Automatic Term Extraction for Conventional and Extended Term Definitions across Domains

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

Vorgelegt von

Anna Constanze Hätty

aus Ulm

Hauptberichter	apl. Prof. Dr. Sabine Schulte im Walde
Mitberichter	Prof. Dr. Jonas Kuhn
Mitberichter	Prof. Dr. Ulrich Heid

Tag der mündlichen Prüfung: 20. März 2020

Institut für Maschinelle Sprachverarbeitung
der Universität Stuttgart

2020

Erklärung (Statement of Authorship)

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst habe und dabei keine andere als die angegebene Literatur verwendet habe. Alle Zitate und sinngemäßen Entlehnungen sind als solche unter genauer Angabe der Quelle gekennzeichnet.

Contents

List of Abbreviations

ANN	artificial neural networks
ATE	automatic term extraction
CBOW	continuous bag-of-words
CNN	convolutional neural networks
DIY	‘do it yourself’
FFNN	feed-forward neural network
GRU	gated recurrent unit
GTT	General Theory of Terminology
LR	logistic regression
LSC	lexical semantic change
LSTM	long short-term memory
MLP	multilayer perceptron
MWE	multi-word expression
MWT	multi-word term
NLP	natural language processing
NN	neural networks
PMI	pointwise mutual information
POS	part-of-speech
RNN	recurrent neural networks
SCS	Simple Compound Splitter
SGNS	skip-gram with negative sampling
SWT	single-word term

Abstract

A terminology is the entirety of concepts which constitute the vocabulary of a domain or subject field. Automatically identifying various linguistic forms of terms in domain-specific corpora is an important basis for further natural language processing tasks, such as ontology creation or, in general, domain knowledge acquisition. As a short overview for terms and domains, expressions like *hammer*, *jigsaw*, *cordless screwdriver* or *to drill* can be considered as terms in the domain of DIY ('do-it-yourself'); *beaten egg whites* or *electric blender* as terms in the domain of cooking. These examples cover different linguistic forms: simple terms like *hammer* and complex terms like *beaten egg whites*, which consist of several simple words.

However, although these words might seem to be obvious examples of terms, in many cases the decision to distinguish a term from a 'non-term' is not straightforward. There is no common, established way to define terms, but there are multiple terminology theories and diverse approaches to conduct human annotation studies. In addition, terms can be perceived to be more or less terminological, and the hard distinction between term and 'non-term' can be unsatisfying.

Beyond term definition, when it comes to the automatic extraction of terms, there are further challenges, considering that complex terms as well as simple terms need to be automatically identified by an extraction system. The extraction of complex terms can profit from exploiting information about their constituents because complex terms might be infrequent as a whole. Simple terms might be more frequent, but they are especially prone to ambiguity. If a system considers an assumed term occurrence in text, which actually carries a different meaning, this can lead to wrong term extraction results. Thus, term complexity and ambiguity are major challenges for automatic term extraction.

The present work describes novel theoretical and computational models for the considered aspects. It can be grouped into three broad categories: term definition studies, conventional automatic term extraction models, and extended automatic term extraction models that are based on fine-grained term frameworks. Term complexity and ambiguity are special foci here.

In this thesis, we report on insights and improvements on these theoretical and computational models for terminology: We find that terms are concepts that can intuitively be un-

derstood by lay people. We test more fine-grained term characterization frameworks that go beyond the conventional term/'non-term'-distinction. We are the first to describe and model term ambiguity as gradual meaning variation between general and domain-specific language, and use the resulting representations to prevent errors typically made by term extraction systems resulting from ambiguity. We develop computational models that exploit the influence of term constituents on the prediction of complex terms. We especially tackle German closed compound terms, which are a frequent complex term type in German. Finally, we find that we can use similar strategies for modeling term complexity and ambiguity computationally for conventional and extended term extraction.

Zusammenfassung

Eine Terminologie umfasst die Gesamtheit der Konzepte, die das Vokabular einer Domäne oder eines Themenfeldes bilden. Verschiedene linguistische Formen von Termen in domänenspezifischen Korpora automatisch zu identifizieren, ist eine wichtige Grundlage für darauf aufbauende Anwendungen der Verarbeitung natürlicher Sprache, wie zum Beispiel dem Ontologieaufbau oder, ganz generell, der Akquirierung von Domänenwissen. Ausdrücke wie *Hammer*, *Laubsäge*, *Akkubohrschrauber* oder *sägen* können beispielsweise als Terme der Domäne Heimwerken gelten; *Eischnee* oder *elektrischer Standmixer* können Terme für die Domäne Kochen darstellen. Die gegebenen Beispiele beinhalten verschiedene linguistische Formen: einerseits einfache Terme, wie *Hammer*, und andererseits komplexe Terme, wie *Akkubohrschrauber*, welche aus mehreren einfachen Wörtern bestehen.

Auch wenn diese Beispiele ziemlich eindeutig zu sein scheinen, ist die Abgrenzung eines Terms von einem 'Nicht-Term' nicht immer einfach. Es gibt keinen einheitlichen, etablierten Weg, um Terme zu definieren, dafür aber verschiedene Terminologietheorien und Ansätze zur manuellen Annotation. Außerdem können Terme als mehr oder weniger terminologisch empfunden werden, und eine klare Trennung von Termen und 'Nicht-Termen' kann unbefriedigend sein.

Zu den Herausforderungen der Termdefinition kommen schließlich bei der automatischen Termextraktion noch weitere Herausforderungen hinzu, denn sowohl einfache als auch komplexe Terme müssen automatisch von einem Extraktionssystem identifiziert werden können. Die Extraktion von komplexen Termen kann davon profitieren, dass Konstituenten betrachtet werden, denn komplexe Terme als Ganzes sind häufig sehr selten. Einfache Terme dagegen sind häufiger, neigen dafür aber auch stärker zu Mehrdeutigkeiten; wortgleiche Vorkommen dieser Terme mit anderen Bedeutungen in ein Extraktionsverfahren miteinzubeziehen kann zu falschen Extraktionsresultaten führen. Daher sind Termkomplexität und Mehrdeutigkeit bedeutende Herausforderungen für die Termextraktion.

Die vorliegende Arbeit beschreibt neue theoretische und komputationelle Modellierungsansätze für die angeführten Aspekte. Sie kann in drei grobe Kategorien eingeordnet werden: Studien zur Termdefinition, konventionelle automatische Termextraktionsmodelle und erweiterte

automatische Termextraktionsmodelle mit zugrundeliegender feinerer Termdefinition. Termkomplexität und Mehrdeutigkeit sind dabei spezielle Fokusthemen.

In dieser Dissertation berichten wir von Verbesserungen und Erkenntnissen aus den theoretischen und komputationellen Termmodellen: Wir kommen zu dem Ergebnis, dass Terme Konzepte sind, die Laien intuitiv verstehen können. Wir testen feiner differenzierte Termdefinitionsmodelle, die über die konventionelle Term/'Nicht-Term'-Unterscheidung hinausgehen. Als neuen Ansatz beschreiben und modellieren wir die Mehrdeutigkeit von Termen als graduelle Bedeutungsvariation zwischen Allgemeinsprache und domänenspezifischer Sprache und nutzen die resultierenden Repräsentationen, um Fehler zu vermeiden, die Termextraktionssysteme typischerweise aufgrund von Mehrdeutigkeit machen. Wir entwickeln komputationelle Modelle, die den Einfluss von Konstituenten auf die Vorhersage von komplexen Termen ausnutzen. Dabei fokussieren wir uns im Speziellen auf deutsche geschlossene Komposita, eine typische Art von komplexen Termen im Deutschen. Schließlich zeigt sich, dass wir ähnliche Strategien nutzen können, um Termkomplexität und Mehrdeutigkeit für die konventionelle und die erweiterte Termextraktion komputationell zu modellieren.

Acknowledgments

The present thesis was developed in the time of my doctoral contract with the Robert Bosch GmbH, Corporate Research, in Renningen, when I was a doctoral candidate at the Institute for Natural Language Processing (IMS) at the University of Stuttgart.

I want to thank Sabine Schulte im Walde, my main IMS advisor, for her good advice and detailed knowledge of my projects. Sabine always was in constant exchange with me, and gave me both directions as well as the freedom to explore my own ideas.

I am also grateful to Michael Dorna, my Bosch advisor, who never hesitated to have a chat with me about the current challenges of my work, and who shared the insights of past and current NLP projects with me. Having Sabine and Michael by my side, I was consistently guided on my way through my PhD.

I further want to thank Jonas Kuhn and Ulrich Heid, for taking part in the doctoral committee, and giving me tips and new impulses for my work. Jonas and Sabine gave me fast and easy support for collecting manually labeled data. I enjoyed the discussions with Ulrich Heid, who gave me new perspectives for my research.

I was also deeply supported by Michael Dambier, Michael Hanselmann (as part of the 'three Michaels', as they introduced to me) and Dietrich Manstetten. The Silicon Valley experience was made possible for me, which allowed me to pursue a six month research cooperation project at the Bosch Research and Technology Center North America. Supporting me during this time in Sunnyvale, I also want to thank Bingqing Wang and Zhe Feng. Further, during my time as a PhD student, I was given the opportunity to participate in numerous conferences (the ACLs in Berlin and Florence, the NAACL in New Orleans, the EACL in Valencia, among others), and to hire students to support me in my work.

The list of people who influenced my work and to whom I want to express my gratitude does not end here, of course: Dominik Schlechtweg started as a PhD student with Sabine nearly simultaneously with me, and the valuable collaboration project about meaning variation inspired me for follow-up work. I further want to thank several other people for their support: the other PhDs Marco, Maximilian, Kim-Anh, Daniel, Patricia, Toni and the Bosch doctorate network; the students, Daniela, Eric, Julia, Anurag and the many annotators, whose work was

the basis for mine; the other colleagues who joined me during lunchtime or conferences, and all the other unnamed people.

