets in context models. However, the currently available geospatial data are not sufficient for all context-aware applications. New ways are needed to augment the existing models with new geospatial data. Our work focuses on places and regions which are not considered in the currently available data collections, because most of them have no administrative relevance. Such regions are commonly used in colloquial language and play a significant role in many ordinary tasks like route planning, or arranging meeting points. Context aware systems should provide an easy interaction to satisfy user needs. Therefore it is important that systems understand the same geospatial terms as the user. Particularly, if the user is in a foreign city and looks for certain places in a region which is named by a vernacular name and if

the user has no opportunity to translate the name to a term understandable by the system. Context-aware applications need very fine-grained geospatial information on different levels of detail to get highly acceptable usability.

The acquisition of so-called vernacular regions is not comparable to that of other geospatial data. For example digital street maps are acquired by cars with a GPS receiver and photo cameras to get a photo of the street name plate corresponding to the actual position. For vernacular places this is not possible because there are no visible indicators in the real world to identify the locations corresponding to the region. Furthermore there is no information in official maps to solve the problem. Other regions like cities have landmarks around their boundaries and can also be determined by surveyors. There are reasons why vernacular regions are also called imprecise regions [Pasley et al., 2007]. In [Joho and Sanderson, 2004] and [Clough, 2005] a set of 1 TB multilingual web pages are analyzed. They used several gazetteers to find geospatial toponyms inside the pages and showed that there is a high amount of web pages with geospatial references. This suggests that acquiring vernacular regions by extracting geospatial information from textual web resources is feasible. We intentionally focus on textual pages instead of pages with mash-up geospatial data because we want to develop a generic approach which is based on natural language and is more powerful and richer in information.

Imprecise regions can be split into three types. *Functionally named* regions which get their name after their purpose (e.g. financial district, business quarter and red-light district), regions which have *vernacular names* (e.g. *Bohnenviertel*, *Lehenviertel*) and, at last, regions which are named after *spatial or historical attributes* (*Altstadt*, *Stuttgart-Süd*, *Zentrum*). Our regions are restricted to be a part of one city or maybe overlapping a few cities. In other work [Arampatzis et al., 2006] larger regions are considered, like the Midlands in the UK or the Midwest in the USA but these are not immediately important for context-aware applications.

4.4.2 Region-Tagging Approach

In this section we illustrate a method for extracting vernacular places with their spatial extension from web pages. This method is a proof of concept and we follow the software engineering principle: keep it as simple as possible.

Experimental Setup

For our experiment we need two resources: first, the Web as a large text resource and second, a Gazetteer. We use the Google API[12] as an interface to the web. In difference to other work in that field we use only street objects as real world references. We focus only on the German state of Baden-Württemberg. The data were provided by NAVTEQ (section 3.1.5). We used 70,000 different street objects. As we showed earlier (section 4.1), the data were highly ambiguous because over 90% of the streets share names with other streets. For example, a search for *Goethestraße* in the Stuttgart Metropolitan Region with retrieve 9 matches on *Google Maps*. That means that there are 9 different street names with the same name in a distance of 30 kilometers.

Algorithm

The following steps are used:

1. query for relevant web pages,
2. toponym recognition,
3. toponym resolution to coordinates,
4. clustering of nearby coordinates,
5. calculation of the convex hull of the clusters.

[12] At the time of our experiments the usage of the Google API was free.

In step one, a search query for the prompted places is generated. The query includes the name of the place as a positive term and some stopwords as negative terms. The stopwords are needed to exclude web pages which contain dozens of different addresses and make the query result noisy. In our case we use the following stopwords: *WG* (abbreviation for flat share), *Immobilie* (property), *Wohnung* (flat), these are used on typical advertisement pages. If, for example, we want to get the spatial region of the campus of the University of Stuttgart, then our query looks like this: **+Campus +Stuttgart -WG -Immobilie -Wohnung** (syntax adequate for Google). After sending the query to the search engine the first ten results are downloaded. The next step is to apply a toponym recognition system. This means that each entity in the web page which refers to a geospatial object is marked. We simplified this task by tagging only street entities. Nevertheless many street names are highly ambiguous in German, but we dismiss this issue in this proof of concept task and use a straightforward approach that includes string matching.

Figure 4.6 shows a resulting page for the *Campus Stuttgart Stadtmitte* query. Each recognized street name entity is colored in blue. The first paragraph describes the core places of the campus. All marked names are very good indicators for the area of the campus, but the last two paragraphs concern leisure time activities in the vicinity of the campus and give bad indicators for the campus area. As a consequence, the names printed in blue italics are misleading and do not refer to relevant street objects in this context.

For this task we simplify the the toponym resolution task which is the third step of our method. Each street name which was recognized during step 2 is used to retrieve equally named street objects from the NAVTEQ data. If a name relates to several streets then all of these streets are retrieved.

The next step is used to filter out street objects which might be selected wrongly (e.g. because of the above mentioned ambiguity issue). The filtering makes use of the assumption that street objects which span the wanted region must be in a neighborhood relation. Our clustering algorithm builds a cluster of street objects

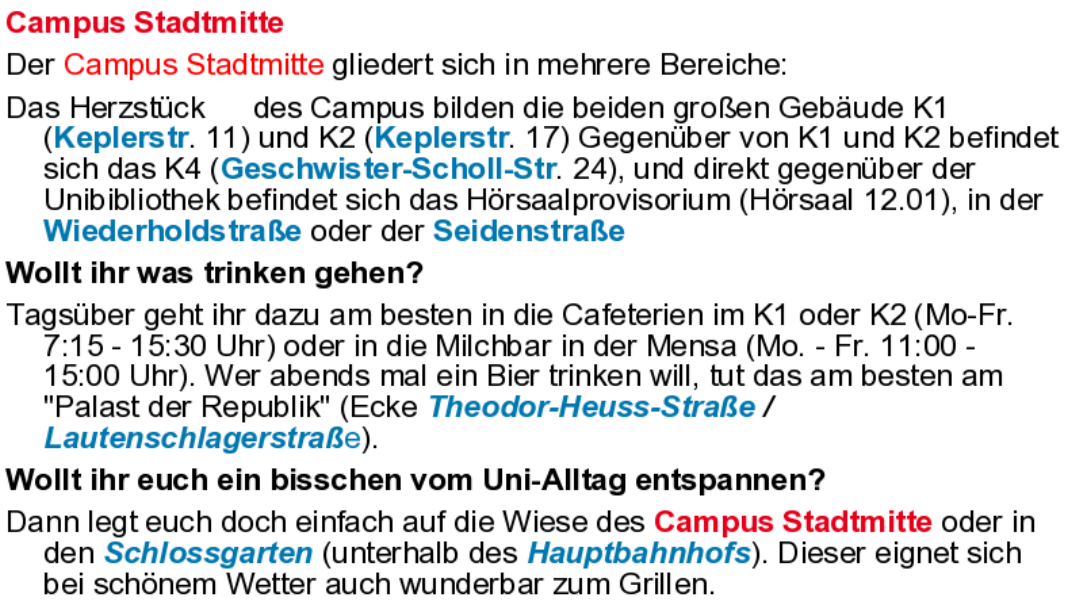

Figure 4.6: Sample web page about 'Campus Stadtmitte'. All recognized entities are colored in blue.

for each possible region.

In the last step the clusters are transformed to polygons by building a convex hull around all included street objects (Figure 4.7 depicts an example for the query *Campus Stuttgart Vaihingen*). The cluster contains street objects with the names *Pfaffenwaldring* and *Allmandring*. The yellow polygon is the convex hull around the streets and corresponds to the spatial extension of the queried *Stuttgart Campus Vaihingen* region.

4.4.3 Evaluation

In this section we provide a shallow evaluation, because the only goal of the approach described above was to show that the concept of web-based extraction of regions is possible. Figure 4.8 contains several queries of named areas in the conurbation of Stuttgart/Germany.

The purple polygon in the center stands for the *Lehenviertel*, which is the best

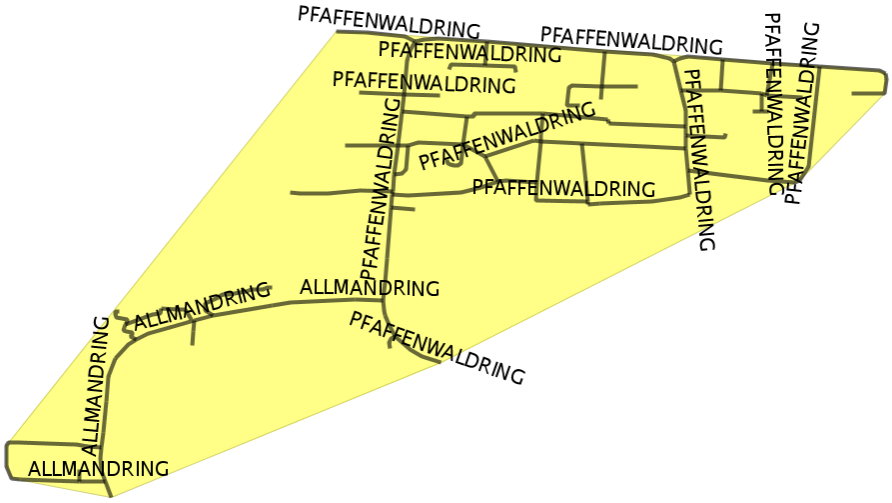

Figure 4.7: Clustering and convex hull building around street objects

recognized region by our approach. However, there are more purple polygons on the map. Outside the borders of Stuttgart exist more polygons: certainly they are all wrong and have nothing to do with the *Lehenviertel*. The grounding step did not disambiguate the referring street objects and so equally named streets in the surroundings of Stuttgart were also marked.

The yellow and green polygons mark named parts of Stuttgart (West (west), Nord (north)). The directions of the polygons are correct but the coverage is poor. As we use only the first ten hits of the search query, the information obtained is not sufficient. An increase of the used hits, in our case only the first ten are used, could be an appropriate countermeasure.

At last we will discuss the two functionally named areas. First the *Bankenviertel* (financial district). Surprisingly it delivers a very good result, although the name is not very common in Stuttgart. The *Campus* retrieves several polygons. The query is very ambitious because the University of Stuttgart is spread over the city and has more than one campus. The two biggest ones were recognized correctly. Again, the

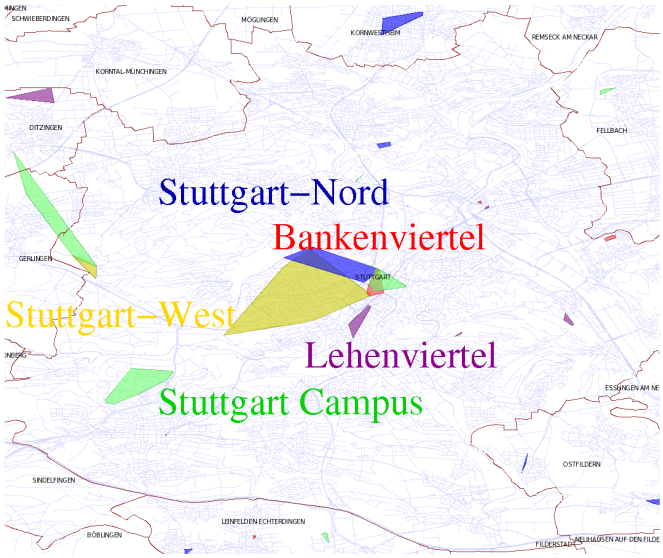

Figure 4.8: Vernacular region detection in Stuttgart.

polygon in the center is a little too big because many streets were associated with the campus (see figure 4.6).

The quality of the retrieved regions thus is not perfect but our very simple approach delivers a good approximation to the correct regions. We therefore claim that the proof of concept is successful.

4.5 Summary and Discussion

In this chapter we introduced methods for toponym recognition, toponym resolution, and toponym aggregation. One of the overall results is that machine learning based systems need large amounts of annotated resources to perform well. To develop and evaluate our systems we had to spend a huge effort on annotation. This is a reason why in many fields of NLP established data sets are used over many years. This limits new innovative research since much work focuses on optimizing methods for well-known resources (including evaluation sets).

Especially if we consider geospatial resources on a fine-grained level, annotated

resources are not available. A complete toponym resolution system would need data annotated on many different levels (streets, cities, regions, countries; but other geospatial entities might also be important: rivers, lakes, mountains, valleys, etc.). Our investigations on street data showed that already for only one level an immense effort is needed to provide guidelines and annotations. In the next chapter we provide two possible solutions. First, we consider relations between entities. Such recognized relations can be used to assist the toponym resolution (e.g. if we can decide that *Stuttgart* is close to the river *Neckar*). Second, we show that distant supervision can be used to minimize the annotation costs.

One common aspect of our approaches of this chapter is that we involve knowledge of existing machine readable resources in our approaches. A few years ago this could have been an issue since not all data were publicly available, but now we see no disadvantage in using such resources, since Geonames and OpenStreetMap are still growing.

One of the main future issues of Geospatial Information Extraction is a common evaluation. This is not trivial, because many applications focus only on local specialties. Thus various evaluation datasets are needed. The problem is comparable to recent issues in natural language processing tasks where each language with all underlying aspects needs individual evaluation corpora. Preparing such corpora is often very expensive and takes years. For regions named by vernacular names the establishment of a gold standard dataset is much more challenging, because the human raters who have to describe the borders of such regions must have a good local geographic knowledge. In [Blessing et al., 2007] we created a dataset of street names which occurred in geospatial and non-geospatial data. Although this is a very simple task, in some cases the annotators had to consult additional resources (e.g. web search) to come to a decision. To conclude we see no immediate solution for commonly usable gold standards in geospatial information retrieval. Instead we propose the use of already known regions as interim solution for our task.

Chapter 5

Distant Supervision for Extracting Relations

In this chapter, we introduce a method to enable supervised relation extraction without any need for manually annotated training data. The proposed method is based on the distant supervision (DS) paradigm [Mintz et al., 2009] which enables the generation of annotated training data without any manual annotation.

This chapter is structured as follows: first, we present details about the motivation and history of DS approaches (Section 5.2). Thereafter, we give an overview of our contributions in the field of DS (Section 5.3). These include DS for German, DS for fine-grained relations, and the observation that linguistic features have a positive influence on DS for German. The approaches which we developed and which underline our contributions are then introduced in Section 5.4. In a subsequent experimental section we describe the application of the developed approaches to different scenarios in the geospatial domain (Section 5.5). Finally, we present and discuss the result of our experiments and show the impact of our contributions in the field of DS. Note that this chapter is largely based on already published work: [Blessing and Schütze, 2010a] and [Blessing and Schütze, 2010b].

5.1 Relation Extraction

In Section 2.2.5, we introduced the idea of relation extraction and discussed the open challenges for the geospatial domain. In [Blessing et al., 2006], the idea of a "text sensor" is presented as using relation extraction methods for context-aware systems. That article shows how relation extraction can assist context-aware systems by gathering information from text. This is important to improve the data quality and coverage of such systems. The "text sensor" approach is complementary to the commonly used data acquisition of context-aware systems which is done by gathering data either from physical sensors or from manually prepared data. However, the approach described in [Blessing et al., 2006] has one drawback: it is a rule-based approach which can only be used for a given very specific task on a very rigid textual structure[1]. In contrast, in this chapter, we focus on machine learning-based methods to enable relation extraction for a broader domain.

Overview of Relation Extraction Systems

This section briefly introduces important and well-known systems for relation extraction (RE). These systems have in common that (i) they are not specifically developed for the geospatial domain and (ii) they can only extract information from English textual resources. In contrast, the approach presented later in this chapter (in Section 5.4) focuses on the extraction of relations in the geospatial domain. Note that the design and implementation of a RE system working for the geospatial domain requires to solve the problems addressed in the previous chapter 4. Furthermore, because our goal is to extract information about German locations, the presented system will have to handle German texts[2].

Although the systems described below do not focus on the geospatial domain for

[1]The system of [Blessing et al., 2006] was developed for messages about traffic jams which have a limited vocabulary and only little syntactic variance.

[2]Note that working with English texts would result in poor coverage for the extraction of German locations.

German, the presented overview gives an idea about which scalable systems for IE are available and how the field has developed in recent years.

Snowball [Agichtein and Gravano, 2000] was one of the first relation extraction systems requiring a minimal amount of training data. This system uses bootstrapping to generate extraction patterns derived from a set of given examples. In each iteration the system generates new examples in turn used to generate more patterns.

The **KnowItAll** Web IE system [Etzioni et al., 2004] took the next step in automating IE by learning to label its own training examples using only a small set of domain-independent extraction patterns. It was the first published system to carry out extraction from Web pages that was unsupervised, domain-independent, and large-scale. Nevertheless, it requires a laborious bootstrapping process for each relation of interest, and the set of relations has to be introduced by giving a number of seed examples. This is a significant obstacle to a more general extraction system because unanticipated concepts and relations are often encountered while processing text.

The **DARE** (Domain Adaptive Relation Extraction) [Xu et al., 2007] system implements a minimally supervised machine learning framework for extracting relations of various degrees of complexity. It bootstraps from a small set of instances of an n-ary relation (seeds) and automatically learns pattern rules from parsed data. These patterns are then used to extract new instances of the considered n-ary relation and its projections. The disadvantage is that such rule-based approaches do not have the same degree of freedom as a machine learning-based system.

SOFIE (A Self-Organizing Framework for Information Extraction) [Suchanek et al., 2009] performs automated ontology extension. SOFIE can parse natural language documents, extract ontological facts from them and link the facts into an ontology. The system uses logical reasoning to link existing and new knowledge in order to (i) disambiguate words to their most probable meaning, (ii) reason on the meaning of text patterns. One of the main drawbacks of SOFIE is

that this framework is not fast enough to be applied on web-scale data.

NELL (Never-Ending Language Learner, also known as ReadTheWeb) [Carlson et al., 2010] started in 2010 within an initial ontology which contains hundreds of categories (e.g. person, sportsTeam, fruit) and relations (e.g. PLAYSONTEAM(athlete,sportsTeam)). For each category a seed of 10 to 15 manually selected examples were used. At present[3], NELL has accumulated a knowledge base (KB) of 2,021,533 asserted instances of 859 different categories and relations. NELL allows interactive correction of wrongly interpreted instances by a web-based user interface. Schema drift is the main disadvantage of NELL since its knowledge base grows too fast, such that not all instances can be manually corrected.

5.2 Motivation for Distant Supervision

While our overview of available IE systems given (cf. Section 5.1) is rather application-centric, a more comprehensive survey based on IE tasks has been presented by [Sarawagi, 2008]. This survey shows that the systems obtaining the best results on the analyzed tasks are supervised. However, supervision is only possible if enough annotated data is available. Furthermore the available data has to fit the task definition. For instance, if the granularity of the relation or the domain of the text differ from the task, then supervision becomes difficult to perform. For illustration, suppose that we have a corpus annotated using the standard NER schema (*PER*, *GPE*, *ORG*, *MISC*) but we are only interested in named entities that are referring to countries. There is no straightforward way to extract only countries from the *GPE* annotated entities. However, there is no way to use supervision for a task defined on countries only using standard NER. In summary, it can be said that supervised approaches work best when they have enough annotated data for the requested task and/or domain, which is in real-life applications often not the case. Especially, when dealing with specific domains such as the geospatial domain.

[3]April 2013, http://rtw.ml.cmu.edu/rtw/overview

Another piece of evidence supporting the need to increase the coverage of supervised approaches is given by [Ji et al., 2010]. This study, based on human annotations, shows that in the context of the 2010 Knowledge Base Population shared task (KBP2010), Wiki-derived databases cover only well-known entities. For example, Freebase [Bollacker et al., 2008] covers only 48% of the slot types and 5% of the slot answers in the KBP2010 evaluation data. Again, this indicates a lack of annotated data for specific domains. As a consequence, supervised methods are not well suited for all domains.

Several methods have been proposed to counter the issue of missing annotated data. The most common one is to avoid the usage of supervised techniques and focus on unsupervised approaches. But unsupervised learning approaches are no general solution to this problem since they often have low accuracy or cannot easily be mapped to the required task.

We promote another solution. Instead of using unsupervised techniques, we want to cover missing annotated data by providing an alternative way to create annotated data. This solution is called "distant supervision"[4] (DS) [Snow et al., 2004, Mintz et al., 2009]. The basic idea is to substitute the manual annotation process by an automatic machine-driven annotation process. As shown in the introductory example (Section 2.1.3, Figure 2.1) DS uses existing knowledge bases to label text and then train a supervised model on the labels. A KB commonly used for this purpose is Freebase, but any reliable source of structured data (e.g., Wikipedia infoboxes) can be used. Closely related to DS are "DS-training" [Liao and Veeramachaneni, 2009, Kaljahi, 2010] and "DS-annotation" [Cimiano et al., 2004].

[4]There is some controversy about the naming of this paradigm. It also known as "weak supervision" and the term "self training" is also used in the literature.

A brief History of DS

Craven et al. [Craven et al., 1999] were the first to combine structured data with unstructured data to enable supervised learning in the field of natural language processing. They call their resulting annotation *weakly labeled data*. At this time, only the bio-chemistry domain provided enough structured resources for such an approach. It took several years before structured resources existed for other domains such as relations between persons. [Snow et al., 2004] developed a method for extracting hypernym relations between entities. They used WordNet[5] as their structured knowledge base (see Section 3.1). In 2009, [Mintz et al., 2009] introduced DS for the combination of the Freebase database and news-wire texts. We call this approach *redundancy-based DS*, since in that case all matches of a KB fact in the text are annotated. This results in noisy labels and requires the use of robust learning algorithms that can learn accurate classifiers from noisy data if the training set is large and redundant enough. Redundancy-based DS works well for simple relations as long as an appropriate knowledge base in the target language exists; but it is unclear how it can be applied to relations which occur less frequently in large corpora.

Riedel et al. [Riedel et al., 2010] used a similar method to annotate Wikipedia articles. We call their approach *structure-based DS*. In contrast to Mintz, they argue that the annotation of arbitrary text leads to many noisy annotations. They exploit structured data in combination with textual data (Wikipedia articles) to reach a higher quality for the annotation. Redundancy and structure-based DS use different assumptions which are shown in Figure 5.1. The *structure-based DS* assumption is weaker than the *redundancy-based DS* assumption. To counterbalance this weaker assumption they citeRiedel2010 propose multi-instance learning as a solution. Note that similar approaches are used in the field of named entity recognition [Nothman et al., 2013] which are called *silver-standard* training annotation.

Our work will follow the more realistic *structure-based DS* assumption.

[5]http://wordnet.princeton.edu/

- [Mintz et al., 2009]: If a pair of two entities that are an instance of a certain relation occur in the same sentence, then the sentence is a positive example to describe the relation between both entities.
- [Riedel et al., 2010]: If a pair of two entities that are an instance of a certain relation occur in many sentences of a document set, then at least one of these sentences is a positive example to describe the relation between both entities.

Figure 5.1: Two different assumptions to define DS.

5.3 Contributions

Our contribution in the field of DS is to apply and adapt DS to new domains, granularities of relations, and to new languages. In this section, we will give an overview of these contributions before we present the details in Section 5.4.

5.3.1 Geospatial Domain and Relation Granularity

For work in the geospatial domain, the need for generating annotated data in a cheap way is critical – especially for the preparation of context-aware systems (e.g. [Dürr et al., 2004]). Such systems need a rich knowledge base to make the right decisions in different user contexts. Therefore, geospatial data on different levels of detail is needed. But data providers do not exhaustively cover all these levels. Furthermore, the quality of the data in context-aware systems plays an important role for users. One aspects to guarantee high quality are frequent updates. Such updates can be provided by an information extraction component which increase the overall user acceptance.

For both issues, an information extraction system is required that can handle fine-grained relations, e.g., “X is a suburb of Y” or “the river X is a tributary of Y” – as opposed to simple containment. The Internet is a good source of knowledge for the extraction task, since it provides a wide spectrum geospatial data which is represented in textual content. The extraction component can be seen as a kind

of sensor that we call "text sensor" [Blessing et al., 2006]. However, no annotated resources are available to train supervised methods for such a task. The results in [Blessing et al., 2006] showed that rule-based methods can be used on special domains which are presented by means of texts with a restricted text structure (e.g. traffic notes, weather reports). When changes occur, even small ones, the whole rule base has to be adapted. This is highly inefficient. One solution is to introduce DS for the geospatial domain. Several tasks on this new domain will be presented throughout this chapter. Moreover, DS systems can also benefit from context-aware systems, since the already modeled data contained in these systems can be used as a knowledge base for the DS system.

5.3.2 DS for German

English is the best investigated language for DS, which is not a big surprise because this situation is analogous to most other fields of computational linguistics. Our goal is to show that DS is not restricted to the English language, and how DS can be used to generate annotated resources for German. One requirement to use the standard DS approach in a new language is that the KB and the textual source which has to be annotated are in the same language. Later in this work (chapter 6) we also provide an novel method to apply DS to new languages without this requirement of having knowledge base and text in the same language.

Some challenges of DS for German are similar to the challenges of NER in German which have been discussed in chapter 2.2.3. In comparison to English, we revealed three reasons for the higher complexity:

- freer word order;
- more complex morphology;
- capitalization of all nouns.

The distance between two related entities which make up a relation is mostly

restricted to a short span of words in English. Furthermore, the ordering of both entities is also mostly fixed[6] and thus predictable. In German, the positions of entities are mostly free. They can be in the Vorfeld, Mittelfeld and in a few cases also in the Nachfeld [Bader and Häussler, 2010]. This makes it more difficult to identify the entities in a sentence which are in a particular relation because all possible binary combinations of entities are candidates to form a relation. This increased search space leads to a more challenging annotation process.

Alignment between entity mentions (KB instances and textual representations) is an important factor in DS. In this respect, German is also more challenging than English since it is a morphologically richer language. The morphological form of words can change depending on the syntactic context of a sentence, which complicates the alignment. For illustration, consider the following example: `Er wohnt am Neckar, welcher einer der beiden Mannheimer Flüsse ist. (He resides near by the Neckar, which is one of Mannheim's two rivers.)` The form of the entity *Mannheim* is changed to the adjective *Mannheimer* because it is used in a possessive context. If we substitute *Mannheim* with another entity like *Neuhausen* then we get *Neuhäuser* in our example. Since inflection and derivation are not restricted to the suffix, a more complex matching component is needed to find the correct references in texts.

In German all nouns begin with a capital letter, which in comparison to English drastically increases the number of candidates for the named entity recognition.

5.4 Approach

We begin by presenting a detailed overview of our DS processing chain in Figure 5.2. Phase 1 represents the knowledge base population. For each of the requested relations, an adequate knowledge base must be identified. The main criterion for the selection is the amount of instances that can be derived from the KBs. Section

[6] If we consider passive constructions as separated case.

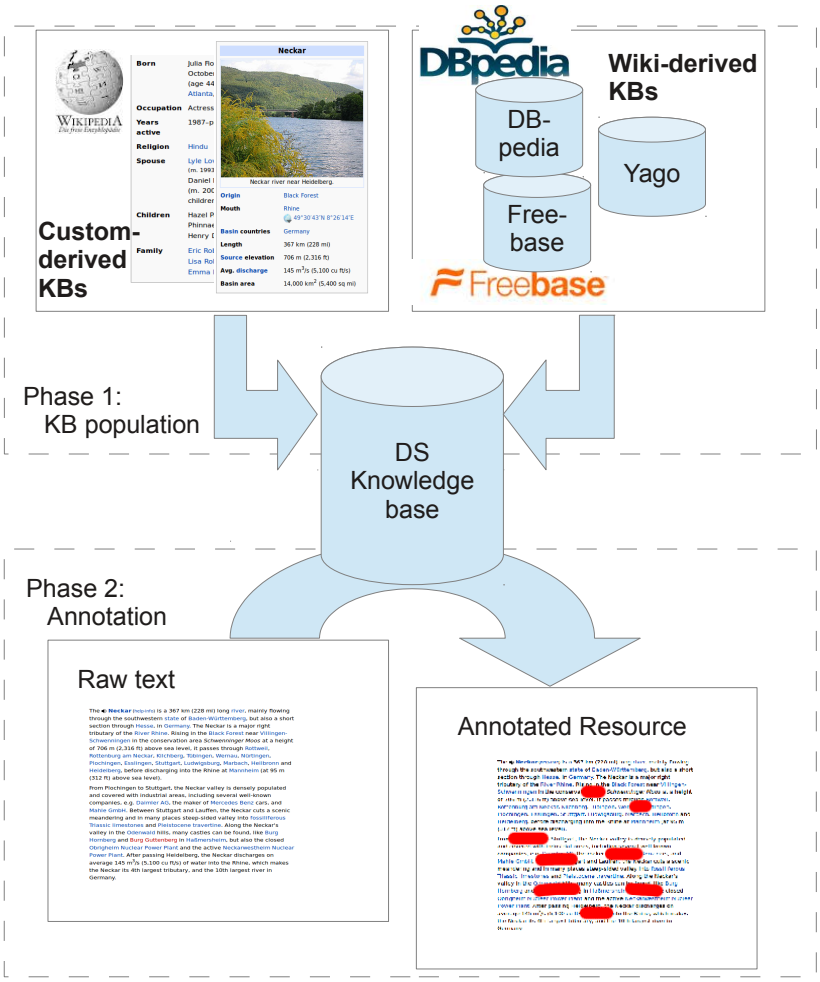

Figure 5.2: General architecture of distant supervision (DS). Phase 1: Knowledge base population. Phase 2: Automatic annotation.

5.4.1 describes the interaction with the KBs in more detail. In phase 2, the textual resource is augmented with labels which have been automatically generated from the DS KB. This annotation process is not in all cases straightforward since the KB instances have to be adapted to simplify the identification in the text. The methods we have developed for automatic labeling are presented in Section 5.4.2.

5.4.1 Knowledge Base Population

Wiki-derived KBs

In the simplest case all pairs of entities that belong to the given relation, which we call instances, can be retrieved from directly accessible KBs like Freebase. This method of acquisition should be preferred if there is enough data for the requested relation in the knowledge bases. If only Wiki-derived KBs are used then the mapping to Wikipedia articles is also simple because both use the same identifiers. This helps to avoid disambiguation. Several approaches [Riedel et al., 2010, Yao et al., 2010] have been proposed that use this kind of KBs for DS. But mostly only the most popular relation types are considered (especially for the evaluation procedure). Less frequent relations are often not included in such Wiki-derived KBs.

Infobox Extraction

A closer inspection of Wikipedia data reveals that some relations are not sufficiently modeled by Wiki-derived KBs. If we look at the German Wikipedia edition we find many infobox classes which are not mapped to any of the Wiki-derived KBs. This low coverage for specific types of relations is due to the fact that the goal of Wiki-derived KBs is to provide high-quality data. This goal is difficult to achieve together with high coverage on all types of relations. This has an important impact on relations appearing at more fine-grained levels. The main issue of these long-tail relations which occur rather rarely is that they are often represented in various ways since they are not popular enough for large numbers of users to unify the representations. Such data are often modeled by different attribute names, or the values are not defined and users have to use natural language to describe the data. In the experimental section 5.5, we give more examples of this fact.