Finally, I want to thank my family and friends. My heartfelt gratitude goes to my parents, my grandmother and my brother, who all supported me in so many ways. I want to especially thank my mom and Alex, who were on the spot at any time when I needed them.

1. Introduction

1.1. Motivation

Whenever one deals with topically restricted domains or subject fields, domain-specific terminology is omnipresent. A term represents a relevant concept of a domain, and in its entirety, a terminology builds the skeletal structure of the knowledge of a domain. The relation to a domain is the main characteristic that distinguishes domain-specific terms from general-language words because terminology is mostly part of natural language. And here the problem begins: it is not always straightforward to recognize terms, neither for humans nor for computational models. Some expressions might be more explicit terms than others. For example, the German expression *Antiblockiersystem* 'anti-lock braking system' would probably be highly associated with the automotive domain. However, should expressions like *Kraft* 'power', which belong to general language but still have some relevance for the automotive domain, be included in a set of terms for that domain? The examples show that there are gradual differences in the strength of association of a term to a domain. This finding leads to the question of what characterizes a term. Is it an intuitive concept or does it need to be defined by terminology or domain experts? Can we find a framework to describe terms in a more fine-grained way, to represent different strengths of a term's association to a domain?

Once we have a concrete idea of what we will consider to be terms, the next step is to design computational models for automatic term extraction. Automatically recognizing terminology is an important area in the field of natural language processing, and is particularly crucial for several follow-up tasks, such as ontology creation, translation of technical texts or compiling technical dictionaries. Of course, automatic techniques for recognizing terminology also struggle with the issues described above and cause new challenges as well. Particularly, term complexity and ambiguity are challenging. By term complexity we mean that terms come in different linguistic forms: complex terms, where a word or phrase consists of several simple words, like *verlorene Eier* 'poached egg' (lit.: 'lost eggs') for cooking, and simple terms, that only consist of one simple word, like *Säge* 'saw' for DIY ('do-it-yourself'). For *verlorene Eier*, the complex term comes in the form of separate strings, interrupted by spaces (further

on called multi-words). Furthermore, especially in German, complex terms occur frequently, if not predominantly, in a unique form: closed compounds. Compounds are lexemes formed by adjoining two or more lexemes (Bauer, 2003), and they are written as one word, such as *Stichsäge* 'jigsaw'. The extraction of different linguistic forms of terms needs to be realized differently. In contrast to simple terms, constituents play a role for the identification of complex terms. Therefore we investigate the interplay of complex terms and constituents. Since the primary language of our studies is German, a particular focus is put on the relationship of closed German compound terms and their constituents. The aim is to get a better understanding and new insights into this relationship and exploit this newly gained knowledge to improve the extraction of complex terms. The underlying idea is that rare complex terms should be more reliably predicted using information about their constituents; this is a necessary step since complex phrases tend to be infrequent, and moreover, complex terms are more limited to occurring in domain texts.

The second problem for term extraction is ambiguity, i.e., that a word or phrase carries different meanings. For example, *Schnee* means 'snow' in general language but also carries the special meaning of 'beaten egg whites' in the cooking domain. Although this problem predominantly concerns simple terms, which usually have more general meanings and are thus more ambiguous, other forms can be ambiguous as well; e.g. the German closed compound *Fuchsschwanz* means 'ripsaw' in DIY, but can be literally understood as the tail of a fox in general language. Ambiguity is a problem for automatic term extraction systems, since they might extract an assumed term occurrence in text which actually carries a different, not domain-related meaning. Extraction results can be wrong then. The problem of term ambiguity has been addressed in the form of word sense disambiguation tasks in specialized domains. However, we compile term lists and thus follow a type-based extraction approach, and a token-based word sense disambiguation would be an unnecessarily complicated method. We thus see the need to represent ambiguity in a new way: ambiguity should be modeled as the error a term extraction system would make, in order to be able to correct the error then. As might have become clear from the examples, the major problem is that a domain-specific (terminological) meaning of a phrase is mixed with a non-domain-relevant general-language one. For those reasons, we represent the ambiguity of a term as its degree of meaning variation between general and domain-specific language. Computing the degree of meaning variation gives us one single value of estimated extraction error, and it can be set off against other values computed in the term extraction process.

Up to this point, we focused on challenges for automatic term extraction but remained with a conventional definition of a term. In other words, a set of terms is extracted but the terms are

not further differentiated. Besides, since we motivated the need for a more fine-grained term definition framework in the beginning, we see the need to test the predictability of granularly distinguished terms in a computational modeling task. We call this an extended term extraction model. We tackle the problems of term complexity and term ambiguity again. Both problems are highly relevant and challenging for conventional and extended term extraction models, and tackling the challenges for both kinds of term definition allows us to find parallels in the modeling possibilities.

1.2. Research questions

As motivated, this thesis concentrates on three problems: i) term definition, ii) term complexity and iii) term ambiguity. These goals are not independent of each other. Investigating term definition with human annotation studies results in the realization that a more fine-grained term definition is needed. That is why term complexity and term ambiguity are not only modeled for conventional binary automatic term extraction (distinguishing terms from non-terms), but also for automatic term extraction systems that are based on a more fine-grained term definition.

The first part of the thesis deals with theoretical considerations about the definition of terminology. This is necessary because past terminology theories are diverse, as well as gold standard annotation procedures. We want to answer the questions: **What defines a term? Is it an intuitive concept? What are the characteristics of a term?** To get more thorough insights, different perspectives of lay people and (semi-)experts are investigated in empirical annotation studies. The studies validate the need to **find a new, more fine-grained term characterization framework** instead of the mere distinction into term and non-term. Such a term characterization framework could then be used for various follow-up applications for term extraction, because the fine-grained characterization allows us to extract a subset of terms for a follow-up task (e.g. for the creation of a glossary, only terms that need to be explained could be selected). We investigate two term characteristics that we deem to be responsible for a more fine-grained term distinction: centrality (topical association of an expression to a domain) and specificity (level of domain-specific information an expression carries, or the level of specialization). Theoretically, these characteristics could even form a two-dimensional description of a term. However, in practical human annotation studies, we find that annotators have difficulties with a fine-grained distinction of centrality, while they can agree more on specificity.

The next two parts of the thesis deal with computational modeling approaches for automatic term extraction. At first, we focus on term complexity and term ambiguity for **conventional binary term extraction**. Regarding term complexity, this means we investigate the **relationship between complex and simple terms or constituents**. The following questions shall be answered:

- How does additional information about constituents of a multi-word term candidate influence the identification of a multi-word expression as a term?
- What are the important features to identify a simple vs. a complex term?

Regarding term ambiguity, we have the following research questions:

- How should meaning variation for terms be annotated and modeled, concerning the several senses the words convey in general and in domain-specific language?
- What methodologies can be used for implementation?
- How can the meaning variation be exploited to improve term extraction?

Both research strands, term complexity and ambiguity, are then tackled again in a second automatic extraction task, this time involving an **extended, more fine-grained term definition**. The human annotation studies showed that there is a need for a more fine-grained term distinction. Since human annotation studies showed that specificity is a clearer indicator for such a distinction, the extended term definition frameworks tested here all principally relate to **term specificity**. There are two basic research questions here:

- Concerning term complexity and ambiguity: how can the challenges of term complexity and ambiguity be tackled for this extended task? Can we transfer ideas and modeling solutions from earlier experiments? Are there some common insights for term complexity and ambiguity on the two different automatic term extraction tasks?
- Concerning specificity: can specificity be reliably modeled? What characterizes specificity?

1.3. Contributions

Term definition studies. For a thorough understanding of the difficulties of manually detecting terminology in text, user studies are designed and conducted. For a more in-depth

analysis, lay understanding of what constitutes a term, and (semi-)expert understanding is investigated. We find that the intuitive understanding of a term is gradual instead of binary.

Investigating term characteristics. From the insights of prior user studies and previous literature, we find two characteristics that can establish a gradual understanding of what constitutes a term: centrality (topical association to a domain) and specificity (level of domain-specific information; specialization). We conduct a human annotation study to test the applicability of these characteristics and find a better common human intuition for specificity than for centrality.

Simple and complex term extraction. We computationally model the extraction of simple and complex terms. We present a new extraction model, where the complex term features are applied to sub-parts or constituents as well. Further, the relationship between complex terms and constituents is analyzed, and the role of constituents with different morphological functions of a complex term is investigated. We find that constituents in front positions have a stronger influence on term prediction than constituents in end positions (which relates to the distinction between modifiers and heads).

Term ambiguity. We present the first annotation and computational modeling approach for meaning variation between general and domain-specific language. Term ambiguity is modeled as degree of meaning variation, as a representation of a term extraction error. Contributions include the annotation procedure design of a gold standard for meaning variation (based on an annotation framework for diachronic meaning shifts), computational modeling of meaning variation, and the integration of the latter into term extraction systems.

Closed compounds in extended term models. We computationally model the prediction of German closed compounds for extended term frameworks (which are based on specificity). The main goal is to include constituent information into the compound prediction process. As one way to achieve that goal, we present the first model that optimizes the training process with an auxiliary prediction process for constituents. In a second study, we further analyze the influence of term, compound and constituent features on specificity prediction of German closed compounds.

Meaning variation in extended term models. Computational representations of meaning variation are included in a supervised model for the prediction of simple and complex

terms, based on an extended term framework. We test two variants: i) enriching the input information of the supervised model with a meaning variation representation, which is based on the previously developed method, and ii) designing a supervised model that computes the meaning variation dynamically. Both models outperform a state-of-the-art term extraction system.

1.4. Publications and Student Work

Parts of the chapters in this thesis are based on published work. All papers are collaboration work. In general, Sabine Schulte im Walde was the principal supervisor of the work, and Michael Dorna supervised as a representative of Bosch. Jonas Kuhn and Ulrich Heid took supervising positions as well. Ulrich Heid supervised in his role as project leader of the project 'terminology extraction and ontology construction'. The work by Julia Bettinger was developed in the context of her master thesis, which was supervised by me and Sabine Schulte im Walde. In the following, published work is listed that is incorporated into this thesis (or parts of it).