Another issue is that attribute values often include free text. This is a barrier to automatic extraction because the words in the free text are often ambiguous names

Maas	[[Hollands Diep]] (ehemalige [[Nordsee]]bucht)
Memel (Fluss)	[[Kurisches Haff]] westlich [[Silute]]; Mündungsarm Atmata:
Eider	bei [[Tönning]] über den [[Purrenstrom]] in die [[Nordsee]]
Wertach (Fluss)	Nahe der [[Augsburg—Augsburger]] [[Wolfzahnau]] in den [[Lech]]
Schelde	[[Nordsee]] (Mündungsarm der Schelde: [[Westerschelde]]) ‘
Weschnitz	beim [[Kernkraftwerk Biblis]] in den [[Rhein]]
Panke	[[Spree]] 100 Meter flussab der Weidendammer Brücke
Wondreb (Fluss)	bei [[Mostov]] ”(Mostau)” in die [[Eger (Fluss)—Eger]]
Mokau_River	-

Table 5.1: Entries for the attribute “Mündung (mouth)” of the Infobox_Fluss (infobox river) template from the German Wikipedia edition.

or are modified by negation or other by linguistic means. As an example consider the river infobox in the German Wikipedia[7], which describes up to 50 different attributes of rivers. Of the 2072 infoboxes with values for the attribute “Großstädte” (large cities, which lists all large cities on the river) 1030 contain exactly one hyperlink, 778 contain enumerations of hyperlinks (separated by delimiters that vary from infobox to infobox), and the remaining 264 contain all kinds of different formats: names without hyperlinks, further descriptions, notes (stating in free text that no such city exists), etc.

Table 5.1 shows some entries for the mouth attribute of the infobox about rivers in the German Wikipedia. The mouth of a river is usually denoted by two entities which are toponyms. One describes the body of water into which the river flows (this can be another river, a lake or an ocean). The other toponym describes the location where this happens, which is usually a town. The examples in the table show that there is no normalized way to describe these two entities. The first line describes the mouth of the *Maas* which flows into *Hollands Diep*. The second toponym in this case denotes that *Hollands Diep* is the former *Nordseebucht (North Sea Bay)*. If we go more deeply into the used MediaWiki syntax, we see that entities, which have a corresponding article in Wikipedia, are encoded by *[[...]]* (e.g. *[[Hollands Diep]]*). But if we look at the *Nordseebucht* entity, we see that the compound

[7]For our experiments we used a dump of the German Wikipedia edition from October 2009 which was imported by means of JWPL (Section 3.2.4).

attribute	value
Großstädte 'large cities'	nächstgelegene Großstadt: [[Reutlingen]] 'nearest large city:' [[Reutlingen]]

Figure 5.3: Sample fact from the Infobox_Fluss (infobox river) template from the German Wikipedia for the article about the river *Erms_(Fluss)*. The value of the attribute "large cities" is misleading: the river-town relation does not hold for the entities river *Erms* and city *Reutlingen*.

is split. The first part which refers to the *North Sea* links to the corresponding entry in Wikipedia which is a holonym of *North Sea Bay*. For automatic processing methods, which only consider links, this usage is misleading. The additional term "ehemalige" (former) describes that the name *Nordseebucht* was used in the past. So the term *Nordseebucht* is only used to give the human reader a better understanding. If you consider the other rows you will see that there is a lot of mixed content that combines structured data and natural language. The last row gives an example where the value *'–'* is used to depict that there is no large city on the river, which is tricky since the *'–'* can also be interpreted as entity which redirects to the Wikipedia article about Viertelgeviertstrich[8].

In order to consider mixed contents for extracting fine-grained relations, we process the values using POS-tagging and chunking. After processing, the heads of the chunks are looked up to find the correct class, for instance to determine if the entity is a a town, a body of water or something else.

The method described above also has some limitations. Sometimes, infobox data contain imprecise information, i.e. data that are somehow related with the expected values, but do not conform the exact syntax and semantics. Figure 5.3 gives an example of imprecise infobox data. It shows an entry of the river infobox in the German Wikipedia article about the river *Erms*. For this river, there is no appropriate value for the attribute *Großstädte (large cities)* which denotes larges cities that are crossed by the river. In such cases, editors sometimes put related information that is not compatible with the semantics of the attribute. In this case,

[8]http://de.wikipedia.org/wiki/-#Bindestrich-Minus which is the article about hyphen.

they put *Reutlingen*, a nearby large city. As a consequence, the semantics of the attribute is no longer clear and consistent.

In Chapter 6, we will introduce a further method that improves the quality of the extracted instances in the KB by using crosslingual filtering (Section 6.2.4).

5.4.2 Annotation

During the annotation step, each mention has to be annotated where a pair of entities from the knowledge base is encountered in a sentence. The challenging part is the matching between the entities of the knowledge base and the tokens of sentences. The selection and identification of certain entities in a text is not trivial. The success of distant supervision highly depends on the matching component which is a central method in the annotation approach. There is no global solution to face the issue of bad matching. In the next paragraph, different matching approaches are discussed before the results of our work are presented in Section 5.5.

As mentioned in Section 5.2 there are two different DS approaches: *redundancy-based DS* [Mintz et al., 2009] and *structure-based DS* [Riedel et al., 2010]. We adopt *structure-based DS* as our approach in this work.

We call the main entity of a Wikipedia article *focus entity* or e_{focus} and the second argument of the relation e_{obj}; e.g., if "Paris is located on the Seine" occurred in the Wikipedia article about *Paris*: e_{focus}=*Paris*, e_{obj}=*Seine*.

The instances in the KB are used to annotate relations in the unstructured Wikipedia text. An effective heuristic for high-precision automatic annotation is to only consider sentences that contain e_{focus}– considering that other sentences increase recall, but at a high cost in terms of precision. A second filter that increases precision is that we only consider mentions of an e_{obj} that are hyperlinked to e_{obj}'s Wikipedia article to avoid a disambiguation step for the named entities.

e_{focus} is – in contrast to e_{obj}– usually not represented as a hyperlink. Recognizing e_{focus} is difficult because the title often has add-ons that disambiguate the entity

article name	anchor text used in other articles to refer to the article
John_F._Kennedy	John F.
John_F._Kennedy	35. US-Präsidenten
John_F._Kennedy	John
John_F._Kennedy	John F. Kennedy: ?Ich bin ein Berliner?
John_F._Kennedy	John Fitzgerald Kennedy
John_F._Kennedy	John-F.-Kennedy
John_F._Kennedy	Kennedy
John_F._Kennedy	Kennedy, John Fitzgerald
John_F._Kennedy	Kennedys
Angela_Merkel	Angela Merkels
Angela_Merkel	Dr. Angela Merkel
Angela_Merkel	Merkel
Angela_Merkel	Merkel, Angela
Angela_Merkel	Regierung Merkel
Stuttgart	?
Stuttgart	Bad Cannstatt
Stuttgart	Freien Kunstschule Stuttgart - Akademie für Kunst und Design
Stuttgart	Kreisfreie Stadt Stuttgart
Stuttgart	Landeshauptstadt Stuttgart
Stuttgart	S
Stuttgart	Schrifstellerhauses Stuttgart
Stuttgart	Stadt Stuttgart
Stuttgart	Stadtkreis Stuttgart
Stuttgart	Stuttgart

Table 5.2: A list of anchor texts which refer to the Wikipedia articles: *John F. Kennedy*, *Angela Merkel* and *Stuttgart*. Wikipedia is not free of noise which leads to some unwanted extracted pairs (e.g. Stuttgart - ?).

(e.g., *Tom Jackson (politician)*). We address this problem by creating a lexicon of all hyperlink anchor names and all redirection pages for the entity used in Wikipedia (cf. [Wu and Weld, 2010]); this gives us an inventory of possible surface representations used for the entity. Table 5.2 and table 5.3 give an overview of the variance of strings which refer to the same entity either by a hyperlink or by a redirection page.

Finally, we perform a simple form of coreference resolution: we assume that all pronouns refer to e_{focus}. We have found that this simple heuristic has a high

article name	redirection name
John_F._Kennedy	John_Fitzgerald_Kennedy
John_F._Kennedy	John_F_Kennedy
John_F._Kennedy	John_F._Kennedy
John_F._Kennedy	Jack_Kennedy
Angela_Merkel	Angela_Merkel
Angela_Merkel	Angela_Dorothea_Merkel
Angela_Merkel	Angela_Kasner
Angela_Merkel	Angela_Dorothea_Kasner
Stuttgart	Stuttgart
Stuttgart	Landeshauptstadt_Stuttgart

Table 5.3: A list of redirection names which refer to the Wikipedia articles: *John F. Kennedy*, *Angela Merkel* and *Stuttgart*.

accuracy of between 95% and 100%[9]

Example

Figure 5.4 shows a complete DS chain for a sentence annotated for the municipality-county relation by using structured and unstructured data of Wikipedia. First, the structured data has to be analyzed. We call this step KB selection. For the case considered here, a true instance of the municipality-county relation set has to be identified in the infobox. Infoboxes describe attributes that are in relation to the main entity of the Wikipedia article. If we consider Figure 5.4, then *Gebroth* is the main entity of the article and the entry in the second row of the infobox (on the right side) denotes: Landkreis (county): *Bad Kreuznach*. This entry describes the municipality-county relation between *Gebroth* and *Bad Kreuznach*. In a second step, the extracted knowledge is used to annotate unstructured data[10]. We call this step the annotation phase. The arrows between text, page title and infobox in Figure 5.4 depict this annotation process. After this step the sentence: `Gebroth ist eine Ortsgemeinde in Landkreis Bad Kreuznach in Rheinland-Pfalz`

[9]We designed a simple experiment to prove our hypothesis. In two domains (person, toponyms) about 50 articles were picked at random and we checked for each appearing pronoun if our assumption that pronouns refer to the main entity is correct.

[10]All text passages in Wikipedia are unstructured data.

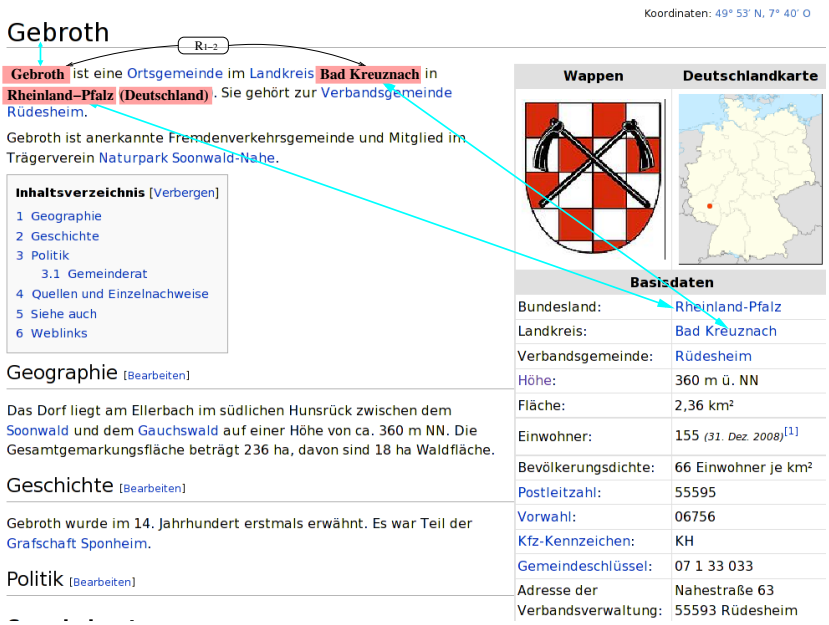

Figure 5.4: German Wikipedia article about *Gebroth*. All named entities in the first sentences are highlighted. For the annotation the structured information of the article name and the infobox are used. In this example the municipality-county relation between *Gebroth* and *Bad Kreuznach* is annotated.

`(Deutschland). (Gebroth is a municipality in the county Bad Kreuznach in Rhineland-Palatinate (Germany).)`[11] is annotated and can be used for training.

5.4.3 Positive and Negative Instances

In order to avoid overfitting and to enable good generalization, classifiers have to be trained on well-balanced resources. Therefore, not only the positive instances have an impact on the quality but also negative[12] instances influence the model of the classifier.

Our approach uses in most cases binary classifiers which decide if a pair of two

[11] For better readability we omitted any markup, however, this is necessary for machine processing.

[12] *Negative instance* means in this context that we also consider pairs of entities that are not members of our relation set. All these pairs that are not a member of a relation under analysis are negative instance which is equal to a complement set of all positive entity pairs.

entities is a member of a given relation or not. If only positive instances are used for training then the classifier is not able to distinguish between both cases. The main challenge is to find negative instances since knowledge bases have no entities for such non-relation-instances.

The assumption that any pair of entities that is not represented in the knowledge base is a negative instance is not adequate, since there is no guarantee for completeness of the KB.[13] The remaining solution is to use heuristics that consider constraint violations to decide if two entities compose a negative example. The simplest constraint is to consider the entities' types for a given relation. For instance, if we want to extract the BIRTHPLACE relation of a person, then we know that one entity has to be of type *person* and the other of type *location*. All pairs of entries that do not meet this requirement can be used as negative instances. Later in this chapter (Section 5.5.1), we will present a relation between suburbs and corresponding towns. In this case, our heuristic uses a more fine-grained distinction between the entity types. In this case we can create negative instances by using object of type *town*: for example, *Stuttgart* and *Esslingen* can not be in a SUBURBTOWN relation.

Such fine-grained distinctions are often not directly applicable, since there are no resources or classifiers with the high accuracy that is needed to avoid false annotations. By using Wikipedia text as an annotation corpus, we can circumvent this issue by using additional structured data (hyperlinks in the text) to get an entity classification on a fine-grained level.

5.5 Experiments

We carried out our experiments on two different relations. First, we consider a part-of relation because it is easier to train. Second, we consider a more complex relation in the geospatial domain.

[13] We have to deal with an open world assumption and we can never assume that a KB is complete.

5.5.1 Taxonomic Relations

The experiments presented here are carried out for two different relations from the geospatial domain, on German Wikipedia articles. All used relations are binary and contain two arguments which are toponyms. We focus on the textual representation of such relations if the complete relation is described into the same sentence. This allows us to avoid the need of any cross-sentence inference.

All experiments have in common that we split the data for evaluation and development purposes in three disjoint sets: training set, development set and evaluation set.

Before presenting the experimental setup and results, we give a brief overview of each evaluated relation. This includes the motivation for using this set of relations and the introduction of some characteristics of the relations.

- relation I : SUBURB - MUNICIPALITY
- relation II : RIVER - BODYOFWATER

Relation I describes the relation between a suburb or district ('Orts-/Stadtteile') and its linked municipality or city ('Gemeinde',). We use *municipality* as a technical term. In particular, a suburb/district is *not* a municipality, but is part of one. For relation II, the first argument contains rivers and the second argument bodies of water into which they flow. In this case, body of water represents any kind of river, lake or sea.

Motivation

One common scenario in DS is to use already existing annotated data (e.g. ACE 2005 [Doddington et al., 2004]) as gold standard. For our experiments, such data sets are not appropriate since there are no common available data sets for German. Another important aspect of our work is to show that DS is applicable to more

specific relations than provided in other DS work. The ACE 2005 data sets for instance provide only 7 main relations, which occur in corpora with high frequency. If we look at other DS work that is not using ACE 2005 data sets, we find that they provide often a large set of different relations [Hoffmann et al., 2010] which they annotate and extract automatically. But these relations have one feature in common: they all occur frequently in one of the Wiki-derived databases. This indicates that these relations are all quite popular and that they will also occur frequently in corpora. Furthermore, we do not consider relations which are modeled by any Wiki-derived KB (e.g. DBpedia) because such relations have already been used for DS. In contrast, we follow another research path by considering relations which are not or not prominently[14] modeled in Wiki-derived KBs.

Relation Characteristics

Table 5.4 lists the different characteristics of our relation experiments. In the case of the SUBURB-MUNICIPALITY relation, a simple infobox extractor can be used, since the relation is defined by exactly one matching entity in the infobox. For the other relations, a more sophisticated method is needed, since these relations are represented by a content which includes natural language text in the infoboxes. For the SUBURB-MUNICIPALITY relation, which is a required fact for suburbs, the first sentence of the Wikipedia article about the suburb defines such a relation in almost all cases. So, the remaining document has no impact and need not be considered. The RIVER-BODYOFWATER relation is not as important, and this fact is normally not defined at the beginning of the document. In that case, the whole document has to be used for the annotation process. Furthermore, coreference is often used when a term appears frequently in a document. This provides an additional challenge for the entity matching. Note that for the SUBURB-MUNICIPALITY relation this has no impact, because the first sentence always includes the focus entity of the document. Furthermore, this entity is also marked by structured data which can be

[14]Not prominent means that such relation has only a few (less than 20) instances in the KB.

used to correctly identify the complete entity. Again, for the RIVER-BODYOFWATER relation, coreference resolution is necessary.

characteristics	SUBURB-MUNICIPALITY	RIVER-BODYOFWATER
knowledge base population	simple infobox extraction	extended infobox extraction
considered text	first sentence	whole document
coreference resolution	no	yes

Table 5.4: Two classes of experiments which differ in 3 aspects.

Classification Task

We use a classification task (Section 2.1.5) to evaluate DS in real conditions. Our relation extraction task is modeled as a classification task which considers a pair of named entities and decides whether they occur in the requested relation or not. The classifier uses extracted features to perform this decision. The features belong to three different classes. The first class contains *token-based features* and their linguistic labels like part-of-speech, lemma, stem. In the second class, we have *chunks* that aggregate one or more tokens into complex units. *Dependency relations* between the tokens are represented in the third class.

Supervised Relation Extraction

As described in Section 3.2.5, we use the ClearTK [Ogren et al., 2008] toolkit, which is a UIMA component, for the relation extraction task. It contains wrappers for different machine learning suites. Our initial experiments showed that the Maximum Entropy classifier (Section 2.1.5) achieved the best results for our classification task. The toolkit provides additional extensible feature methods. Because we view self-annotation and fine-grained named entity recognition as our main contributions, but not feature selection, we only give a brief overview of the features we use.

The feature set F0 represents the baseline features in our experiments (Table

	linguistic effort	description
F0	none	BoW + distance + NE position (1st, 2nd,..)
F1	pos-tagging	POS and LEMMA
F2	chunk-parse	parent chunk
F3	dependency-parse	dependency paths between NEs

Table 5.5: List of feature sets used for the relation extraction experiments.

5.5). It is a simple bag of words (BOW) approach without any linguistic information. The words in a window size of three around both named entities which have to be classified are represented in this feature set. Additionally, the distance between both entities and the positions of both (counted in the number of words in the sentence) are modeled. Feature F1 is a window based bag-of-words feature that additionally considers the lemma and part-of-speech tags of tokens which have already been used in F0. Feature F2 is a phrase-based feature set that uses the parent phrases of both entities (maximally 2 levels). Feature set F3 is a representation of all possible dependency paths between the article's main entity and a target entity, where each path is represented as a feature vector. In most cases, more than one path is returned by the partial dependency parser (the used FSPar parser makes no disambiguation decisions) and included in the feature representation. Figure 5.5 depicts the dependency parser output of our example sentence. Each pair of square and circle with the same number correspond to one dependency. These possible dependency combinations give rise to 8 possible paths between the relation entities *Tauber* and "sie" (she) although our example sentence is a very simple sentence.

Our main analysis engine is a wrapper around the FSPar NLP engine as described in Section 3.2.4. A lexicon is used to mark all named entities in the first sentence of the Wikipedia article. We use heuristics to address spelling errors and variants (which occur frequently). Table 5.6 shows a sample sentence including POS and lemma tags. 4 named entities are found (NE_1...NE_4). All possible binary relations are built (R(NE_1,NE_2),R(NE_1,NE_3),...) for the classification step.

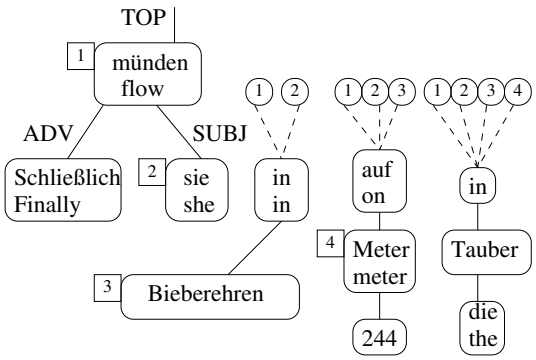

Figure 5.5: Visualization of the output of the FSPar reader for the sentence: `Schließlich mündet sie in Bieberehren auf 244 m in die Tauber.` (`Finally, it discharges in Bieberehren at 244 m above MSL into the Tauber.`)

token	POS	lemma
Gebroth	NE_1	Gebroth
ist	VAFIN	seinA
eine	ART	ein
Ortsgemeinde	NN	Orts#@gemeinde
im	APPRART	in
Landkreis	NN	Land#@kreis
Bad_Kreuznach	NE_2	Bad_Kreuznach:Stadt
in	APPR	in
Rheinland-Pfalz	NE_3	Rheinland-Pfalz:Region
Deutschland	NE_4	De:Region|Deutschland:H
.	$.	.

Table 5.6: Tagged output of the FSPar parser.

5.5.2 SUBURB-MUNICIPALITY Extraction

An example for the SUBURB-MUNICIPALITY relation is given by following sentence:

- *Sohlbach* is a suburb of *Netphen* in the county *Siegen-Wittgenstein* in *Nordrhein-Westfalen* with 143 inhabitants.

The pair of *Sohlbach-Netphen* is a positive instance of that relation. All other possible pairs between entities (*Sohlbach-Siegen-Wittgenstein*, *Sohlbach-Nordrhein-Westfalen*, *Netphen-Siegen-Wittgenstein*, *Netphen-Nordrhein-Westfalen* and *Siegen-Wittgenstein-Nordrhein-Westfalen*) are negative instances.

Classifier	features	precision	recall	FP	FN
1	F0	79.0%	55.7%	279	833
2	F0+F1	92.4%	89.3%	138	202
3	F0+F2	90.2%	89.5%	182	198
4	F0+F3	97.7%	97.4%	43	48
5	F0...F3	98.8%	97.8%	23	41

Table 5.7: Results of different feature combinations on the test set

For our experiments we considered German Wikipedia articles about suburbs and municipalities. 9037 articles met our criteria concerning the completeness of the infoboxes and the integrity of the first sentence (the main entities of the infobox must be mentioned in the first sentence of the article) and are concatenated to form a small corpus. In the next step the structured information is used to annotate the unstructured textual corpus with the two relations defined above. To support the supervised approach we split this annotated corpus into three parts to enable a clean evaluation. The first 60% (5357 sentences) are used as the training set during development. The next 20% (1840 sentences) are used for the evaluation in the development phase. The remaining 20% (1840 sentences) are the test set and used for evaluation.

5.5.3 Evaluation of SUBURB-MUNICIPALITY Relation Extraction

Table 5.7 shows the application of different feature combinations. The results confirm the need for linguistic analysis to successfully solve the extraction task (classifier 5). Surprisingly, the simple window-based feature (classifier 3) performs better than the chunk based-feature (classifier 2). Classifier 4 demonstrates the importance of dependency parsing for successful IE in German. Classifier 5 combines all features. This halves the number of false positive cases in comparison to the already well working classifier 4.

Finally, we give four examples from the development set to illustrate the performance of the F0+F1+F2+F3 classifier (5). The correct entities for the relations are

printed in bold. Additionally, we marked in parenthesis at the beginning of each example the evaluated result of our classifier. FN stands for a missed relation and TP for a correctly recognized relation.

1. **(FN) Jesingen** ist eine ehemals selbständige Gemeinde im Landkreis Esslingen und gehört seit dem Jahre 1974 zur Großen Kreisstadt **Kirchheim unter Teck**. (**Jesingen** is a formerly independent municipality . . . and belongs since 1974 to **Kirchheim unter Teck**.)

2. **(TP) Ostdorf** war bis 1971 eine Gemeinde im Zollernalbkreis in Baden-Württemberg und ist heute ein Stadtteil mit Ortschaftsrat von **Balingen**. (**Ostdorf** was until 1971 a municipality . . . and is today a suburb of **Balingen**.)

3. **(FN) Hülhoven** liegt in NRW im Regierungsbezirk Köln und ist ein Ortsteil **Heinsbergs**, der westlichsten Kreisstadt Deutschlands. (**Hülhoven** lies in NRW . . . and is a suburb of **Heinsberg** . . .)

4. **(TP) Hürtgenwald** ist eine Gemeinde in Nordrhein-Westfalen, Deutschland und gehört zum Kreis **Düren**. (**Hürtgenwald** is a municipality . . . and belongs to the county **Düren**.)

Sentences 1 and 2 state that a suburb *was* a former municipality. In the first case only the word *ehemals* (formerly) indicates this fact, and the sentence is not classified correctly. In the second case, the past tense of the main verb indicates the "past" meaning, and this sentence is correctly classified. Sentence 3 shows that coordinations are sometimes not handled correctly by the classifier. Sentence 4 is an example of a difficult coordination (large distance between elements of the relation) being handled correctly.

5.5.4 RIVER-BODYOFWATER Extraction

We use the RIVER-BODYOFWATER relation between the two rivers *Gollach* and *Tauber* to describe the DS-annotation steps.

engine	features	F_1	recall	precision
FSPar	F1	64.9	79.0%	55.7%
FSPar	F1, F2	89.6	90.2%	89.5%
FSPar	F1, F2, F3	98.3	98.8%	97.8%

Table 5.8: Results of different feature combinations on the test set forSUBURB-MUNICIPALITY relation

Figure 5.6 depicts a part of the infobox for the German Wikipedia article about the river *Gollach*. For this relation the attribute "Mündung" (mouth) is relevant. The value contains unstructured information (i.e., text, e.g. `bei Bieberehren (at Bieberehren)`) and structured information (the link from *Bieberehren* to its Wikipedia page). The relation we want to extract is that the river *Gollach* flows into the river *Tauber*. Figure 5.7 shows the textual content of the *Gollach* article. We have highlighted all relevant named entities for the DS-annotation process. This includes the name of the article and instances of the pronoun "sie" (she) referring to *Gollach*. Our matching algorithm identifies two sentences as positive samples for the relation between *Gollach* and *Tauber*:

- (i) Die *Gollach* ist ein rechter Nebenfluss der *Tauber* in Mittel- und Unterfranken. (The *Gollach* is a right tributary of the *Tauber* in Middle and Lower Franconia.)
- (ii) Schließlich mündet *sie* in Bieberehren auf 244 m in die *Tauber*. (Finally, *it* discharges in Bieberehren at 244 m above MSL into the *Tauber*.)

For the extraction of the RIVER-BODYOFWATER relation, the infobox processing is more difficult. We have to handle more attributes because there is schema drift between the different users. This means that attributes are not uniformly used by the Wikipedia editors. It is hence necessary to merge information coming from different attribute values. The other difficulty is the usage of mixed contents in the values. Another main difference to the SUBURB-MUNICIPALITY relation is that the RIVER-BODYOFWATER relation is often not mentioned in the first sentence (which usually gives a short definition of the object denoted by the the main entity). Thus,

Gollach

Die Gollach bei Aub-Baldersheim

Daten	
Lage	Bayern (Mittelfranken, Unterfranken), Deutschland
Gewässerkennzahl	DE: 2462
Länge	29,06 km
Quelle	zwischen den *Markt Nordheimer* Ortsteilen *Herbolzheim* und *Ulsenheim* am südwestlichen Rand des Steigerwaldes 49° 33′ 34″ N, 10° 18′ 43″ O
Quellhöhe	337,6 m
Mündung	bei Bieberehren (am Ende des Gollachtals) in die Tauber 49° 31′ 14″ N, 10° 0′ 2″ O

Figure 5.6: Infobox of the German Wikipedia article about *Gollach*.

Gollach

Die **Gollach** ist ein rechter Nebenfluss der **Tauber** in Mittel- und Unterfranken.

Die **Gollach** ist etwa 29 km lang und entsteht zwischen Herbolzheim und Ulsenheim am südwestlichen Rand des Steigerwaldes auf 337,6 m. **Sie** fließt in westlicher Richtung an Ulsenheim (Markt Nordheim), der Kleinstadt Uffenheim und den Orten Gollachostheim (Gollhofen), Lipprichhausen (Hemmersheim) und Hemmersheim vorbei zur Kleinstadt Aub. Nach Aub zieht **sie** dann in südwestliche Richtung und schneidet sich dabei tief in das nach ihr benannte Gollachtal ein. Schließlich mündet **sie** in **Bieberehren** auf 244 m in die **Tauber**.

Die Landschaft um die **Gollach** wird Gollachgau genannt; nach ihr heißen auch die Orte Gollhofen und Gollachostheim. Das Einzugsgebiet umfasst ca. 160 Quadratkilometer, nach Norden und Osten wird es von denen einiger Nebenflüsse des Mains begrenzt, insbesondere der Aisch, im Süden und Westen konkurrieren andere Nebenflüsse der **Tauber** mit ihr.

Figure 5.7: Textual content of the German Wikipedia article about *Gollach*. All named entities which are relevant for the RIVER-BODYOFWATER relation are highlighted. This article contains two instances for the relation between *Gollach* and *Tauber*.

the DS-annotation method has to deal with the more complex sentences that are common later in the article. This also contributes to a more challenging extraction task.

Our RIVER-BODYOFWATER relation corpus consists of 3000 DS-annotated articles. Again we split our annotated set into three parts: (60%) are used as training corpus, (20%) are used as development corpus and the remaining 20% are used for the final evaluation and were not inspected while we were developing the extraction algorithms. As shown in table 5.4 (page 119) the population of the KB which is used for DS requires more effort. Also the annotation step is harder because not only the first sentence of the article is used. The following section shows in more detail how this is done for the RIVER-BODYOFWATER relation.

engine	features	F_1	recall	precision
FSPar	F1	51.8%	56.6%	47.8%
FSPar	F1,F2	72.1%	68.9%	75.7%
FSPar	F1,F2,F3	78.3%	74.1%	83.0%
OpenNLP	F1	48.0%	62.8%	38.8%
OpenNLP	F1,F2	73.3%	71.7%	74.7%

Table 5.9: Results of different feature combinations on the test set for RIVER-BODYOFWATER extraction

5.5.5 Evaluation of the RIVER-BODYOFWATER Extraction

Table 5.9 shows the performance of the extractor described in Section 5.5.1. In contrast to the previous experiment we used two different NLP components in our pipeline. The OpenNLP pipeline[15] represents a component which is freely available, but the results show that the performance is not as good as with the FSPar pipeline. This is also true if we only consider basic processing steps on part-of-speech and lemma annotation level. As in the case of SUBURB-MUNICIPALITY extraction the classifier that uses all features of the FSPar pipeline, including dependency features, performs best.

5.5.6 Evaluation of DS Annotation Accuracy

To evaluate the quality of the annotations which are done by our DS approach, we randomly selected one set of 100 DS-annotated articles from each data set and labeled these sets manually. These annotations are used to calculate the inter-annotator agreement between the human annotated and machine annotated (DS) instances. We use Cohen's κ as a measure and get a result of 1.00 for the SUBURB-MUNICIPALITY relation. For the RIVER-BODYOFWATER relation we got a κ-value of 0.79, which also indicates good agreement.

We also use a gazetteer to evaluate the quality of all SUBURB-MUNICIPALITY relations that were extracted for our DS-annotated training set. The accuracy is

[15]http://opennlp.apache.org/

nearly perfect (only one single error), which is good evidence for the high quality of Wikipedia.

Required Size of DS-annotated Training Set

The performance of a supervised system depends on the size of the training data. In the DS-annotation step a minimum of instances has to be annotated, but it is not necessary to DS-annotate all available articles.

We reduced the number of articles used in the training size to test this hypothesis. Reducing the entire training set of 9000 (respectively, 3000) DS-annotated articles to 1000 reduces F1 by 2.0% for SUBURB-MUNICIPALITY and by 2.4% for RIVER-BODYOFWATER; a reduction to 100 reduces F1 by 8.5% for SUBURB-MUNICIPALITY and by 9.3% for RIVER-BODYOFWATER (compared to the 9000/3000 baseline).