1. Anna Hätty, Dominik Schlechtweg, Michael Dorna and Sabine Schulte im Walde (2020). Predicting Degrees of Technicality in Automatic Terminology Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (ACL 2020), Seattle, USA (online conference due to COVID-19).

   → I implemented and evaluated the models, Dominik Schlechtweg contributed the idea and implementation of centering (10%), which I integrated into the models. I wrote the paper and the other authors assisted me in revising it.

2. Julia Bettinger, Anna Hätty, Michael Dorna, Sabine Schulte im Walde (2020). A Domain-Specific Dataset of Difficulty Ratings for German Noun Compounds in the Domains DIY, Cooking and Automotive. In *Proceedings of the 12th International Conference on Language Resources and Evaluation* (LREC 2020).

   → The publication describes a part of Julia Bettinger's master thesis. Roughly 10% (estimated) from my dissertation's chapter describing the master thesis has been reused in the paper.

3. Anna Hätty, Ulrich Heid, Anna Moskvina, Julia Bettinger, Michael Dorna and Sabine Schulte im Walde (2019). Akku-Bohr-Hammer vs. Akku-Bohrhammer: Experiments

towards the Evaluation of Compound Splitting Tools for General Language and Specific Domains. In *Proceedings of the 15th Conference on Natural Language Processing* (KONVENS 2019), pages 59-67, Erlangen, Germany.

→ The first part of the study was conducted under the guidance of Ulrich Heid und Anna Moskvina, with participation of me and Michael Dorna. The second part of the study was the joint work of Julia Bettinger and me, under supervision of Sabine Schulte im Walde; only this second part of the paper is part of this thesis (which is roughly one third of the paper): Julia Bettinger generated the compound selection and computed the compounds splits. I evaluated the data and wrote the second part of the paper (and made adaptations to the rest of the paper). Ulrich Heid and Sabine Schulte im Walde assisted me in revising the paper.

4. Anna Hätty, Dominik Schlechtweg and Sabine Schulte im Walde (2019). SURel: A Gold Standard for Incorporating Meaning Shifts into Term Extraction. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics* (*SEM 2019), pages 1-8, Minneapolis, Minnesota, USA.

   → In this work, I created the meaning shift dataset SURel, described its relevance for terminology (extraction), integrated it into an automatic term extraction task and evaluated it. Dominik Schlechtweg contributed the annotation concept and annotation guideline format from his work DURel for the dataset SURel (which is further used in the ACL 2019 study). He assisted me in setting up the annotation, and supervised the annotation process undertaken by German-speaking annotators (since I was abroad at that time), and created some of the dataset plots for the paper. I wrote the paper, all authors assisted me in revising the paper.

5. Dominik Schlechtweg, Anna Hätty, Marco Del Tredici, Sabine Schulte im Walde (2019). A Wind of Change: Detecting and Evaluating Lexical Semantic Change across Times and Domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (ACL 2019), pages 732–746, Florence, Italy.

   → The publication describes results of a cooperation project that goes beyond research for term extraction.[1] It describes results on two parallel studies on meaning variation, i) diachronic lexical meaning change (relevant to the first author's disser-

[1] In more detail, the cooperation project on semantic meaning shifts and meaning variation arose from an overlap in our research interests: diachronic lexical semantic change (Dominik Schlechtweg), lexical meaning variation between general language and specific domains (me), and meaning variation that occurs between online communities (Marco del Tredici).

tation) and ii) lexical meaning variation between general and domain-specific language (relevant to my dissertation). The tasks were divided as follows: Dominik Schlechtweg, Marco del Tredici and I conducted a literature review for all our research areas to find models for detecting semantic change and semantic variation, which we implemented jointly. I implemented one-third of the models. Dominik Schlechtweg and I evaluated the results of the models together. I participated in the writing of the paper.

6. Anna Hätty and Sabine Schulte im Walde (2018). Fine-Grained Termhood Prediction for German Compound Terms Using Neural Networks. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions* (LAW-MWE-CxG-2018), pages 62-73, Santa Fe, New Mexico, USA.

7. Anna Hätty and Sabine Schulte im Walde (2018). A Laypeople Study on Terminology Identification across Domains and Task Definitions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (NAACL 2018), pages 321-326, New Orleans, Louisiana, USA.

8. Anna Hätty, Simon Tannert and Ulrich Heid. Creating a gold standard corpus for terminological annotation from online forum data. In *Proceedings of language, ontology, terminology and knowledge structures workshop* (LOTKS 2017), Montpellier, France.

   → Ulrich Heid and I jointly designed the annotation guidelines in iterative steps, and conducted the annotation and the evaluation of the results. For that purpose, Simon Tannert set up WebAnno, supervised the annotation with it (e.g., if the refinement of the guidelines required changes in the WebAnno setup), and computed the annotation agreements (i.e., he did 95% of the implementation work). I further participated in the annotation process and computed the statistics of the jargon words (5% of the implementation). I wrote the paper and Ulrich Heid assisted me.

9. Anna Hätty, Michael Dorna, Sabine Schulte im Walde (2017). Evaluating the Reliability and Interaction of Recursively Used Feature Classes for Terminology Extraction. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics* (SRW@EACL 2017), pages 113-121, Valencia, Spain.

   → Michael Dorna implemented roughly 50% of the metrics, which is roughly 25% of the total implementation work, and I implemented the rest. I did the evaluation and

Michael Dorna assisted me. The writing of the paper was a joint effort of Michael Dorna and me, and Sabine Schulte im Walde assisted us.

Finally, student work has been conducted under my supervision. The work by Julia Bettinger and Anurag Nigam has been co-supervised by Sabine Schulte im Walde.

10. Daniela Gierscheck (2017). Disambiguierung von Termen in einem domänenübergreifenden Korpus von Sachtexten. *Master thesis*.

11. Julia Bettinger (2019). Predicting Term Difficulty of closed German Noun Compounds. *Master thesis*.

12. Anurag Nigam (2020). PageRank for detecting domain-specific terms. *Project work*.

1.5. Outline of Thesis

The thesis can be divided into three main parts: background, human annotation models and computational models.

The **background** part comprises chapters 2 and 3. **Chapter 2** describes the related theoretical work, i.a. terminology theories, sublanguages and sub-technical terms. In addition, **Chapter 3** describes related work for the application, i.e. automatic term extraction. Related automatic term extraction systems are described, as well as underlying datasets and standard evaluation measures. In addition, more general methodologies for computational modeling are described, such as word embedding architectures and the basics of neural networks.

Chapter 4 describes the **human annotation studies** that were conducted for a better understanding of what constitutes a term. The studies comprise the lay people experiment (Hätty and Schulte im Walde, 2018b) and the semi-expert annotation (Hätty et al., 2017b) in the first section. Based on the insight from that, the second section then describes the newly developed term model, based on centrality and specificity. The new model is again evaluated with a user study to test its robustness. Since annotators have difficulties agreeing on the concept of centrality, the computational modeling in chapter 6 focuses more on specificity.

The **computational modeling** part comprises chapters 5 and 6. Chapter 5 addresses problems for binary term extraction, chapter 6 addresses the same problems and partially incorporates the previously developed solutions into models for predicting extended term definition frameworks, focusing on term specificity. **Chapter 5** deals in the first section with the recognition of multi-word terms and simple terms, which is described in Hätty et al. (2017a). The second section then deals with annotating and modeling meaning variation and incorporating

them into term extraction. This is described in Hätty et al. (2019b) and Schlechtweg et al. (2019).

Chapter 6 describes work for German closed compounds and constituents for predicting an extended term characterization framework in the first section. This is described in Hätty and Schulte im Walde (2018a) and Bettinger et al. (2020), and the underlying student thesis (Bettinger, 2019). The auxiliary study on compound splitting in specific domains comprises the second study described in Hätty et al. (2019a). The second section deals with incorporating meaning variation into a supervised system for an extended term characterization prediction, which is also described in Hätty et al. (2020).

Chapter 7 finally summarizes the main insights and results, and describes ideas for further research.

2. Theoretical Background on Terminology

The chapter describes background research for the central concepts and ideas of this thesis. The first section describes and defines the two main concepts, 'terms' and 'domains'. The last two sections describe phenomena related to terminology, which are special challenges for term definition and automatic term extraction. Regarding the meaning of a term, we describe tiers of terminology and the problem of ambiguity. Regarding the linguistic forms of terms, we describe the variants in the last section.

2.1. Terminology and Domains

2.1.1. Defining Terms

Terms, in this sense denoting the items of a terminology, are integral parts of specialized text. Especially over the last century, the study of terminology has received a constant interest, even more with text digitization and the rise of natural language processing methodologies for automatic term extraction. Nevertheless, the status of terms is unclear, various definitions and theories have been presented in the past.

The International Organization for Standardization (ISO) defines terminology in the following way (ISO 1087-1, 2000):

> **3.4.3**
> **term**
> verbal **designation** (3.4.1) of a **general concept** (3.2.3) in a specific **subject field** (3.1.2)
>
> NOTE A term may contain symbols and can have variants, e.g. different forms of spelling.

The given definition shows that a fundamental attribute of a term is its association to a subject field. This definition can be seen in context of the traditional view on terminology, which distinguishes between the concept (the 'general concept' stands in contrast to an individual concept or an instance) and its sign (the 'designation', for example a linguistic expression).

The foundations of this traditional view were laid by Eugen Wüster in the 1930s, who promoted terminology as a new discipline, separated from lexicology.[1] Wüster defines concepts as mental constructs which emerge by perceiving real world objects and phenomena. The term is then the label we assign to those concepts, and is thus isolated from them. Wüster does not examine terms with respect to context, morphology and syntax. Ideally, there is a one-to-one correspondence between term and concept per domain, there is no synonymy and ambiguity. Concepts are clearly separated from each other and can be arranged in a concept system. Wüster thus marked the beginning of a school of terminology, which aims at separating terms from words. Representatives of this view elaborate on this distinction. For example, Sager (1990) describes terms as "items which are characterised by special reference within a discipline" in contrast to words, which are items that "function in general reference". In this line of thought, "terminology" stands in contrast to "vocabulary" (Sager, 1990, p. 19).

Later theories criticize this strict division between terminology and linguistics (Cabré, 1999; Cabré Castellví, 2003; Temmerman, 2000; Faber Benítez et al., 2005). The approaches are very different in nature, e.g. basing their ideas on prototype theory (Temmerman, 2000) or frame semantics (Faber Benítez et al., 2005). Nevertheless, they all see the need to examine terms in context and also accept term variation and ambiguity. Since we aim at extracting term lists from text, our understanding of terminology necessarily involves context and we also acknowledge ambiguity, as we go into detail further on. Since we want to identify term types and not term tokens for the practical application part, we characterize terms across all contexts. In a sense, we have an opposition of words and terms: we start from the domain under consideration, and we have a 'term' if an expression has a domain-relevant meaning in the respective domain. In reverse conclusion, we have a general-language 'word' if it has no special relevance for the domain. Anyway, words and terms cannot be seen as completely separate per se, since general language and domain language overlap. Furthermore, in later practical application parts, we deal with other non-term phenomena as well, such as extracting linguistic phrases as term candidates which contain evaluative parts and need to be ruled out [2]. Therefore we take a less restrictive definition as starting point, which originates from the area of automatic term extraction (Kageura and Umino, 1996, p. 259):

[1] This paragraph follows Pearson (1998), Faber Benítez (2009) and Fuertes-Olivera and Tarp (2014). For further reading see Wüster (1974) and Wüster (1979).

[2] This is why we prefer to use'non-term' as the general counterpart of 'term'.

> [...] terms [are] linguistic units which characterise specialised domains [...]

This definition implicitly expresses the two practical challenges we will need to cope with for automatic term extraction: we need to identify expressions in texts and have to make a decision if these are terms or not. We thus take this definition as basic working definition for now.

2.1.2. Defining Domains

As described in the previous section, terminologists intensively addressed the problem of what constitutes a terminology. Since the definitions of 'term' made already clear that terms have to be seen in tight connection to a subject field or domain, this leads to the question what constitutes a domain. We find short notes about that in articles addressing terminology research:

> In the context of terminological research, *subject field* should be construed as broadly as possible. In some quarters, there is a tendency to restrict terminology to scientific and technical fields (understanding *technical* as the field of applied science). In light of needs of translators, however, it is more common to regard any specialized area of human endeavor or study as a subject field for the purposes of terminological research. (Cole, 1987, p. 78)

> Terminology is only concerned with terms or words of a specialized field (such as physics, chemistry, anthropology, art an so on) or a professional domain (trading, industry, sports, etc.). (Cabré, 1996, p. 22)

> A field of knowledge is a discipline to the extent that it is institutionally and socially acknowledged through a university degree qualification or a branch of research or kind of activities carried out in a research centre. A field of knowledge is a semantically much wider term: it is an intellectual endeavour concerned with an object of study or research. (Cabré Castellví, 2003, p.195)

These descriptions give some intuitions about what can be considered as a domain in general, but shed little light on where are the thematic boundaries of a domain. However, in practical application tasks such as automatic term extraction, we will need to decide which texts can be considered as domain texts, and which ones should be excluded. In this respect, the ISO 1087-1 (2000) definition of 'subject field' is interesting:

3.1.2
subject field
domain
field of special knowledge

NOTE The borderlines of a subject field are defined from a purpose-related point of view. (ISO 1087-1, 2000)

The definition is essentially kept minimalistic, and sees the topical restriction of a domain on the application-side. We also keep a pragmatic approach for that in the thesis, and define the contents of a domain implicitly by all texts we can find with performing thematically focused crawling, or by already available specialized corpora. Since we involve different registers of texts (from user-written instructions to handbooks) and carry out experiments for several domains, we both aim at minimizing random effects and at finding various term phenomena which are not restricted to certain domains or registers. Nevertheless, this is a simplification we do in order to keep at least one of the notions 'term' and 'domain' fixed, while inspecting the other one. Since the definition of a term is connected to the one of a domain, we believe that they actually mutually condition each other: a domain is established by the terms which are used in it, and a term is established by its domain association.

2.2. The Heterogeneous Nature of Terms

The previous section showed that an important attribute of terms is their association to a domain. In the following, we give an overview of research strands related to terminology, which emphasize the heterogeneous nature of terms, and which show why the degree of association to a domain can differ from term to term.

2.2.1. Sublanguage

A notion which is connected with terms and domains is *sublanguage* (depending on the field of study, sublanguage is also called specialized language, special language, Language for Specific Purposes or scientific language). Sublanguages describe subsets of a language, restricted to certain subject fields (or domain) and exhibiting special linguistic characteristics. Terms are part of a sublanguage, but the sublanguage comprises more, for example syntax. A sublanguage is the language of a domain. The relation between a sublanguage and a domain is explained in more detail by Hirschman and Sager (1982, p. 27):

> We define sublanguage here as the particular language used in a body of texts, dealing with a circumscribed subject area (often reports and articles on a technical speciality or science subfield), in which the authors of the documents share a common vocabulary and common habits of word usage. As a result, the documents display recurrent patterns of word co-occurrence that characterize discourse in this area and justify the term sublanguage.

The concept of sublanguage is especially of interest to us because the status of sublanguages to each other and to general language can be defined. These interrelations are depicted in figure 2.1. Cabré (1999) states that sublanguages are subsets of a language, as well as general language is a subset of it. General language intersects with sublanguages, with which "it not only shares features but also maintains constant exchange of units and conventions" (Cabré, 1999, p. 65f). For example, there may be a conceptualization of a general word in a special language (e.g. with the emergence of the computer science area, the word *mouse* became a term in that domain), or a deconceptualization of a specific term to a general-language word. Sublanguages also intersect with each other, and share some common terminology. These intersections of sublanguages and general language play an important role for defining different tiers of terminology, as will be explained next, and as a theoretical basis for the corpus based techniques we rely on for the automatic extraction part.

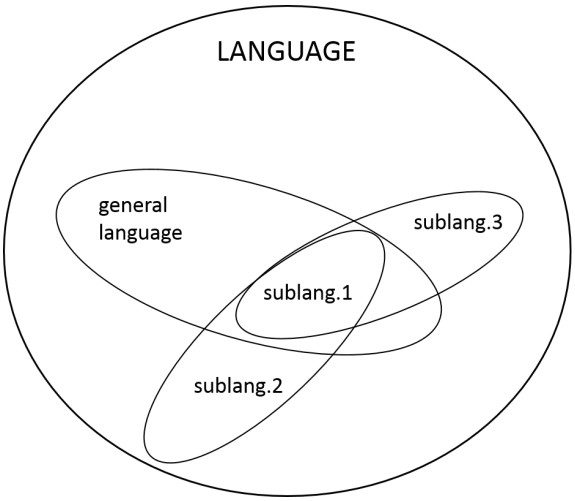


Figure 2.1.: The relation between sublanguages and general language within a language (q.v. Cabré, 1999).

To sum up, sublanguages cannot be considered as independent from general language or other sublanguages. This has the consequence that terminology also cannot be seen as completely independent from general vocabulary. Figure 2.2 illustrates the relation of terminology

with general vocabulary. An important aspect which is depicted here is that the intersection with general vocabulary evokes ambiguity for terminological expressions. These overlaps with other domains and general language, and the resulting ambiguities triggered theories early in terminological research history that separate terminology into different tiers.

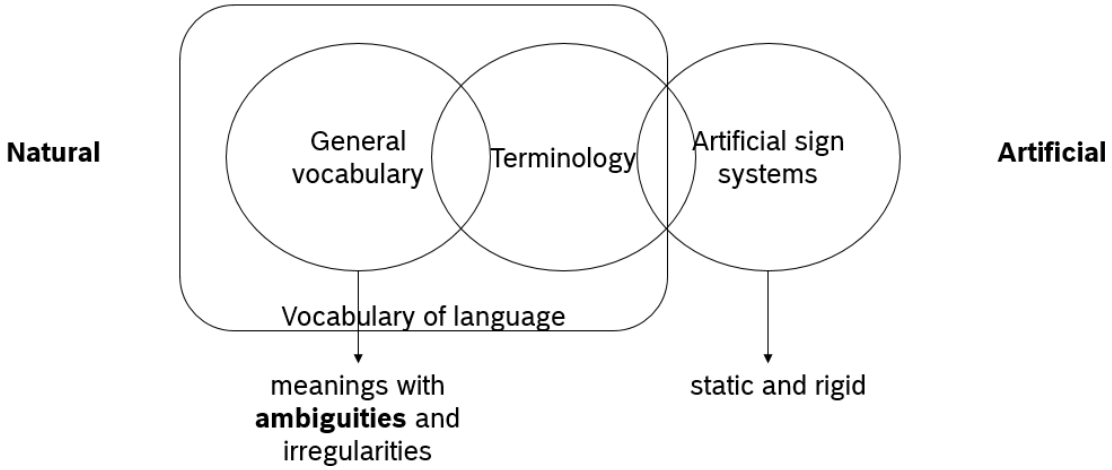


Figure 2.2.: The relation between general vocabulary, terminology and artificial sign systems (q.v. Kageura, 2012).