5.6 Summary and Discussion

[Wu and Weld, 2007] observed schema drift in their work: Wikipedia authors do not not use infobox attributes in a consistent manner. However, we did not find schema drift to be a large problem in our experiments. The variation we found can easily be handled with a small number of rules. This can be due to the fact that the quality of Wikipedia articles improved a lot in the last years through the introduction of automatic maintenance tools like bots[16]. Nevertheless, the development of DS-annotation for a new relation type requires some manual work. The developer has to check the quality of the extraction relations in the infoboxes. This can lead to some additional adaptation work for the used attributes such as merging or creating rules. However, a perfect coverage is not required because the extraction system is only used for training purposes; we only need to find a sufficiently large

[16] See en.wikipedia.org/wiki/Wikipedia:Bots. The edit history of many articles shows that there is a lot of automatic maintenance by bots to avoid schema drift.

number of positive training instances and do not require exhaustive labeling of all articles.

It is important to note that considering partially found relations as negative samples has to be avoided. Wrong negative samples have a generally unwanted impact on the performance of the learned extraction model. A developer has to be aware of this fact. In one experiment, the learned classifiers were applied to the training data and returned a number of false positive results – 40 in case of the RIVERBODYOFWATER relation. 31 of these errors were not actual errors because the DS-annotation missed some true instances. Nevertheless, the trained model recognizes these samples as correct; this could perhaps be used to further improve the quality of DS-annotation. Manually labeled data also includes noise and the benefit of DS-annotation is substantial when the aim is to build a fine-grained relation extraction system in a fast and cheap way.

The difference of the results between the OpenNLP and FSPar engines are smaller than expected. Although sentence splitting is poorly done by OpenNLP the effect on the extraction result is rather small. Another crucial point is that the lexicon-based named entity recognizer of the FSPar engine that was optimized for named entities used in Wikipedia has no significant impact on the overall performance. Thus, a basic set of NLP components with moderate error rates may be sufficient for effective DS-annotation.

This chapter described a new approach to developing and implementing an end-to-end system to extract fine-grained geospatial relations by using a supervised machine learning approach without expensive manual labeling. Only some manual work has to be done to find the right attributes in the infoboxes. The matching process between infoboxes and text is not in all cases trivial and for some attributes additional rules have to be modeled. Our result showed that DS can be successfully transferred to German. We also presented the usage of DS in a fine-grained setting. Finally, our relation extraction approach showed that linguistically motivated features improve the quality of the extraction results.

Chapter 6

Crosslingual Distant Supervision

One important requirement for distant supervision is that structured and unstructured data have to be represented in the same language. This is a bottleneck since most available knowledge bases which are the typical representatives of structured data are represented in English. This fact again restricts the generation of new annotated resources by DS approaches to English, which is contrary to the idea of creating more annotations for languages which have not enough manually annotated data. In this chapter, we tackle this issue by a novel method we call crosslingual distant supervision (CDS). To enable DS for languages that have not enough structured data to perform the usual DS approach we use an additional language which we call *pivot language*. This pivot language is used as a source to provide enough structured data to allow DS. In more detail, a KB in the pivot language is used to annotate text in a target language. The approach has two benefits compared to standard DS: higher accuracy and increased coverage.

Additionally, we will provide a method that we call crosslingual filtering or *CL-filtering* that occurs during label transfer from the pivot to the target. The idea is that training examples are generated only if pivot language KB, pivot language raw text, and translation are consistent with respect to the training instances. This makes it unlikely that erroneous training examples are generated. We will present

extensive experimental evidence for the benefits of CL-filtering in our following evaluation.

Crosslingual DS increases coverage in cases where the target language KB has no coverage of a relation or less than the pivot language KB. Differences in coverage are widespread because KBs in any given language tend to lack instances of the relations that are less relevant to the region or culture where the language is spoken – e.g., *Islais Creek* in *San Francisco* has no French Wikipedia article and the Parisian *rivulet Grange-Batelière* has no English Wikipedia article. There also exist differences in the coverage of KBs that are due to the community-based and sometimes random nature of what is modeled; e.g., there is no comprehensive German KB for the SPOUSE-OF relation.

Because of these differences in coverage of relations in KBs and Wikipedias across languages, crosslingual methods have a great potential of increasing coverage. By extending monolingual DS crosslingually, the advantages of DS (automatic generation of large training sets for supervised machine learning) become available to languages that currently do not have good KBs for many or most relations.

Note that this chapter is largely based on already published work: [Blessing and Schütze, 2012].

6.1 Motivation

In the chapter 5 we already introduced many DS approaches [Craven et al., 1999, Mintz et al., 2009, Riedel et al., 2010, Blessing and Schütze, 2010b].

All this previous work on DS has in common that knowledge base (structured data) and text (unstructured data) must be in the same language. Our work is the first to show that crosslingual methods can be used to widen the applicability of DS to many additional languages. This offers novel opportunities to improve the quality of DS. Our filtering step is only a simple example of how to use the potential

power of crosslinguality for DS.

Significant work on crosslingual resources has been conducted in the Cross-Language Evaluation Forum (CLEF) [Peters, 2001]. Wikipedia is a major resource for this type of crosslingual research. Other relevant crosslingual research includes [Adafre and de Rijke, 2006] on generating parallel corpora; [Silberer et al., 2008] on creating a multilingual named entity resource; and [Kim et al., 2010] on projecting annotations to a target language, a method that, in contrast to crosslingual DS, can only be applied to aligned parallel corpora. Our work for the first time uses crosslingual information for DS.

The two benefits of crosslingual DS are of particular importance for what we call *complex relations*, which are harder to learn and extract than the simple relations that most work on relation extraction has focused on. We investigate SPOUSE-OF 'person *X* is spouse of person *Y*' as a representative of a simple relation; and RIVER-TOWN 'settlement *X* is on river *Y*' as a representative of a complex relation. There is no simple categorical property that distinguishes simple and complex relations. Rather, there are a number of dimensions along which relations vary; and RIVER-TOWN is more complex on each of those dimensions than SPOUSE-OF.

One dimension is that marriage imposes a strict constraint on the categories of its arguments: only human beings can be married. RIVER-TOWN is more complicated in that there are a number of related, but distinct relations between geopolitical entities and bodies of water; e.g., the *Rhine* flows through the *Netherlands* or *Cleveland* is located on *Lake Erie*. When recognizing an instance of RIVER-TOWN, the extraction system must verify the correctness of the categories of the two arguments. Ontological variability of arguments also causes higher variability in the linguistic means used to express the relation: different words will be used to express that a village *is located on* the *Nile* vs that *Islais Creek is in San Francisco*.

Another important dimension is that marriage is a central fact of life. If a person is the main subject of a Wikipedia article, then it is likely that their spouse(s) will be described as well. This is less likely for RIVER-TOWN: many Wikipedia articles

about rivers do not contain an exhaustive list of the towns on that river. This means that SPOUSE-OF is more predictable and has more training material for DS than *river-town*.

Prominence of a relation and ontological simplicity have two further consequences that distinguish simple and complex relations. First, simple relations are more likely to have a consistent and formally correct modeling in Wikipedia infoboxes. This is the case for SPOUSE-OF in the English Wikipedia. In contrast, RIVER-TOWN is modeled in a number of different ways in infoboxes: attribute names are inconsistent and attribute values are often difficult to parse – e.g., if a value is a list of strings that are not hyperlinked.

Second, freely available KBs provide high-quality data for DS for simple relations, but often do not for complex relations. Again, this is the case for SPOUSE-OF (which has good coverage in Freebase) and RIVER-TOWN (which is not systematically covered in any existing KB). The rigorousness of infoboxes and the availability in KBs are of course highly correlated because KBs are to a large extent based on infoboxes.

Most work on DS and unsupervised relation extraction has been done on simpler relations like SPOUSE-OF. We will take advantage of this prior work and use an existing gold standard for SPOUSE-OF. In contrast, little work has been done on complex relations like RIVER-TOWN. One of the contributions of this chapter is that we systematically conduct experiments for both types of relations. We will show below that the benefits of crosslingual DS (in particular, better coverage and higher accuracy) are of particular value for complex relations. Complex relations have fewer instances that can be automatically labeled using only the target language; hence they benefit from the additional automatic labels provided by the pivot language. Crosslingual DS counteracts the lack of noisy data by filtering out incorrect labels.

Section 6.2 describes the three steps of crosslingual DS. Experiments on a simple and a complex relation are described in Section 6.3 and evaluated in Section 6.4.

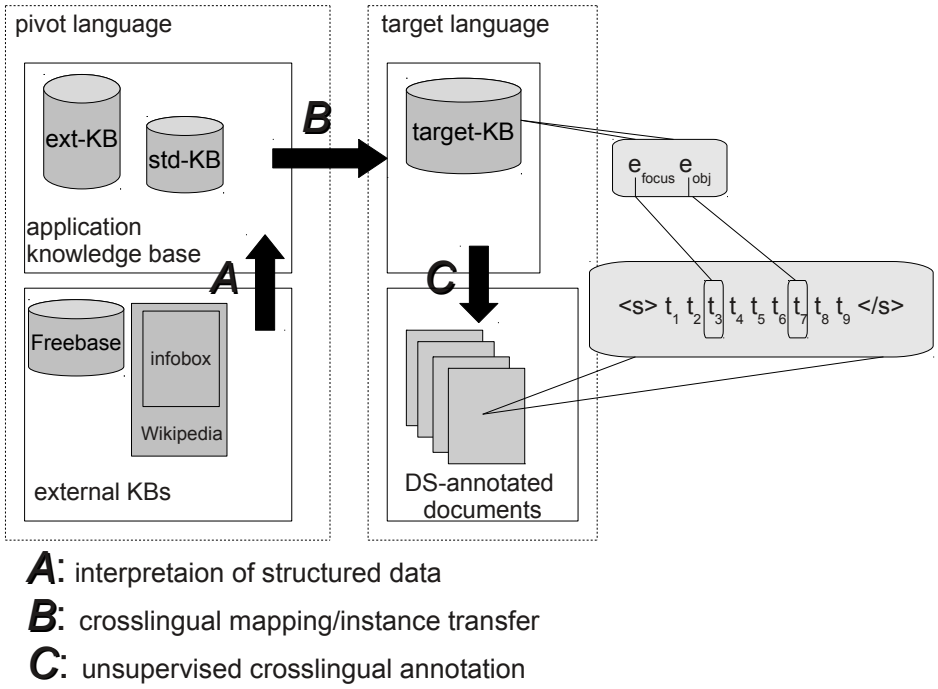

Figure 6.1: The three steps of crosslingual DS (A, B, C), discussed in detail in Sections 6.2.1–6.2.3

Section 6.5 states our conclusions.

6.2 Crosslingual DS Method

Figure 6.1 shows the three steps of crosslingual DS. After selecting a pivot language, we extract instances of the requested relation from an appropriate KB (Section 6.2.1, A in Figure 6.1); the KB can be a publicly accessible KB like Freebase, Wikipedia infoboxes or another (semi)structured data source. Depending on the quality of the extracted instances, this data is stored in the two different application specific knowledge bases, standard(std)-KB and extended(ext)-KB, to be described in Section 6.2.1. In the transfer step (Section 6.2.2, B), a crosslingual mapping is used to translate the instances and transfer them to target-KB. In the final step

(Section 6.2.3, C), the instances in target-KB are used to label relations on the sentence level in the target language (right half of figure Figure 6.1, labeling step is magnified: "$e_{\text{focus}} e_{\text{obj}}$" and "$< s > t_1 \ldots t_9 < /s >$").

6.2.1 Interpretation of Structured Data

We first need to select a pivot language for which a KB with good coverage of the relation exists. Infoboxes in Wikipedia are often a good choice because Wikipedia's more than 200 language editions contain a wide range of fine-grained knowledge for many regions and cultures. English, French, German and Spanish are particularly well suited as pivot languages because their Wikipedia editions define a wide variety of infoboxes that are instantiated in a large number of articles. Of the languages that provide an infobox of the right type, we select the language with the highest coverage. This is then the source of arrow A in Figure 6.1.

Once the pivot language KB has been identified, we populate the application KBs std-KB and ext-KB. std-KB is used for clean data with clear semantics. There are two types of KBs that have this property: (i) high-quality databases like Freebase, DBpedia and Yago and (ii) Wikipedia infoboxes that are consistently and cleanly modeled. We define consistent and clean modeling as (i) there is a single consistent attribute name for the relation used in the infobox and (ii) the value of the attribute is a single hyperlinked entity or a list of hyperlinked entities.

Instances of complex relations that are not cleanly modeled are stored in a separate database, ext-KB. Complex relations are usually not covered in public KBs. Since they are not cleanly modeled in infoboxes, automatic extraction (which produces almost error-free results for simple relations like SPOUSE-OF) is error-prone. One reason they are not cleanly modeled is schema drift: no consistent naming scheme for the attributes of the relation is enforced. More details about the challenges of knowledge base population were discussed in Section 5.4.1.

To populate ext-KB, we first create an inventory of all hyperlink anchor names

(the *anchor inventory*)[1] for rivers and towns in Wikipedia; this inventory covers possible surface representations used for the entities. The infobox extractor then extracts all phrases that match an entry in the anchor inventory. Each such anchor phrase found in the value of settlement attributes (for river articles) and river attributes (for settlement articles) is stored as one instance of RIVER-TOWN in ext-KB. This infobox extraction increases coverage compared to std-KB, but generates noisy instances in cases like *Erms-Reutlingen* (figure 5.3 on page 111).

Additionally, as described in detail below (Section 6.2.4), we use crosslingual filtering (*CL-filtering*) by filtering out pivot language labels that cannot be consistently transferred and applied to the target language. This method removes noisy instances from the training data and leads to a better performance of the CDS approach.

6.2.2 Instance Transfer

After populating std-KB and ext-KB, the pivot language instances are mapped to target-KB (B in Figure 6.1). We do this using interwiki links following a number of recent papers that show how to deal with uncertainty and asymmetry of links (e.g., [de Melo and Weikum, 2010], [Pateman and Johnson, 2010]). Since the Wikipedias are not parallel corpora, many instances cannot be transferred from pivot to target; e.g., no appropriate interwiki links may exist because one of the two entities does not have an article in the target.

Using Wikipedia for mapping is the easiest solution for this step. If Wikipedia is not sufficient, transliteration methods (e.g., [Hermjakob et al., 2008]) from statistical machine translation can be used to translate entities between different languages.

[1] Some examples for anchor names in Wikipedia are given in Table 5.2 on page 113.

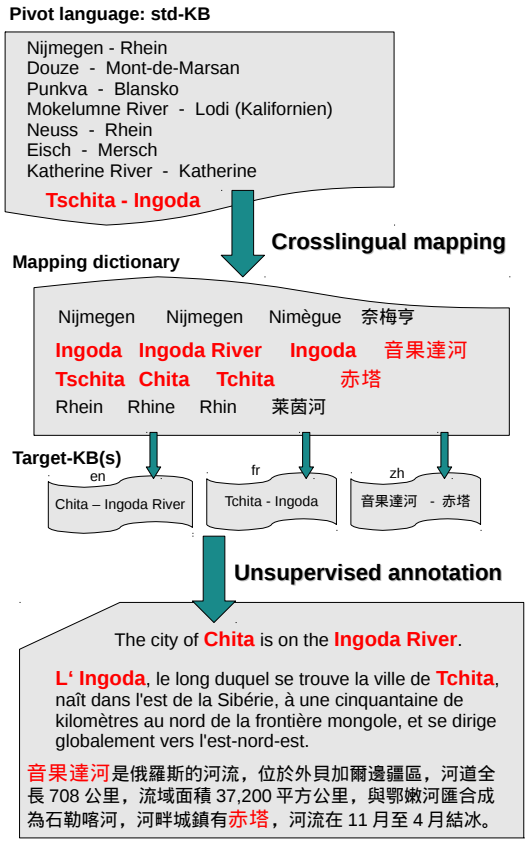

Figure 6.2: Illustration of crosslingual DS annotation for the RIVER-TOWN relation: relation between the Ingoda river and the town Chita. This instance is part of a German knowledge base. In the transfer step ("Crosslingual mapping"), the names are translated to English, French and Chinese by a mapping dictionary. In the "Unsupervised annotation" step, a sentence in each of the three target languages is automatically annotated.

6.2.3 Unsupervised Crosslingual Annotation

The crosslingual annotation process is analogous to monolingual annotation which was described in Section 5.4.2. The instances in target-KB are used to annotate relations in the unstructured text of the target language Wikipedia (C in Fig. 6.1).

We have so far only discussed how to create positive labels: target-KB facts only give rise to positive labels. We use a simple heuristic for negative labels building on [Elkan and Noto, 2008, Wang et al., 2007]. First, we infer the entity's type from

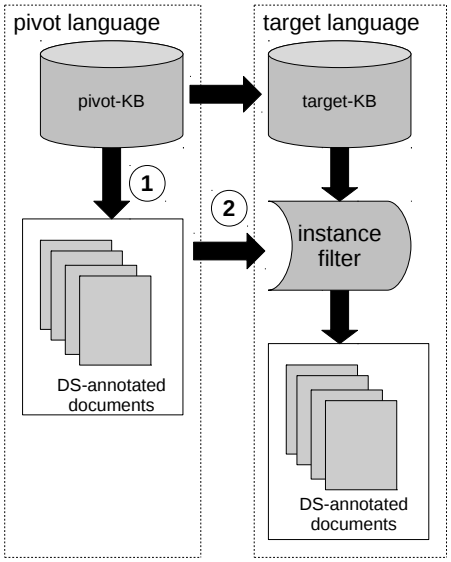

Figure 6.3: CL-filtering step: An instance (pair of entities) passes through the instance filter and is then used for annotation only if both pivot and target articles contain it.

the entity's article's infobox – e.g., an entity is a river if its article contains a river infobox. This heuristic works well because infoboxes are widely used in Wikipedia. An instance is negative if the inferred type of e_{obj} violates the constraints of the relation. This is the case if e_{obj} is not a settlement (if e_{focus} is a river) or is not a river (if e_{focus} is a settlement) for RIVER-TOWN; and if it is not a person for SPOUSE-OF.

6.2.4 CL-Filtering

We can now precisely state how CL-filtering works. A pair of entities is only used for annotation if both pivot and target contain a sentence that contains both e_{focus} and e_{obj}. In practice, we first annotate a corpus in the pivot language (left half of Figure 6.3). Then an additional filtering step is used during the annotation process

on the target language (right half of Figure 6.3). For each possible annotation in the target language, we check if the corresponding instance (pair of entities) is also annotated in the pivot language. The annotation is rejected if there is no corresponding instance. As a result, the annotated training corpus in the target language contains only annotated instances that also occur in the pivot language.

Our previously discussed example, the relationship between Erms and Reutlingen (figure 5.3 on page 111), is an example of the beneficial effect of CL-filtering. The river *Erms* and the city *Reutlingen* occur in the same sentence in the German Wikipedia and would be incorrectly labeled as a positive instance of RIVER-TOWN in a monolingual approach. But they do not occur in the same context in the English Wikipedia, and CL-filtering therefore blocks the labeling. This is an example of CL-filtering applied to the *pivot*; see Table 6.1, de1–de3. The evaluation below will show that CL-filtering is also beneficial for the *target*.

6.2.5 Illustrative Example

Figure 6.2 illustrates crosslingual DS as we have described it in Sections 6.2.1–6.2.3 for a concrete example. We consider the river-town relation between the Ingoda river and the city Chita. Since this relation is only modeled in the German Wikipedia, the only way to use distant supervision for a language other than German is a crosslingual approach. Since the relation occurs in the German Wikipedia, German is the pivot language.

Our goal is to annotate sentences in three target languages: English, French and Chinese. Since we have already given detailed examples for the knowledge base population step in Section 6.2.2, we start our walk-through with a pivot language KB (in this case: std-KB) in the upper part of Figure 6.2 that is already populated. All terms (e.g., “Nijmegen”, “Lodi (Kalifornien)”) refer to German Wikipedia articles. In the instance transfer step B (see Section 6.2.2, “Crosslingual mapping” in

the figure), we have to find the matching articles for each of the target Wikipedias. The center part of Figure 6.2 depicts some entries of the mapping dictionary. We used only interwiki links to generate this dictionary, but other transliteration methods (e.g. to transliterate between zh:音果達河 and en:Ingoda River) and translation methods (e.g. to translate between de:München and en:Munich) would also be appropriate. The populated target KBs after instance transfer are shown below the mapping dictionary.

In the bottom part of Figure 6.2 the corresponding text passages in the articles are annotated by the method described in Section 6.2.3. As a result, we end up in each of our three target languages with a sentence annotated by crosslingual DS. These annotated sentences can then be used to train a native relation extractor.

Algorithm 2 Crosslingual DS

for all $r_{\text{pivot}}(e_{\text{i}},e_{\text{j}}) \in \text{KB}_{\text{pivot}}$ **do**
  $\text{KB}^{+}_{\text{target}} \mathrel{+}= \{\text{map}(r_{\text{pivot}}(e_{\text{i}},e_{\text{j}}))\ \}$
end for
for all $s \in \text{CORPUS}_{\text{target}}$ **do**
  for all $(e_{\text{focus}},e_{\text{obj}}) \in s$ **do**
    if $r_{\text{target}}(e_{\text{focus}},e_{\text{obj}}) \in \text{KB}^{+}_{\text{target}}$ **then**
      $\text{ANNOTATED-CORPUS}^{+}_{\text{target}} \mathrel{+}= \{< s,e_{\text{focus}},e_{\text{obj}}>\}$
    else if wrong-type(e_{obj}) **then**
      $\text{ANNOTATED-CORPUS}^{-}_{\text{target}} \mathrel{+}= \{< s,e_{\text{focus}},e_{\text{obj}}>\}$
    end if
  end for
end for

Algorithm 2 gives pseudo-code for crosslingual DS. The first loop is used to map all entity pairs from the pivot language KB to the target language KB using the methods of Section 6.2.2. The next loop is used for the annotation process described in Section 6.2.3. We iterate over all sentences of the target corpus. In each sentence, all possible pairs of e_{focus} and e_{obj} are considered. If target-KB contains a pair, then a positive annotation is created. If the pair is not a member of target-KB and if the type of e_{obj} is not admissible for the corresponding slot of the relation, then a negative annotation is created.

Algorithm 3 Crosslingual DS extended by CL-filtering

for all $s \in \text{CORPUS}_{\text{pivot}}$ **do**
  for all $(e_{\text{focus}},e_{\text{obj}}) \in s$ **do**
    if $r_{\text{pivot}}(e_{\text{focus}},e_{\text{obj}}) \in \text{KB}_{\text{pivot}}$ **then**
      $\text{KB}^{+}_{\text{target-filtered}} \mathrel{+}= \{\text{map}(r_{\text{pivot}}(e_{\text{i}},e_{\text{j}}))\ \}$
    else if wrong-type(e_{obj}) **then**
      $\text{KB}^{-}_{\text{target-filtered}} \mathrel{+}= \{\text{map}(r_{\text{pivot}}(e_{\text{i}},e_{\text{j}}))\ \}$
    end if
  end for
end for
for all $s \in \text{CORPUS}_{\text{target}}$ **do**
  for all $(e_{\text{focus}},e_{\text{obj}}) \in s$ **do**
    if $r_{\text{target}}(e_{\text{focus}},e_{\text{obj}}) \in \text{KB}^{+}_{\text{target-filtered}}$ **then**
      $\text{ANNOTATED-CORPUS}^{+}_{\text{target}} \mathrel{+}= \{< s,e_{\text{focus}},e_{\text{obj}}>\}$
    else if $r_{\text{target}}(e_{\text{focus}},e_{\text{obj}}) \in \text{KB}^{-}_{\text{target-filtered}}$ **then**
      $\text{ANNOTATED-CORPUS}^{-}_{\text{target}} \mathrel{+}= \{< s,e_{\text{focus}},e_{\text{obj}}>\}$
    end if
  end for
end for

Algorithm 3 gives pseudo-code for the version of algorithm 2 that includes CL-filtering. We first collect all instances in the pivot language corpus, map them to the target and then store them in filtered target-KBs for positive instances ($\text{KB}^{+}_{\text{target-filtered}}$). For balancing reasons we also filtered the negative instances ($\text{KB}^{-}_{\text{target-filtered}}$). Without that filtering the amount of negative instance in the target languages was too high compared to the amount of positive instances. This had a negative impact on the classification results. The second part of the algorithm is similar to the second part of Algorithm 2: a pair $(e_{\text{focus}},e_{\text{obj}})$ of entities is used for a positive annotation only if $\text{KB}^{+}_{\text{target-filtered}}$ contains it and for a negative annotation only if $\text{KB}^{-}_{\text{target-filtered}}$ contains it. This ensures that each target annotation is justified by a target language sentence as well as a pivot language sentence.

6.3 Experiments

We conduct experiments for the complex relation RIVER-TOWN and the simple relation SPOUSE-OF. While it is a limitation of this thesis that we conduct experiments on only one relation of each type, it is important to consider that creating large test sets in four languages is a considerable effort.

6.3.1 RIVER-TOWN Relation

RIVER-TOWN is the relation between a river and a settlement – a village, town or city – located on the river, e.g., (Seine, Paris). This relation is needed in applications like the semantic web and in geographic information systems (GIS). As a GIS use case for RIVER-TOWN consider the 2010 cholera outbreak in Haiti. The source of the outbreak was the Artibonite river; in a case like this an international aid organization will need to obtain a complete list of all the settlements on the river. Even though there are geospatial datasets such as OpenStreetMap available, we are not aware of any that represent RIVER-TOWN. The crosslingual DS method introduced in this chapter addresses this problem.

For the experiments on RIVER-TOWN, we use the target languages Chinese, English and French. These languages have no (Chinese, English) or too few (French[2]) infobox data on RIVER-TOWN for monolingual DS to work. Freebase also does not represent this relation. In contrast, the German Wikipedia does represent RIVER-TOWN in infoboxes; thus, we can use it as a pivot. To address the non-rigorousness of RIVER-TOWN infoboxes discussed above, we use CL-filtering to increase the accuracy of automatic labeling in DS.

[2]The German Wikipedia edition contains about 8,600 instances of RIVER-TOWN and the French Wikipedia edition contains about 3,300 instances.

6.3.2 SPOUSE-OF Relation

As an example of a simple relation, we run experiments on the well-known SPOUSE-OF relation [Culotta et al., 2006, Ji et al., 2010]. In contrast to RIVER-TOWN, SPOUSE-OF is clear-cut and not open to interpretation in the cultures represented in Wikipedia. We ignore temporal aspects, so the actual relation we extract is "X was Y's spouse at any time in the past". The German Wikipedia has no structured information about SPOUSE-OF and does not include it in infoboxes. So we have to consult a pivot language to enable an automatic annotation of German by DS. In this case English is a good pivot since it provides enough SPOUSE-OF instances in Freebase that are linked to English Wikipedia articles.

6.3.3 Data

We use the JWPL framework [Zesch et al., 2008] to process Wikipedia[3] and extended it for recursively nested infoboxes. For RIVER-TOWN, we only consider Wikipedia articles about rivers and towns. The selection process is done in several stages. First we populate std-KB and ext-KB with instances from German infoboxes. These instances are split into three sets: training (80%), development (10%) and test (10%). The development set was used to develop the feature representation. To ensure a fair evaluation, we impose the constraint that any given entity may only occur in one of training, development or test. Next we collect for each entity in target-KB the corresponding Wikipedia articles in the target and pivot languages. The resulting training set contains 1580 Chinese, 4068 English, 2673 French and 5741 German Wikipedia articles for RIVER-TOWN. CL-filtering mandates that a given instance occur in both pivot and target language articles. Thus, each of the non-German articles has a corresponding German article.

[3]We used dumps of the Chinese (20110412), English (20110405), French (20110409) and German (20110410) Wikipedias.

The test set contains 330 Chinese, 1299 English, 858 French and 1302 German articles. English, French and German test sets were annotated by two annotators ($.77 \leq \kappa \leq .88$); for Chinese only one annotator was available.

For SPOUSE-OF, we use the set of [Culotta et al., 2006] as test set. It consists of 78 Wikipedia articles. We then extract 40,000 relation instances from Freebase[4] and remove all instances that occur in the 78 articles. The next steps are analogous to the river-town selection processing. A total of 3741 German and English articles pass CL-filtering and can be used as training set. To reduce the amount of manual annotation necessary for SPOUSE-OF, we translate German entity pairs found by the extractor in testing back to pairs of English entities using interwiki links. A German pair extracted from the test set is defined to be correct if and only if its translation via interwiki links is a member of the Culotta set.

6.3.4 Relation Extraction

We use the maximum entropy classifier Mallet [McCallum, 2002] for relation extraction. Each instance, consisting of a pair of named entities, is represented as a feature vector. Our feature representation is similar to that of [Zhou et al., 2005]. It includes information about the tokens in a window around and between the two entities, including POS tags and capitalization. Additional features capture the dependency path between the two entities and all tokens on this path.

We are following the same path as in Section 5.5.1. We only use features for the relation classifiers that are extracted from textual content. Therefore we can not exploit knowledge like hyperlinks as features.

We experimented with standard NER systems to replace the gold labels by automatic labels for toponyms. But the results are not encouraging since the standard training data provide too few instances in this domain and we get a low coverage.

[4]We use Freebase to ensure comparability with earlier work. We confirmed that we can extract a very similar data set directly from Wikipedia using simple extraction.

The same is true for the systems that we discussed in chapter 4 since we can only train specific models for specific toponyms like streets but we were not able to scale them to arbitrary toponyms since we had no facility to get the needed manual annotated data. These results show the potential of DS, since in a DS-setup, we can avoid manual annotations and furthermore we restricted our feature space to only features that are directly accessible without any need for manual data in previous steps[5].

6.3.5 Computational Cost

To reduce implementation effort we use a previously developed DS system (chapter 5) that is implemented in the UIMA framework (section 3.2.1) as the basis for the experiments reported in this chapter. The ClearTK toolkit (section 3.2.5) is our interface to Mallet [McCallum, 2002].

The step which is computationally most costly within crosslingual DS is the linguistic processing of the textual data: sentence splitting, tokenizing, tagging and parsing using tools described in [Schmid, 1995, Bohnet, 2010]. These processes together take several hours on a standard workstation. The linguistically annotated corpora are stored in a relational database which enables fast access and avoids the need to repeat any of the expensive linguistic processing. In comparison with the linguistic processing, the automatic annotation by DS is quite fast and takes only a few minutes. The training process and application of the model to the test data is also very fast and takes less than a minute.

6.4 Evaluation

We use precision, recall and F_1-score as evaluation metrics (cf. Section 4.1.6)

[5]The used pipeline uses tools like taggers or parsers which are trained on manually annotated data but in this cases the standard data are sufficient and no domain specific annotations are needed

		F_1		precision		recall	
		unfil.	fil.	unfil.	fil.	unfil.	fil.
zh1	human	40.0	**55.4**	50.0	62.1	33.3	50.0
zh2	std-KB	9.8	**34.8**	40.0	80.0	5.6	22.2
zh3	ext-KB	56.6	**64.3**	88.2	90.0	41.7	50.0
en1	human	72.3	**74.3**	88.5	86.7	61.2	65.1
en2	std-KB	61.0	**66.5**	89.0	87.8	46.4	53.6
en3	ext-KB	70.8	**75.5**	87.8	87.1	59.3	66.6
fr1	human	65.0	**70.2**	86.2	87.3	52.2	58.7
fr2	std-KB	60.4	**64.5**	89.1	86.6	45.6	51.4
fr3	ext-KB	67.1	69.8	88.6	87.5	54.0	58.1

Table 6.1: Performance of crosslingual DS on RIVER-TOWN for Chinese (zh1-zh3), English (en1-en3), and French (fr1-fr3).