2.2.2. Tiers of Terminology

Some of the models, which distinguish terminology into different tiers, focus on the problem of shared vocabulary across domains. To cope with that problem, Hoffmann (1985) suggests three categories of terms: The *subject-specific terms*, the *non subject-specific terms* and the *general-language words*. General language words are not considered as terms, and non subject-specific terms are shared between more than one domain. A more fine-grained model is given by Roelcke (1999), who distinguishes between four tiers (Figure 2.3): *intra-subject terminology* is specific to a certain domain, *inter-subject terminology* is used in one domain, but also in others. *Extra-subject terminology* is terminology which does not belong to a domain but is used within it and *non-subject terminology* consists of all items used across almost all specific domains. Tutin (2007) calls the latter 'transdisciplinary vocabulary': it includes the domain-unspecific language of scientific writing (e.g. *evaluation, estimation, observation*) and non-specialized abstract vocabulary (e.g. *to present a problem, to result in*).

In conclusion, these models decompose terminology into groups of different strength of association to a certain domain.

In order to address the overlap of terminology with general language, Trimble and Trimble

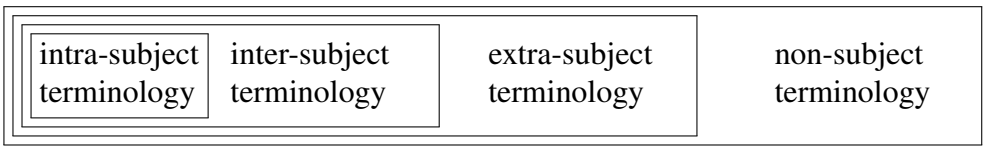


Figure 2.3.: Tiers of terminology (Roelcke (1999)) [our translation].

(1978) propose a slightly different tier model. They distinguish between three categories of terms: *highly technical terms*, a *bank of technical terms* and *sub-technical terms*. Highly technical terms are specific to the domain, the bank of technical terms are the basis for all domains and sub-technical terms are general-language words with special meanings in a domain. In that way, they define a separate class for terms which have a separate general-language meaning. Pearson (1998) criticizes that some terms in the category 'sub-technical' could still belong to the highly technical terms or the bank of technical terms with regard to their domain sense(s), but their additional usage in general language enforces them to be categorized as sub-technical. In fact, elements of the category 'sub-technical term' can be distinguished in more detail, as it will be elaborated next.

To sum up, tier models exist that build both on shared vocabulary across domains and with general language. While both kinds of models have their validity (also with respect to sub-languages, as described before), we mainly focus on the domain and general language as contrast[3]. There are several reasons for that, a) our perspective on terminology only considers all relevant terms for a domain under consideration, and not for others, b) most studies are performed from the lay perspective, and knowledge about other domains cannot be presupposed and c) for automatic term extraction we (hypothetically) would need to consider all possible other domain corpora instead of one general-language corpus, which is not feasible.[4]

2.2.3. Sub-technical Terms and Ambiguity

Trimble and Trimble (1978) and their description of sub-technical terms are an early reference for apparently general-language words with special meanings in a domain. We will use 'sub-technical term' further on to describe the latter, which should be noted because experts do not actually agree on the notion of sub-technicality; Cowan (1974) first introduced sub-technical

[3] as one exception, we use the Roelcke model as a general idea for term heterogeneity and a gradual term understanding

[4] Term extraction approaches exist that consider other domain corpora, but a selection should be carefully chosen with regard to the task.

terms as the terms being shared across disciplines. Due to the varying definitions given by experts, Baker (1988, p. 92) lists the various cases concerning words that can fall into the category of sub-technical terms, and gives examples:

1. general-language vs. domain-specific meaning:
   a) words with a specialized meaning in one discipline and a different general-language meaning (e.g. *bug* in computer science and general language)
   b) words with a specialized meaning in several disciplines and a different general-language meaning (e.g. *solution* in mathematics and chemistry, and general language)
2. words which have different meanings in several domains, but which do not exist in general language (e.g. *morphological* in linguistics and botany)
3. words that are general to several or all domains (e.g. *factor*, *method* or *function*)
4. general-language words with a restricted meaning in a specialized domain (e.g. in botany, *effective* means 'take effect' and has no evaluative meaning)
5. general-language words which are used in preference to semantically equivalent words (e.g. in biology, photosynthesis effects do not *happen*, they *take place*)
6. words which perform specific rhetorical functions in specialized texts (e.g. 'One *explanation is*...', '*Others* have *said* ...' or 'It has been *pointed out* by ...').[5]

Overall, the cases differ in at least two aspects: a) number of senses and b) second occurrence in general language and/or other domain. Concerning a), for half of the cases (3,5,6) words might be shared across domains or with general language, but the meaning stays the same. For the other half (1,2,4) we find a variation in meaning. This demonstrates how important ambiguity is when making considerations about what belongs to a terminology. Further, there is another important aspect concerning ambiguity. Consider the meaning divergences from general language to a specific domain, i.e. case 1a. and 4. One can see that gradual differences in the strength of ambiguity exist, which might also have an influence on the inclusion of a word in a terminology. All in all, as it was already brought up by Pearson (1998), it might not be satisfactory to just have a bag of sub-technical terms as Trimble and Trimble

[5]Note that this category can be considered as general technical language (Tutin, 2007), and expressions are not associated to concrete domains. Given the previous term definitions, we would not consider these expressions as terms.

(1978) proposed, but to investigate further the influence of other meanings and usages across domains and general language onto the capability of a word to be a term.

2.2.4. Implications

The tier models in this section showed that the definition of terminology is heterogeneous. Pearson (1998) (who calls these kind of models *pragmatic definitions of terms*) finds two kinds of distinctions, which are commonly made by these approaches to separate terms into tiers: a) distinguishing according to **familiarity** of a term, and b) distinguishing according **subject-specificity** of a term. Although she further states that these attributes are hard to measure, one can infer that the attributes are the two important intuitions people have for characterizing terms. We conclude that a more elaborated term representation is necessary which addresses the heterogeneous nature of terms, and that basing this representation on these two attributes would result in an intuitive characterization of terms. The 'extended term definition', given in the title of this thesis and described and evaluated in chapter 4.4, is based on these two attributes.

A third aspect is **ambiguity**. Terms exhibiting ambiguity are generally put into the category of sub-technical terms. However, this might not need a separate category, but influences the classification into other categories (again, as pointed out by Pearson, 1998). For example, a word with a highly technical term sense and a general-language sense might still be perceived to be more technical than a word with a less technical term sense and a general-language sense as well. A monosemous highly technical term might still be considered to be the most technical. Since we aim at extracting all terms that are relevant for a given domain in this work, we follow this line of thought; we evaluate expressions for their capability to be a term, even if other general-language meanings exist. The degree of meaning variation between domain and general language is important here, as given in Baker's examples. Detecting the degree of ambiguity is especially relevant for automatic term extraction systems, which cannot easily distinguish between meanings.

2.3. Linguistic Forms of Terms

The most important linguistic forms of terms can be broadly divided into simple and complex terms. **Simple terms** (ISO 1087-1, 2000, 3.4.4) contain only one root, for example *sound* and *light*. **Complex terms** (ISO 1087-1, 2000, 3.4.5) contain two or more roots, for example terms separated by white space (*fault recognition circuit*), hyphenated terms (*know-how*) or closed

compound terms (*bookmaker*). There exist other forms of terms, such as abbreviations and acronyms.

Closed compound terms are a type of complex terms which are especially frequent in German, which is why we especially focus on them for complex term annotation and extraction. **Closed compounds** are complex expressions that consist of several lexemes and that are written in a single string of characters. The lexemes are called **constituents**. The constituents of a compound can be divided into **modifier** and morphological **head**, which usually is word-final in German.

For automatic term extraction and term annotation within context, closed compound terms and terms separated by white space trigger two opposing challenges: For terms separated by white space, the phrase needs to be identified and all words that belong to the phrase (e.g. that the term is *fault recognition circuit* and not *recognition circuit*). For closed compound terms, the phrase boundaries are obvious, but the compound needs to be split in order to identify the constituents (e.g. *book* and *maker* in *bookmaker*). In order to distinguish between these two kinds of challenges, we either talk about closed compounds, or about other kinds of multi-word expressions[6]. In this work, the cases where we use the notion 'multi-word expression' or 'multi-word term' are restricted to those complex forms which contain white spaces, in order to emphasize the implications that the form has on annotation and extraction within text context.

2.4. Summary

The chapter dealt with theoretical aspects of terminology, and relevant definitions for the following work. The basic definitions of terminology and domains are kept rather vague at first, but we further elaborate on related work on term attributes which makes clear that terminology is heterogeneous, and individual terms differ in their strength of association to a domain. We further described different linguistic forms of terms which are relevant for the later annotation and automatic term extraction procedures.

[6]Multi-word expressions (MWEs) are expressions formed of two or more words, which actually include closed compounds. In contrast to closed compounds, other forms of MWEs can be separated by white space in German or English, but do not necessarily have to.

3. Methodologies, Data and Evaluation for Automatic Term Extraction

The chapter describes foundations and background work of automatic term extraction, as well as other methodologies which were used for the practical parts of the thesis which deal with automatic term extraction.

3.1. Automatic Term Extraction

Automatic term extraction (ATE) describes the automatic extraction or recognition of terminology in text, or text corpora, and is related to fields like information retrieval or automatic keyword extraction (Kageura and Umino, 1996). While automatic term extraction is concerned with extracting linguistic expressions from domain texts which are relevant for the respective domain, automatic keyword extraction deals with identifying the most relevant expressions for a document. In the following, we describe basic concepts underlying ATE (3.1.1), important term extraction measures (3.1.2) and using term extraction measures as input for machine learning approaches (3.1.3).