6.4.1 Results for RIVER-TOWN

Table 6.1 shows results for RIVER-TOWN. To compare automatic DS labels with manual labels, we manually annotated the training data in the pivot (German). These instances are transferred to the target languages and classifiers are trained on the resulting three annotated sets. This is referred to as "human" in the table.

There are three main results of the experiments. (i) Crosslingual DS using ext-KB produces classifiers that have a performance comparable to or better than classifiers trained on labels transferred from manual labels in the pivot language: lines {zh,en,fr}1 vs {zh,en,fr}3 in Table 6.1. The largest performance decrease is for English unfiltered (unfil.) F_1 and recall (72.3 vs 70.8 and 61.2 vs 59.3); in all other cases the difference is smaller or crosslingual DS is clearly better. The fact that there is so little difference between manual pivot labels transferred to the target and automatic labels transferred to the target is evidence for the high quality of crosslingual DS labels.

(ii) The ext-KB approach (which exploits all infobox information, not just a clean and easily parsable subset) is superior in F_1 to the standard approach std-KB (which only uses clean infoboxes): lines {zh,en,fr}3 vs {zh,en,fr}2 in Table 6.1. Precision is slightly worse for English and French, but recall and F_1 are consistently better.

		F_1		precision		recall	
		unfil.	fil.	unfil.	fil.	unfil.	fil.
de1	human	57.4	**66.7**	87.5	84.6	42.7	55.1
de2	std-KB	37.2	**46.1**	89.9	85.7	23.4	31.5
de3	ext-KB	57.5	**66.0**	84.6	80.9	43.6	55.8

Table 6.2: Performance of unfiltered vs filtered "monolingual" DS. Filtering in this case filters German annotations using the English data.

		F_1		precision		recall	
		unfil.	fil.	unfil.	fil.	unfil.	fil.
1	de	70.1	71.0	61.8	69.1	81.0	73.1
2	en (pivot)	57.5	**62.4**	45.9	55.7	77.8	70.8

Table 6.3: Performance of crosslingual DS on SPOUSE-OF

Increases in F_1 are at least 5% and much larger in several cases. Chinese benefits the most because the number of transferable instances is much smaller than for the other two languages, resulting in a relatively small std-KB. This result suggests that the extended approach performs better than standard DS for complex relations like RIVER-TOWN – relations that are more likely to have non-rigorous infoboxes.

(iii) CL-filtering (fil.) further improves ext-KB F_1 by about 2% for French, 5% for English and 8% for Chinese (lines {zh,en,fr}3 in Table 6.1). Each filtered F_1 score that is significantly better than the corresponding unfiltered score at $p < .05$ is printed in bold (approximate randomization test [Noreen, 1989] as implemented in [Padó, 2006]).

The main effect of CL-filtering is that it increases the recall of the trained classifier because it eliminates many false negative training instances. Recall that a negative label is only applied if the two entities occur in the same sentence in both target and pivot Wikipedia articles. In the case of Chinese, CL-filtering increases both recall and precision because there is a larger number of false positives (similar to Erms-Reutlingen) that are eliminated.

Table 6.2 shows that CL-filtering is even beneficial in a "monolingual" setting – that is, if it is applied to pivot language text: $F_1 = 57.5$ (unfil) vs 66.0 (fil) on

line de3.[6] Again, the reason for this effect is that many noisy German instances are removed by CL-filtering because they do not have English equivalents. This confirms the potential of using crosslingual information in DS.

The success of ext-KB (compared to std-KB and human) on the three language pairs shows that the approach can be applied to different languages with equal success and suggests that the basic design of our method is language-independent.

6.4.2 Results for SPOUSE-OF

For SPOUSE-OF, we perform experiments for the target language German and use English as the pivot. Table 6.3 shows that (i) crosslingual DS supports the creation of a well performing classifier on the target language for which no KB exists that contains SPOUSE-OF; (ii) CL-filtering is beneficial for the pivot language (significantly) as well as for the target language.

These results suggest that crosslingual DS is a promising approach for both simple relations like SPOUSE-OF and complex relations like RIVER-TOWN.

6.4.3 Impact of Training Set Size

Figures 6.4 and 6.5 show that – except for very small training sets in English – filtered is consistently better than unfiltered in crosslingual DS ("filtered" curves vs "unfiltered" curves). The figures also demonstrate the high quality of the ext-KB labels: the ext-KB classifier is very close to (or better than) the classifier trained on human labels transferred to target-KB for training set sizes ≥ 350.

Figure 6.6 shows the performance of "monolingual" DS when pivot language an-

[6] To avoid confusion, we keep referring to German as the pivot and English as the target here even though their roles are partially reversed. We use the term "monolingual" to stress that one major benefit of crosslingual DS – positive instances are transferred from one language to the other – does not apply here. But even this monolingual variant of crosslingual DS is of course still crosslingual in the sense that it uses crosslingual filtering. We used English for CL-filtering here, but verified that F_1 is similar if French is used instead.

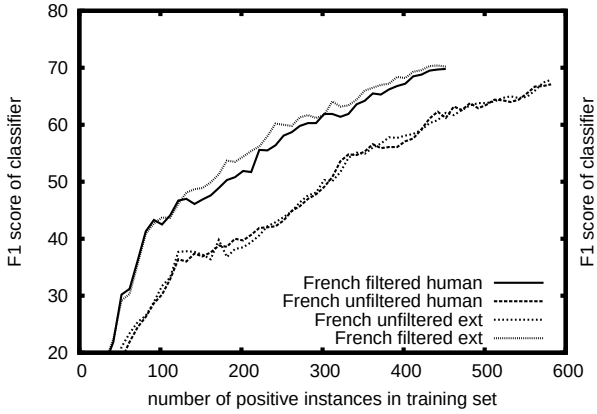

Figure 6.4: RIVER-TOWN: F_1 for French (target) as a function of training set size

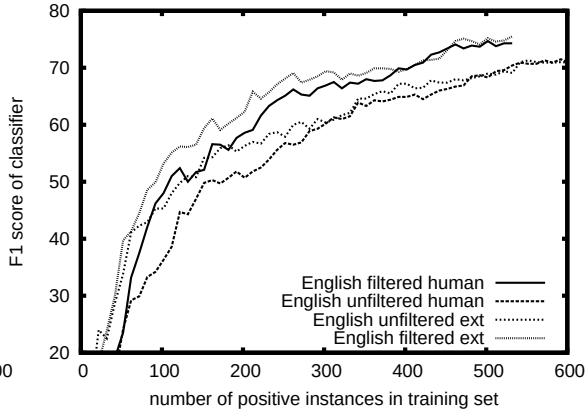

Figure 6.5: RIVER-TOWN: F_1 for English (target) as a function of training set size

notations (German) are filtered using a target language (English). In this setting a pair of entities is only used for annotation in monolingual DS if there is a sentence in the target language that also contains both entities. In our case we use German as the language for the monolingual annotation and the English corpus for filtering. The difference to the unfiltered – purely monolingual – approach is clearly observable. This shows that crosslingual distant supervision can also help to improve the quality of monolingual distant supervision.

Finally, we show the performance for the SPOUSE-OF data (Figure 6.7). There is also an improvement by using filtering, but the impact is lower than for RIVER-TOWN. The reason is that in comparison with RIVER-TOWN more sentences represent possible instance candidates since there is a large number of possible relations between persons. The current filtering approach cannot distinguish instances on such a fine-grained level. This problem will be addressed in future work.

6.4.4 Comparison to SMT

We cannot directly compare crosslingual to monolingual DS since there are no appropriate structured data available for monolingual DS in the target; e.g., there is

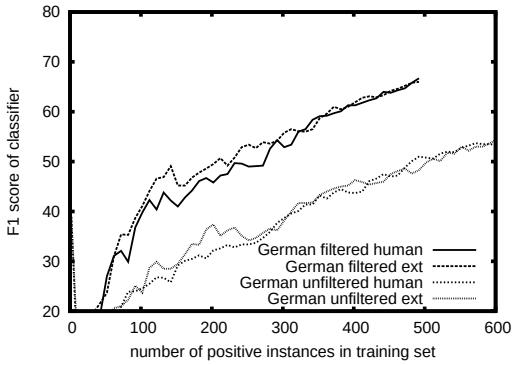

Figure 6.6: RIVER-TOWN: F_1 for German (pivot) as a function of training set size

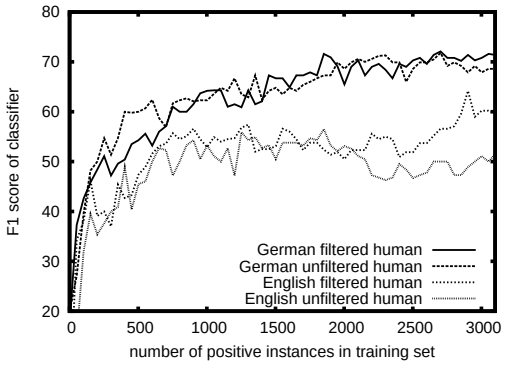

Figure 6.7: SPOUSE-OF: F_1 for English (target) and German (pivot) as a function of training set size.

no RIVER-TOWN structured data for English. However, we can automatically create a complete English monolingual setting for RIVER-TOWN using statistical machine translation (SMT) [Parton et al., 2009]. We used a subset of the test set, a set of 30 English Wikipedia articles, for evaluation. This set was translated into German using Google Translate. All e_{focus} and e_{obj} were manually marked in the translated sentences. Then the "human" model for German (Table 6.1, de1) was used as a classifier. In Table 6.4, we compare the result of this experiment ("SMT DS") with the classifier ext-KB (Table 6.1, en3), run on the test set of 30 articles. The F_1 score of crosslingual DS is about twice as large as the F_1 score of SMT-based DS.

If SMT produced perfect translations, we would expect performance to be comparable to crosslingual DS. The fact that SMT DS performs so much worse indicates that the quality of SMT is not yet good enough to be competitive with crosslingual DS. One reason is that the linguistic processing which is necessary for the feature extraction of the relation extraction system is not optimized for SMT-produced input which leads to the performance drop in recall.

	F_1	precision	recall
SMT DS	34.4	48.1	27.7
crosslingual DS	69.8	73.1	66.7

Table 6.4: SMT DS vs crosslingual DS for RIVER-TOWN

6.5 Summary and Discussion

In this chapter, we introduced crosslingual DS and evaluated it on four languages. We showed that it has two benefits: (i) higher accuracy of automatic annotation due to CL-filtering and (ii) better coverage because automatic labels can be transferred from the pivot to a target language. Benefit (i) is particularly valuable for complex relations that are modeled in non-rigorous infoboxes. Benefit (ii) applies to both simple and complex relations. We demonstrated that classifiers trained by crosslingual DS are competitive with classifiers trained monolingually on manual labels transferred to the target.

Our long term goal is to use crosslingual DS to automatically generate richly annotated resources for training machine learning algorithms on a large variety of tasks – with relation extraction being only one of these tasks.

Chapter 7

Summary of Main Results

This thesis provided several methods to identify and extract geospatial information from textual data. In this chapter we summarize the main contributions of our work.

In the last years, the wide availability of mobile devices increased the need for context-aware systems. Such systems have to model and represent the current state of the world, and this can not only be reached by means of physical sensors and manual input. Textual content is an important complementary source to obtain up-to-date information to track the current state of the world. To enable geospatial information extraction from texts we had to find anchor points which link entities from texts to real world objects. In this domain, toponyms (Section 2.2.1) are such bridges between both kinds of entities.

Our first step was to identify such toponyms, a task which can be seen as a special case of named entity recognition (Section 2.2.3). We identified two major drawbacks of NER for this task: Firstly, the classification operated by state-of-the-art NER systems is too coarse-grained for the geospatial domain. Many toponyms share their names with other toponyms of different types (e.g. cities vs. rivers). Secondly, most NER systems are designed and tested for English, but context-aware systems should not be restricted to the English speaking parts of the world. Therefore more

research is needed to test, adapt or redesign NER techniques for other languages. We were primarily interested in methods for German which is, in contrast to English, a morphologically rich language. We created a manually annotated corpus of street names for our test scenario (Section 4.1.2). Our experiments with an SVM-based recognition system which was trained on the annotated corpus provided two insights: the system reaches good F_1-scores (88.1-93.7%) if the task is simplified to a binary classification process: an entity either is a street or not. If the task is extended to a more fine-grained setting then the performance for several classes drops to F_1-scores under 50%. We concluded that the main reason for that loss is missing knowledge. To solve this issue, a more sophisticated approach is needed, which involves more knowledge, i.e. toponym resolution (Section 4.2). We therefore presented a model that exploits global knowledge and local knowledge which is dynamically updated during the resolution process. We implemented a small application (Section 4.2.3) using the OpenJump framework that allows to input text and to visualize the resolved toponyms on a map.

Another important aspect of geospatial information extraction is to link different data sets which represent overlapping information. We tackled the task of entity linkage for toponyms (Section 4.3) by combing features from two distinct research communities which are used by a supervised classifier. We took linguistics-based features from the NLP-community which model the names of the toponyms, and we took geometry-based features from the geography-community to model spatial aspects of the toponyms. The feature engineering for such a hybrid approach can be very time consuming. We used techniques from visual analytics to simplify that task. These methods helped us to find annotation errors which improved the used data sets and helped us to find the best features.

We used the previously described findings to implement a proof of concept system that allows the acquisition of vernacular names and a link to the respective region from text in web pages (Section 4.4).

One of our overall results is that all provided methods need large amounts of an-

notated resources to perform well. In the second part of this thesis we introduced an approach which compensates this drawback by using an automatic method to create annotated resources. This approach is called distant supervision (DS, Section 5.4). In our scenario we investigated DS for relation extraction since the resolution of toponyms is not fine-grained enough if we want that context-aware systems benefit from our methods. Again, we had to generalize and adapt the common DS approach which is originally designed for the English language in order to apply it to German data. Wikipedia is a good resource to enable DS for German: it provides enough structured data which can be used as a knowledge base and a rich spectrum of textual content which can be annotated. We exploited the hyperlinks in Wikipedia articles to avoid error-prone string matching, which increases the annotation quality. This is especially important for German since names can be inflected, which makes the automatic annotation process even more challenging. Our studies also showed that the knowledge bases which are typically used for DS, like DBpedia or Freebase, have not enough coverage in the geospatial domain. We proposed a new method (Section 5.4.1) that allows for a better extraction of structured data from Wikipedia infoboxes. Our experiments (Section 5.5) on two geospatial relations (SUBURB - MUNICIPALITY and RIVER - BODYOFWATER) showed that the DS approach reaches the same quality as a method which is trained on a manually annotated data set.

One drawback of standard DS is that the knowledge base and the textual content have to be in the same language. This is an obstacle if we try to extract region-specific information about foreign countries where the potentially usable texts are not in a language which provides rich knowledge bases. As a solution we developed crosslingual DS (Section 6.2) which allows us to apply DS to languages without knowledge bases. Our approach consists of three methods: (i) interpretation of structured data, (ii) instance transfer, and (iii) unsupervised crosslingual annotation. Our experiments in four languages (Chinese, English, French, and German) showed the success of crosslingual DS. As an extension we also invented crosslingual filtering (Section 6.2.4) which provides a method to increase the annotation

quality. This has a positive impact on the extraction results (Section 6.4.1). Additionally, we provided a comparison (Section 6.4.4) to a state-of-the-art crosslingual extraction system which uses SMT-based techniques. The results showed that the naively trained system which was enabled by crosslingual DS outperforms the SMT-based extraction system.

As a conclusion, we think that the provided results give additional evidence that DS is an important contribution to overcoming the high cost of creating training sets, one of the main impediments to using machine learning more widely in NLP today.

Outlook

Digital humanities (DH) applications for language technologies are an emerging topic in NLP research. Many methods of this thesis, which were presented for the extraction of geospatial information from text to assist context-aware systems, work also for DH-related textual analysis. Moreover, the recognition and resolution of toponyms in historical or fictional texts is even more challenging, since they are not mentioned in standard gazetteers. Nevertheless, such data, to some extent, are also represented in special knowledge bases, which enables the usage of the described DS approaches.

Over the last years, many of the presented linguistic tools (e.g. TreeTagger, Stanford NER) of our pipeline have also become available as web services. A future topic is how to make such services adaptable, such that they can be applied to new domains. The integration of DS methods into web services for named entity recognition and information extraction can be an appropriate solution.

We showed that structured data in Wikipedia are often not rigid enough to be used as an input for computational tasks. Wikidata[1] has means to counteract such noisy structured data by providing a centralized multilingual knowledge base. DS

[1]Wikidata is a project of the Wikimedia Foundation that started in 2012. http://www.wikidata.org

can benefit from this project in two ways. Firstly, the quality of structured data will increase. Secondly, the amount of available structured data will also increase since facts from different Wikipedia editions are merged into one knowledge base. At the current stage, there is no established API yet to access that knowledge, but in the near future this resource will help DS approaches a lot.

Bibliography

[Abney, 2007] Abney, S. (2007). *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC, 1st edition.

[Adafre and de Rijke, 2006] Adafre, S. F. and de Rijke, M. (2006). Finding Similar Sentences across Multiple Languages in Wikipedia. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 62–69.

[Agichtein and Gravano, 2000] Agichtein, E. and Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM Conference on Digital Libraries*, pages 85–94.

[Amitay et al., 2004] Amitay, E., Har'El, N., Sivan, R., and Soffer, A. (2004). Web-a-where: Geotagging web content. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 273–280.

[Arampatzis et al., 2006] Arampatzis, A., van Kreveld, M. J., Reinbacher, I., Jones, C. B., Vaid, S., Clough, P., Joho, H., and Sanderson, M. (2006). Web-based delineation of imprecise regions. *Computers, Environment and Urban Systems*, 30(4):436–459.

[Auer et al., 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., and Ives, Z. (2007). DBpedia: A nucleus for a web of open data. In *In 6th International Semantic Web Conference*, pages 11–15.

[Bader and Häussler, 2010] Bader, M. and Häussler, J. (2010). Word order in german: A corpus study. *Lingua*, 120(3):717–762.

[Besag, 1986] Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302.

[Biernacki and Waldorf, 1981] Biernacki, P. and Waldorf, D. (1981). Snowball sampling. problems and techniques of chain referral sampling. *Sociological Methods & Research*, pages 141–163.

[Blessing, 2006] Blessing, A. (2006). Annotationsrichtlinien für Straßen- und Platznamen. Technical report, IMS.

[Blessing et al., 2006] Blessing, A., Klatt, S., Nicklas, D., Volz, S., and Schütze, H. (2006). Language-derived information and context models. In *Proceedings of 3rd IEEE PerCom Workshop on Context Modeling and Reasoning*, pages 24–29.

[Blessing et al., 2007] Blessing, A., Kuntz, R., and Schütze, H. (2007). Towards a context model driven German geo-tagging system. In *Proceedings of the 4th ACM Workshop on Geographic Information Retrieval*, pages 25–30.

[Blessing and Schütze, 2008] Blessing, A. and Schütze, H. (2008). Automatic acquisition of vernacular places. In *Proceeding of the 10th International Conference on Information Integration and Web-based Applications Services*, pages 662–665. ACM.

[Blessing and Schütze, 2010a] Blessing, A. and Schütze, H. (2010a). Fine-grained geographical relation extraction from wikipedia. In *Proceeding of the 7th International Conference on Language Resources and Evaluation*, pages 2949–2952.

[Blessing and Schütze, 2010b] Blessing, A. and Schütze, H. (2010b). Self-annotation for fine-grained geospatial relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 80–88.

[Blessing and Schütze, 2012] Blessing, A. and Schütze, H. (2012). Crosslingual distant supervision for extracting relations of different complexity. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1123–1132.

[Bohnet, 2010] Bohnet, B. (2010). Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97.

[Bollacker et al., 2008] Bollacker, K. D., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

[Brakensiek and Rigoll, 2004] Brakensiek, A. and Rigoll, G. (2004). Handwritten address recognition using hidden markov models. *Reading and Learning*, pages 103–122.

[Brunner and Purves, 2008] Brunner, T. J. and Purves, R. S. (2008). Spatial autocorrelation and toponym ambiguity. In *Proceedings of the 2nd International Workshop on Geographic Information Retrieval*, pages 25–26.

[Carletta, 1996] Carletta, J. (1996). Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

[Carlson et al., 2010] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E. R. H., and Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Proceedings of the 24th Conference on Artificial Intelligence*, pages 1306–1313.

[Cimiano et al., 2004] Cimiano, P., Handschuh, S., and Staab, S. (2004). Towards the self-annotating web. In *Proceedings of the 13th International Conference on World Wide Web*, pages 462–471.

[Clough, 2005] Clough, P. (2005). Extracting metadata for spatially-aware information retrieval on the internet. In *Proceedings of the 2005 Workshop on Geographic Information Retrieval*, pages 25–30.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

[Craven et al., 1999] Craven, M., Kumlien, J., et al. (1999). Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*, pages 77–86.

[Culotta et al., 2006] Culotta, A., McCallum, A., and Betz, J. (2006). Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 296–303.

[de Melo and Weikum, 2010] de Melo, G. and Weikum, G. (2010). Menta: inducing multilingual taxonomies from wikipedia. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1099–1108.

[Doddington et al., 2004] Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). The automatic content extraction (ace) program: Tasks, data and evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 837–840.

[Dürr et al., 2004] Dürr, F., Hönle, N., Nicklas, D., Becker, C., and Rothermel, K. (2004). Nexus–a platform for context-aware applications. In *1. Fachgespräch Ortsbezogene Anwendungen und Dienste der GI-Fachgruppe KuVS*, pages 15–18.

[Elkan and Noto, 2008] Elkan, C. and Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 213–220.

[Etzioni et al., 2004] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2004). Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th International Conference on World Wide Web*, pages 100–110.

[Etzioni et al., 2011] Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam (2011). Open information extraction: The second generation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 3–10.

[Eugenio and Glass, 2004] Eugenio, B. D. and Glass, M. (2004). The kappa statistic: a second look. *Computational Linguistics*, 30(1):95–101.

[Faruqui and Padó, 2010] Faruqui, M. and Padó, S. (2010). Training and evaluating a german named entity recognizer with semantic generalization. In *Proceedings of the Conference on Natural Language Processing (KONVENS)*, pages 129–133.

[Ferrucci et al., 2010] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J., Nyberg, E., Prager, J., et al. (2010). Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79.

[Ferrucci and Lally, 2004] Ferrucci, D. and Lally, A. (2004). Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.

[Florian et al., 2003] Florian, R., Ittycheriah, A., Jing, H., and Zhang, T. (2003). Named entity recognition through classifier combination. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 168–171.

[Giunchiglia et al., 2008] Giunchiglia, F., Yatskevich, M., and Shvaiko, P. (2008). Semantic matching. Technical report, University of Trento, Department of Information and Communication Technology.

[Grishman and Sundheim, 1996] Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, pages 466–471.

[Hermjakob et al., 2008] Hermjakob, U., Knight, K., and Daumé III, H. (2008). Name translation in statistical machine translation: Learning when to transliterate. *Proceedings of the 46th Annual Meeting on Association for Computational Linguistics: Human Language Technologies*, pages 389–397.

[Hoffmann et al., 2010] Hoffmann, R., Zhang, C., and Weld, D. S. (2010). Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 286–295.

[Jensen et al., 2004] Jensen, D., Neville, J., and Gallagher, B. (2004). Why collective inference improves relational classification. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 593–598.

[Ji et al., 2010] Ji, H., Grishman, R., Dang, H. T., Griffitt, K., and Ellis, J. (2010). An overview of the tac2010 knowledge base population track. In *Proceedings of the 3rd Text Analytics Conference*. electronic.

[Joho and Sanderson, 2004] Joho, H. and Sanderson, M. (2004). The spirit collection: an overview of a large web collection. *SIGIR Forum*, 38(2):57–61.

[Kaljahi, 2010] Kaljahi, R. S. Z. (2010). Adapting self-training for semantic role labeling. In *Proceedings of the ACL 2010 Student Research Workshop*, pages 91–96.

[Kang et al., 2007] Kang, H., Sehgal, V., and Getoor, L. (2007). Geoddupe: A novel interface for interactive entity resolution in geospatial data. In *Proceeding of the 11th International Conference on Information Visualization*, pages 489–496.

[Kim et al., 2010] Kim, S., Jeong, M., Lee, J., and Lee, G. G. (2010). A cross-lingual annotation projection approach for relation detection. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 564–571.

[Leidner et al., 2003] Leidner, J. L., Sinclair, G., and Webber, B. (2003). Grounding spatial named entities for information extraction and question answering. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References*, pages 31–38.

[Liao and Veeramachaneni, 2009] Liao, W. and Veeramachaneni, S. (2009). A simple semi-supervised algorithm for named entity recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 58–65.

[Lu and Getoor, 2003] Lu, Q. and Getoor, L. (2003). Link-based text classification. In *IJCAI Workshop on Text Mining and Link Analysis*, pages 496–503.

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

[McCallum, 2002] McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

[Medelyan et al., 2009] Medelyan, O., Milne, D., Legg, C., and Witten, I. H. (2009). Mining meaning from wikipedia. *International Journal of Human-Computer Studies*, 67:716–754.

[Mintz et al., 2009] Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. 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 1003–1011.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine learning*. McGraw Hill series in computer science. McGraw-Hill.

[Müller et al., 2008] Müller, C., Zesch, T., Müller, M.-C., Bernhard, D., Ignatova, K., Gurevych, I., and Mühlhäuser, M. (2008). Flexible uima components for information retrieval research. In *Proceedings of the LREC 2008 Workshop Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP*, pages 24–27.

[Nadeau and Sekine, 2007] Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

[National Imagery and Mapping Agency, 2000] National Imagery and Mapping Agency (2000). Department of defense world geodetic system 1984: its definition and relationships with local geodetic systems. Technical Report TR8350.2, National Imagery and Mapping Agency.

[Noreen, 1989] Noreen, E. W. (1989). *Computer-intensive methods for testing hypotheses: an introduction*. Wiley.

[Nothman et al., 2013] Nothman, J., Ringland, N., Radford, W., Murphy, T., and Curran, J. R. (2013). Learning multilingual named entity recognition from wikipedia. *Artifical Intelligence*, 194:151–175.

[Ogren et al., 2008] Ogren, P. V., Wetzler, P. G., and Bethard, S. (2008). Cleartk: A uima toolkit for statistical natural language processing. In *UIMA for NLP workshop at Language Resources and Evaluation Conference*. electronic.

[Overell and Rüger, 2008] Overell, S. and Rüger, S. (2008). Using co-occurrence models for placename disambiguation. *International Journal of Geographical Information Science*, 22(3):265–287.

[Overell, 2009] Overell, S. C. (2009). *Geographic Information Retrieval: Classification, Disambiguation and Modelling*. PhD thesis, Imperial College London.

[Overell, 2011] Overell, S. C. (2011). The problem of place name ambiguity. *SIGSPATIAL Special*, 3(2):12–15.

[Padó, 2006] Padó, S. (2006). *User's guide to `sigf`: Significance testing by approximate randomisation*.

[Parton et al., 2009] Parton, K., McKeown, K., Coyne, B., Diab, M. T., Grishman, R., Hakkani-Tür, D., Harper, M. P., Ji, H., Ma, W. Y., Meyers, A., Stolbach, S., Sun, A., Tür, G., Xu, W., and Yaman, S. (2009). Who, what, when, where, why? Comparing multiple approaches to the cross-lingual 5W task. In *Proceedings of ACL 2009*, pages 423–431.

[Pasley et al., 2007] Pasley, R. C., Clough, P. D., and Sanderson, M. (2007). Geotagging for imprecise regions of different sizes. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 77–82.

[Pateman and Johnson, 2010] Pateman, B. M. and Johnson, C. (2010). Using the wikipedia link structure to correct the wikipedia link structure. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 10–18.

[Peters, 2001] Peters, C., editor (2001). *Cross-Language Information Retrieval and Evaluation*, volume 2069. Springer.

[Quinlan, 1986] Quinlan, J. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

[Riedel et al., 2010] Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163.

[Russell and Norvig, 1995] Russell, S. J. and Norvig, P. (1995). *Artificial intelligence - a modern approach: the intelligent agent book*. Prentice Hall series in artificial intelligence. Pearson Education.

[Sanftmann et al., 2009] Sanftmann, H., Blessing, A., Schütze, H., and Weiskopf, D. (2009). Visual Exploration of Classifiers for Hybrid Textual and Geospatial Matching. In *Proceedings of Vision, Modeling, and Visualization VMV*, pages 245–253.

[Sarawagi, 2008] Sarawagi, S. (2008). Information extraction. *Foundations and trends in databases*, 1(3):261–377.

[Schiehlen, 2003] Schiehlen, M. (2003). A cascaded finite-state parser for german. In *Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics*, pages 163–166.

[Schmid, 1995] Schmid, H. (1995). Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

[Schmid, 2000] Schmid, H. (2000). Unsupervised learning of period disambiguation for tokenisation. Technical report, IMS, University of Stuttgart.

[Schmid and Laws, 2008] Schmid, H. and Laws, F. (2008). Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 777–784.

[Sehgal et al., 2006] Sehgal, V., Getoor, L., and Viechnicki, P. D. (2006). Entity resolution in geospatial data integration. In *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*, pages 83–90.

[Silberer et al., 2008] Silberer, C., Wentland, W., Knopp, J., and Hartung, M. (2008). Building a Multilingual Lexical Resource for Named Entity Disambiguation, Translation and Transliteration. In *Proceedings of the 6th International Language Resources and Evaluation*. electronic.

[Smith and Crane, 2001] Smith, D. A. and Crane, G. (2001). Disambiguating geographic names in a historical digital library. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, pages 127–136.

[Snow et al., 2004] Snow, R., Jurafsky, D., and Ng, A. Y. (2004). Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems*. This is a draft version from the NIPS preproceedings; the final version will be published by April 2005.

[Srihari, 1993] Srihari, S. (1993). Recognition of handwritten and machine-printed text for postal address interpretation. *Pattern recognition letters*, 14(4):291–302.

[Suchanek et al., 2007] Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International World Wide Web Conference*, pages 697–706.

[Suchanek et al., 2009] Suchanek, F. M., Sozio, M., and Weikum, G. (2009). SOFIE: A Self-Organizing Framework for Information Extraction. In *Proceedings of the 18th International Conference on World Wide Web*, pages 631–640.

[Wang et al., 2007] Wang, G., Yu, Y., and Zhu, H. (2007). Pore: positive-only relation extraction from wikipedia text. In *Proceedings of the 6th International The Semantic Web Conference*, pages 580–594.

[Wu et al., 2008] Wu, F., Hoffmann, R., and Weld, D. S. (2008). Information extraction from wikipedia: moving down the long tail. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 731–739.

[Wu and Weld, 2007] Wu, F. and Weld, D. S. (2007). Autonomously semantifying wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 41–50.

[Wu and Weld, 2010] Wu, F. and Weld, D. S. (2010). Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127.

[Xu et al., 2007] Xu, F., Uszkoreit, H., and Li, H. (2007). A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 584–591.

[Yao et al., 2010] Yao, L., Riedel, S., and McCallum, A. (2010). Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023.

[Zesch et al., 2008] Zesch, T., Müller, C., and Gurevych, I. (2008). Extracting lexical semantic knowledge from wikipedia and wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation*. electronic.

[Zhou et al., 2005] Zhou, G., Su, J., Zhang, J., and Zhang, M. (2005). Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 427–434.