3.1.1. Unithood and Termhood

Kageura and Umino (1996) base automatic term recognition on two methodological concepts: *unithood* and *termhood*. Unithood measures the strength of association of the constituents of a complex phrase. Kageura and Umino describe it as not only to be relevant for terms, but for other complex expressions as well, such as grammatical collocations or idiomatic expressions. Termhood measures the extent to which such a phrase then would be a term. Thus, termhood is relevant to both simple and complex terms. Association measures like *chi squared* or *pointwise mutual information* (PMI) are common measures for unithood (for an overview, see Evert

2005). For example, pointwise mutual information is computed as follows:

$$pmi(x, y) = \frac{p(x, y)}{p(x)p(y)} \tag{3.1}$$

Here, x and y are events (in our case words) for which the ratio of their joint probability and the probability that they occur together by chance is computed. For example, PMI scores should be higher for a valid phrase like 'dependency parsing' than for two words that do not actually belong together, like 'green ideas'.

Termhood measures are described in the following sections. Note that evaluating a term for its unithood and termhood does not necessarily have to be a two-step-process, both attributes can be combined in one measure. To that effect, Zadeh and Handschuh (2014) describe a typical ATE pipeline as shown in figure 3.1. First, term candidates are extracted from text with a linguistic filter, for example a POS pattern. A still frequently used linguistic filter for identifying term or collocation candidates (Manning et al., 1999; Bonin et al., 2010a, i.a.) was introduced by Justeson and Katz (1995). They define a regular expression which summarizes POS patterns for multi-word terms[1]. After candidate term extraction, the candidate is evaluated for unithood and termhood, which finally leads to a ranked term list.

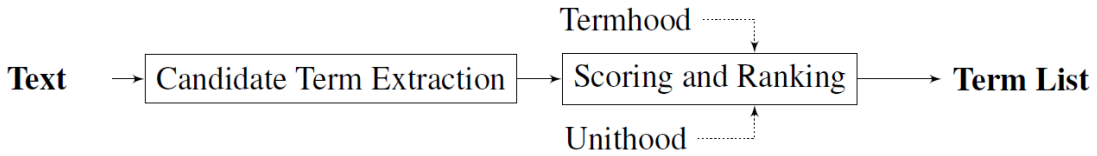


Figure 3.1.: Term extraction procedure (Zadeh and Handschuh, 2014).

Zadeh and Handschuh (2014) point out that unithood characterizes syntagmatic relations (relations holding between elements that can be combined with each other), while termhood characterizes paradigmatic relations (relations between elements that can be substituted for each other).

3.1.2. Term Extraction Measures

Concerning the distinction between unithood and termhood, the following term extraction measures fall into the category of termhood measures[2].

[1] ((A | N)+ | ((A | N)*(NP)?)(A | N)*)N (Justeson and Katz, 1995, p. 16), where A stands for adjective, N for noun and NP for noun phrase

[2] However, as we already pointed out, unithood and termhood measures do not need to be strictly separated, and term extraction measures might implicitly contain some measure of unithood as well.

A well-known term extraction method is the C/NC-value (Frantzi et al., 1998, 2000). The **C-value** is a measure sensitive to nested multi-word terms. Both the total occurrence of a candidate term and its occurrence in a longer candidate term are included in the measure. The C-value is defined as

$$c(a) = \begin{cases} log_2(|a|)f(a), & \text{a is not nested} \\ log_2(|a|)(f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b)), & \text{otherwise} \end{cases} \tag{3.2}$$

where a is the candidate string, $f(a)$ is the frequency of a, T_a is the set of extracted candidate terms that contain a and $P(T_a)$ is the number of these candidate terms. The first factor in the formula gives higher weight to longer phrases, while the second one reduces weight if the candidate string is frequently nested within longer phrases. In other words, the extraction of partial term phrases shall be impeded. However, longer phrases will be preferred in general, but that a term is often used within other terms does not necessarily mean that it should be given less weight. The C-value has some further drawbacks: In its original definition, it is not suitable to extract single word terms, since $log_2(|a|)$ would then be equal to 0 and make the whole formula 0. Work exists to adapt the C-value to consider single words terms (Barrón-Cedeno et al., 2009). Secondly, Frantzi et al. do not precisely define the set of extracted candidate terms T_a (Mykowiecka et al., 2018). As an extension of the C-value, **NC-value** is defined. The measure is a combination of the C-value and information from context words of the term candidate. It is defined as

$$nc(a) = 0.8\ c(a) + 0.2 \sum_{b \in C_a} f_a(b) * weight(b) \tag{3.3}$$

where C_a is the set of distinct context words of a, b is a word from C_a, $f_a(b)$ is the frequency of b as a term context word of a, and $weight(b)$ is the weight of b as a term context word. That means that the second summand (the 'nc-weight') increases if the frequent context words of a term are typical term context words. The hypothesis that terms occur in similar contexts is exploited in our work by using context-based word embeddings (c.f 3.3.3).

Inspired by the C-value, Nakagawa and Mori (2003) introduce the GM and **FGM** as termhood measures for noun compounds. However, no candidate term list is presupposed, which makes the measure more convenient to apply. The principle of the measures is to look at the left and the right context in a compound. GM (Geometric Mean) includes the left and right

context words of a noun compound

$$GM(cn,k) = \sqrt[2l]{\prod_{i=1}^{l}(LN(n_i,k)+1)*(RN(n_i,k)+1)} \tag{3.4}$$

where cn is the compound noun with length $l : n_1, \ldots, n_l$ (n_i is a simple noun). LN and RN are defined as follows:

$$LN(N,k) = \sum_{i=1}^{f_{left}(N)} f(L_i)^k \tag{3.5}$$

$$RN(N,k) = \sum_{j=1}^{f_{right}(N)} f(L_j)^k \tag{3.6}$$

where $f_{left}(N)$ and $f_{right}(N)$ are the number of distinct simple nouns which directly precede or succeed N, $f(L_i)$ and $f(L_j)$ denote the frequency of these simple nouns, and k is a parameter that can be individually set, with small k downgrading the frequency of a noun bigram, and for higher k (k>1) upgrading it.

Finally, the FGM score is computed as:

$$FGM(cn,k) = GM(cn,k) * f(cn) \tag{3.7}$$

with $f(cn)$ denoting the frequency of the compound. That means that the final score of a compound term is dependent on a combination of compound and constituent frequency. Note that the measure uses a constituent word's frequency as constituent, not its total corpus frequency. Since we do experiments on German closed compounds, where an important goal is to combine compound and constituent information for scoring a compound for its termhood, we also use the FGM-score as a model feature there.

One of the most important groups of measures for term extraction are **contrastive measures** (Ahmad et al., 1994; Rayson and Garside, 2000; Drouin, 2003; Kit and Liu, 2008; Bonin et al., 2010a; Kochetkova, 2015; Lopes et al., 2016; Mykowiecka et al., 2018, i.a.), which we use in several studies of this thesis. Contrastive measures compare term candidates in a domain-specific corpus and one or more reference corpora, mostly founded on frequency-based calculations. Reference corpora can represent other domains or general language. The basic idea behind contrastive measures is to score term-related discrepancies between domain-specific and reference corpora. For example, if an expression occurs more often in a domain-specific corpus than in a general-language reference corpus, it is more likely that it is a term than if it

would occur equally frequently in both corpora. In addition, contrasting general-language and domain-specific corpora operationalizes the idea that domain-specific terminology stands (to some degree) in contrast to general-language vocabulary (see 2). This is an important hypothesis underlying this thesis, which is why we repeatedly use contrastive measures. We give an overview of contrastive measures used in this thesis in the following. We describe them as a comparison of a domain-specific corpus and a general-language reference corpus, as they are used in this work. In the strict sense for some measures several reference corpora can be used.

Weirdness ratio (Ahmad et al., 1994) computes the ratio of the relative frequency in a domain-specific and a general-language corpus.

$$\text{WEIRD}(x) = \frac{f_{spec}(x)/s_{spec}}{f_{gen}(x)/s_{gen}}, \tag{3.8}$$

where f_{spec} and f_{gen} correspond to the frequencies of a term candidate x in a general-language and a domain-specific corpus, and s_{spec} and s_{gen} are the respective sizes of the corpora. Weirdness ratio is the simplest of the contrastive measures.

Contrastive Selection via Heads (CSvH) (Basili et al., 2001) incorporates the fact that multi-word terms are not always found in a general-language reference corpus, but that the heads are good indicators of a domain-specialization of a complex term.

$$\text{CSvH}(x) = log(f_{spec}(h(ct))) * log(\frac{s_{gen}}{f_{gen}(h(ct))}) * f_{spec}(ct) \tag{3.9}$$

In formula 3.9, ct stands for the complex term, and $h(ct)$ for its head. Only the domain frequency of the complex term is used for evaluation (last factor in product), otherwise the head is evaluated. The actual contrastive part, the factor in the middle, is inspired by tf-idf: it represents the inverse word frequency. Only head frequencies are used to compute this factor. This idea is advantageous because it is more likely that a simple word like the head occurs in the reference corpus. Nevertheless, it is also likely that the head meaning is similar to the meaning of the complex term.

Term Frequency Inverse Term Frequency (TFITF) (Bonin et al., 2010a) is a variant of CSvH. Unlike CSvH, the contrastive function for the head is directly applied to the complex term, which is why the last factor in the formula is neglected:

$$\text{TFITF}(x) = log(f_{spec}(x)) * log(\frac{s_{gen}}{f_{gen}(x)}) \tag{3.10}$$

As the measure's name makes already clear, it is still based on tf-idf. While tf-idf captures the importance of a term to a document, TFITF captures the importance of a term to a domain-

specific corpus. Instead of inverse document frequency, inverse term frequency in a general corpus is used as reference.

Although TFITF is designed for scoring multi-word expressions, it is still problematic if these expressions are rare. Thus, Bonin et al. (2010a) propose a second measure, that is suitable for handling variation for low frequencies, **Contrastive Selection of multi–word terms** (CSmw):

$$\text{CSMW}(x) = arctan(log(f_{spec}(x)) * \frac{f_{spec}(x)}{\frac{f_{gen}(x)}{s_{gen}}}) \tag{3.11}$$

The arc tangent has the beneficial attribute that it increases fast for small values, but soon gets close to an asymptote as values increase. We illustrate the effect of applying the arc tangent to the fraction in the above formula: In the first case, imagine all term candidates have the same general-language frequency. With higher domain frequencies, the CSmw value will also be higher. If we keep the domain frequencies fixed, however, with higher general-language frequencies the CSmw value will be lower. Finally, the positive effect of the low general-language frequencies is moderated by introducing the factor $log(f_{spec}(x))$.

To sum up, we described the best-known measures and the most relevant ones for this thesis. These measures only account for a small proportion the vast amount of existing term extraction measures. There are measures addressing all possible attributes of terms (e.g. addressing term variation), domains (e.g. using topic models), and also attributes used in closely related tasks, such as keyword extraction.

3.1.3. Machine Learning for Term Extraction

Due to the vast amount of unsupervised measures and metrics for term extraction, it stands to reason that many supervised term extraction systems use a combination of these measures as features. Classical machine learning algorithms such as random forests or conditional random fields are used with unithood and termhood measures as input features. For example, linguistic, statistical and hybrid term extraction features are used, as well as related features such as for collocation or keyword extraction. Of course, the set of features can be unlimitedly extended by other features, such as characterizing words in general (e.g. n-grams) or more uncommon features that characterize terms (e.g. more 'creative' features as using web searches or historical domain-specific data) or using external resources as thesauri. Examples for experiments using classical machine learning algorithms with termhood features are given in Zhang et al. (2010); Dobrov and Loukachevitch (2011); Nokel et al. (2012); da Silva Conrado

et al. (2013); Fedorenko et al. (2014); Yuan et al. (2017). To illustrate feature combinations we amplify some of the examples: Nokel et al. (2012) experiment with several machine learning methods for term extraction. They test typical term extraction measures (for example contrastive measures) but also rather uncommon features like novelty or orthographic features. Da Silva Conrado et al. (2013) group their features into the mentioned categories 'linguistic', 'statistical' and 'hybrid': as linguistic features they use POS-tags or occurrence frequencies of certain POS categories, as statistical features they use tf-idf or term frequency, and as hybrid features they use C-value and NC-value. Yuan et al. (2017) again compare several machine learning algorithms and use unithood (e.g. chi-squared) and termhood features (e.g. Weirdness ratio), but also a measure for the related task of glossary extraction.

Since a recent trend is using deep learning technologies and trained word embeddings, this has arrived in the field of automatic term extraction as well. An advantage of these methods is that there is no need for handcrafted features anymore. For unigram term extraction, Amjadian et al. (2016, 2018) combine word embeddings trained on general language and word embeddings trained on a specific domain as input for a Multilayer Perceptron. This method can be seen as related to the contrastive termhood measures, with exploiting information both from the domain and from general language. For multi-word term extraction, Kucza et al. (2018) treats the extraction as a sequence labeling problem, and applies two kinds of recurrent neural networks (LSTM and GRU) on the basis of word-level and character-level embeddings. Wang et al. (2016) perform a bootstapping approach to augment minimally labelled data, by training two deep learning classifiers (CNN vs LSTM) with word embedding input on the same data. The best term predictions are then incrementally added to the training data.

To sum up, next to unsupervised term extraction methods it is common practice to use rich term feature sets as input to machine learning algorithms such as decision trees or conditional random fields. This has the advantage that features can be optimally designed for the task, and the influence of individual features on the algorithm's performance can be evaluated. The more recent trend of using word embeddings and neural networks can boost performance, and there is flexibility for designing the network structure. However, these models are harder to interpret. We apply both kinds of practices. Note that word embeddings are general representations of word semantics, and not specific to term extraction (in contrast to term features). We will amplify word embeddings in the background second to last section, as well as other machine learning methodologies.

3.2. Data and Evaluation for Terminology Extraction

In this section we introduce existing gold standard terminology datasets along with the domain-specific corpora they are extracted from. Gold standard datasets represent the assumed correct solution for a task, and one way to create such data is to let test persons label the data. We compare the corpora and datasets to the domains and domain-specific text material we selected for the practical part of the thesis. For evaluation, we describe general measures that are frequently used for term extraction studies and in this thesis.

3.2.1. Domain-Specific Corpora and Terminology Gold Standards

In order to perform automatic term extraction we need text corpora as basis, and in order to evaluate the term extraction procedure we need a terminology gold standard dataset. While corpora and gold standards are needed in other NLP tasks as well, there are several specific problems here. The problem starts with copyright issues that often restrict the use of domain-specific data. As a result, either the annotation itself or the background corpora cannot be published (e.g. Bernier-Colborne and Drouin, 2014). There are further difficulties, which are mostly related to the fact that defining terminology and domains is already a challenging task. Such difficulties are:

- It has to be decided if term types or terms in context should be annotated. If term types are annotated, it has to be decided how to proceed with ambiguous entities. For example, what do we do with a term like *aspect* in the domain of linguistics, which occurs in text both as a term (in the sense of a grammatical category) and as a general- language word?
- Term variation has to be considered; inflectional variants or multi-word term candidates and their constituents, which possibly are stand-alone terms as well, need to be addressed appropriately.
- There are no universally accepted guidelines on when to make the decision for a term. The degree of a term candidate's termhood and objective of the annotation might result in the selection of different terms, i.e., only highly technical terms might be considered, or only a special kind of terms (e.g. only parts of cars for the automotive domain). This might depend on the nature of a domain and the occurring terms. For example, a domain can be broad and may have many subdomains, and the heterogeneous set of terms might need to be restricted.

For all these reasons, only a few public-access, elaborated and annotated terminological resources exist. A notable effort has been made in the biomedical domain. One of the best-known terminological resources is **GENIA** (Ohta et al., 2002; Kim et al., 2003), a collection of 2000 labeled abstracts of biomedical journals with almost 100,000 annotations by two domain experts. A more recent dataset is the **ACL RD-TEC** (Zadeh and Handschuh, 2014) based on publications in the domain of computational linguistics. The domain was chosen to facilitate evaluation of automatic term extraction systems, because it is likely that researchers are familiar with the computational linguistics domain. The terminology dataset is extracted from ACL ARC, an automatically segmented and POS-tagged corpus of 10,922 ACL publications from 1965 to 2006. More than 83,000 term candidates types were extracted from the corpus. ACL RD-TEC adds a manual annotation of 22,044 so-called *valid terms* and 61,758 *non-terms*. The term annotations are further refined with a labeling of 13,832 *terminology terms* which are defined as means to accomplish a practical task, such as methods, systems and algorithms used in computational linguistics. **ACL RD-TEC 2.0** (QasemiZadeh and Schumann, 2016) is an extension of ACL RD-TEC. It comprises a further terminology annotation for 300 computational linguistics publication abstracts. Terms are marked within their sentence contexts and a subclassification of the terms is introduced. GENIA and the ACL RD-TEC are frequently used as a basis for term extraction and evaluation tasks (e.g. recent usages of both datasets can be found in Zhang et al., 2016b; Yuan et al., 2017; Kucza et al., 2018).

Concerning these three corpora and associated gold standard datasets, we only use the ACL RD-TEC for our experiments. We address the problem of term extraction on a general, domain-independent level, focusing to a high extent on lexical semantics. The area of medicine would be special, for example it contains many Latin expressions, that would require special extraction mechanisms. We nevertheless address a variety of domains in this thesis: cooking, DIY, automotive, natural language processing, hunting and chess. In addition, we do not restrict to technical, expert-written texts to create the corpora. We use expert texts (papers, handbooks, expert-written instructions) and semi-expert and user-written texts (Wikipedia, user-written instructions on domain-specific homepages etc.), in order to explore various specifications of terms.

3.2.2. Evaluation Measures

Efforts to evaluate a term extractor's output can be broadly divided in two procedures (Bernier-Colborne, 2012; Astrakhantsev et al., 2015):

- A manual inspection of a term extractor's output, for example with the help of domain

		predicted class	
		p′	**n′**
gold value	**p**	True Positives (TP)	False Negatives (FP)
	n	False Positives (FP)	True Negatives (FP)

Table 3.1.: Confusion matrix.

experts

- A comparison of the output to a terminology gold standard or reference list

While the second procedure is standard for evaluation not only for term extraction, applying the first procedure is often due to the lack of gold standard terminologies and the challenge that the gold standard adequately represents different characteristics of a terminology (e.g. term variation).

For the second procedure, depending on the term extractor's output (e.g. a ranked term list, an unordered list or discrete classes), different evaluation measures can be applied. If a distinction into terms and non-terms is provided by the term extractor, **Precision, Recall** and **F1-score** can be computed. Using the information about correctly found (TP), misclassified (FP) and overseen (FN) elements in the classification shown in table 3.1, the measures can be computed in the following way:

$$Precision = \frac{TP}{TP + FP} \tag{3.12}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.13}$$

$$\textit{F1-score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.14}$$

However, term extraction systems often return a ranked list of terms, but make no distinction between a term or non-term class. In these cases, ranking measures are used for evaluation, as for example **Precision at k** ($P@k$) and **Average Precision** (AP). For $P@k$ only the top k elements of a ranked list are evaluated. It is thus defined as

$$P@k = \frac{1}{k} \sum_{i=1}^{k} x_i, \tag{3.15}$$

where x_i equals 1 if it is a term in the gold standard, and 0 otherwise. AP can then be computed to measure the overall performance on the basis of $P@k$ considering all possible k's for the length of the gold standard set T:

$$AP = \frac{1}{|T_{terms}|} \sum_{k=1}^{|T|} x_k P@k \tag{3.16}$$

T contains both terms and non-terms (or for more general applications, the positive and the negative class), and T_{terms} is the subset which only contains terms.

3.3. Machine Learning and Word Embeddings

In this section, we describe the general machine learning techniques we use in our experiments. We mainly use a) decision trees and b) artificial neural networks for term classification. For decision trees the focus of our experiments lies on feature design and on the interpretation of the tree structures. Concerning the artificial neural networks, we experiment with the structure of feed-forward neural networks, in order to optimally design them for our term extraction challenges. This is why we describe feed-forward neural networks in more detail than decision tree classifiers. In the last part, we describe the two kinds of word embeddings we use as input for the classification systems. Word embeddings are vector representations for word meaning, and we describe the machine learning techniques used to create them. In our experiments, we pre-train the word embeddings and use the static resulting representations as input for the classification systems.

3.3.1. Decision Tree Classifier

Decision tree classifiers are supervised machine learning methods that are represented as tree structures. For classification, an output class is learned from given input features, whereby data is recursively split by several decision rules learned from these features. The decision for

the output class is made in the leaves of the tree.

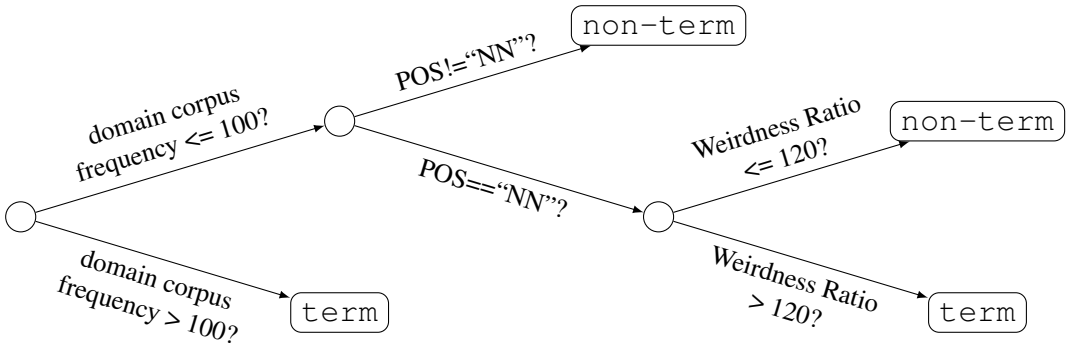


Figure 3.2.: Schema of a decision tree for term classification (with example decisions).

We give a schematic example for a decision tree that is trained from data for binary term classification in figure 3.2. In the example, it is distinguished between two classes, "term" and "non-term" (but more classes can be predicted, as we do in the experiments as well).

The split points in each step can be computed by metrics designed to measure the quality of a split. Commonly used metrics for decision tree classification are *Gini impurity* and *entropy* (c.f. Breiman, 1996), which are similar. Formulas are given in 3.17 and 3.18, as defined for each node. C denotes the number of classes.

$$G = \sum_{c \in C}^{C} p(c)(1 - p(c)) \tag{3.17}$$

$$H = -\sum_{c \in C}^{C} p(c) log(p(c)) \tag{3.18}$$

The quality of a split is characterized by the homogeneity of a node. If a node contains only elements of one class, it is homogeneous, and this characterizes a good split. Both Gini impurity and entropy measure homogeneity: If a node contains only elements of one class, we reach a perfect homogeneity with Gini impurity and entropy being zero.

In our work, we use decision tree classifiers because of their advantageous characteristics: This includes implicit feature reduction, that they are able to handle both numerical and categorical data, and that decisions are easy to understand and tree structures can be visualized. We concentrate on the feature design and do not intensively vary the parameters of the models. We use the above splitting criteria, and restrict tree size intuitively for that the classifiers do not overfit. We restrict to using the CART (Classification and Regression Trees) algorithm (Breiman et al., 1984), that only allows binary tree structures.

In the next section we describe artificial neural networks, which stand in contrast to decision tree algorithms due to their 'blackbox' nature.

3.3.2. Artificial Neural Networks

Artificial neural networks (commonly *neural networks*, NN) are computational models which are inspired by the biological neural networks in the human brain. Their usage is a recent trend in the field of machine learning, especially for computer vision and speech and language processing. In the following, a short introduction of artificial neural networks will be given, mostly based on the overviews by Haykin (1994), Goodfellow et al. (2016) and Goldberg (2017).

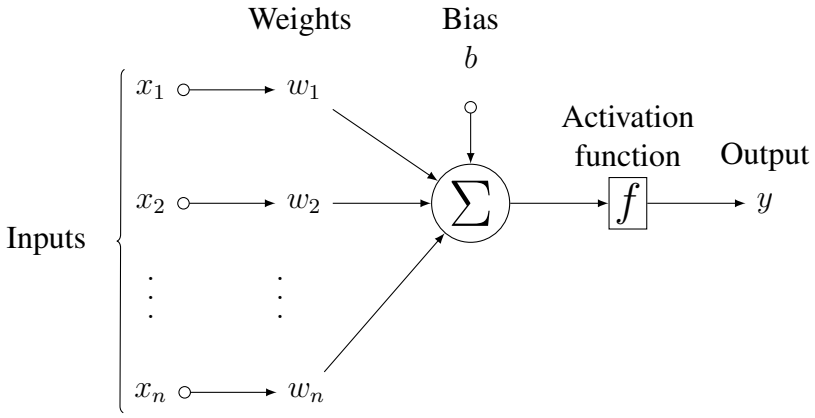


Figure 3.3.: Schema of a neuron (q.v. Haykin, 1994).

Artificial neurons. The basic unit of a neural network is a **neuron** or *node*. A neuron is a mathematical function that maps one (or more) input values to an output value (see figure 3.3). The input signals come in through connecting links (*synapses*) and are weighted and summed with a linear combination. The output is then determined by an *activation function f*. The purpose of the activation function is to regulate the permissible value range of the output. The activation is further influenced by a *bias value b* which increases or decreases the net input. In mathematical terms, the output signal y of a neuron can be computed as:

$$y = f(\sum_{j=1}^{n} w_j x_j + b) \tag{3.19}$$

There are several activation functions to use, depending on the application. For example,

these can be step functions, to ensure that the neuron only fires after exceeding a certain threshold, partially linear and non-linear functions such as *sigmoid*, *tanh*, *rectified linear unit (ReLU)* and *softmax*. The formulas are given in equations 3.20 - 3.23.

$$sigmoid(x) = \frac{1}{1 + exp(-x)} \tag{3.20}$$

$$tanh(x) = 2 \cdot sigmoid(2x) - 1 \tag{3.21}$$

$$relu(x) = max(0, x) \tag{3.22}$$

$$softmax(x) = \frac{e^x}{\sum_{i=1}^{k} e^{x_i}} \tag{3.23}$$

For *sigmoid* and *softmax* activations, there is the advantage that they are bound to be between 0 and 1, which makes them suitable for being the final classification output activation. The function *tanh* is similar to the *sigmoid* function, but is zero-centered, which allows for modeling inputs that have strong positive, negative or neutral values. *ReLU* is computationally efficient and allows for a quick convergence. All those functions have a derivate, which is beneficial for the backpropagation algorithm used to train a neural model.

Feed-forward artificial neural networks. Feed-forward neural networks (FFNN) have a directed acyclic graph structure with only forward connections. Neural networks with backward feedback connections are called **recurrent neural networks (RNN)**. **Convolutional neural networks (CNN)** are a special kind of FFNNs.

FFNNs are basically composed of three parts, an input layer, none or more hidden layers and an output layer. Each layer consists of at least one neuron. A network restricted to three layers, an input, hidden and output layer, is called a **multilayer perceptron (MLP)**. Such a network is depicted in figure 3.4. The **input layer** receives the real world data. In the area of natural language processing, this could be words, text or speech signals. For example, each word is mapped to a unique index. Since this is a rather weak representation, the input layer will often be succeeded by an *embedding layer* where for each index a word embedding will be retrieved (word embeddings will be further described in section 3.3.3). The lookup table underlying the embedding layer will have the size $|V| \times d$, with V denoting the vocabulary and d the dimensionality of the word embeddings. The input information is forwarded to a **hidden**

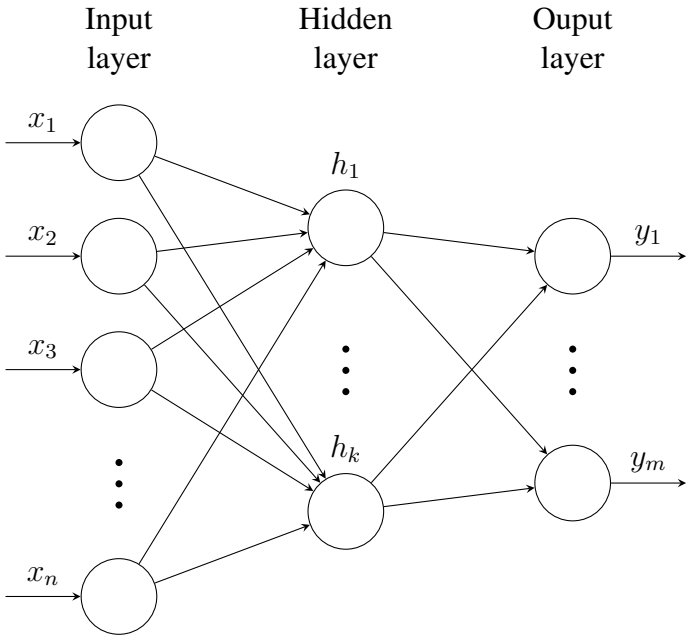


Figure 3.4.: Depiction of a fully-connected feed-forward neural network with a single hidden layer.

layer, where the non-linear transformation is computed, i.e. $h = f(Wi)$, where h is the hidden layer output, i is the input, W is the weight matrix and f is an activation function. The signal is forwarded to the **output layer**, where the final prediction is made. For classification, the number of output neurons represents the number of classes to be predicted. For a binary classification, the *sigmoid* function is used as activation function. If we assume two target classes 0 and 1, then 0 is predicted if the output is <0.5, and 1 otherwise. For multiclass classification, *softmax* activation is used, which normalizes the output of each class between 0 and 1 and the output values for all classes sum to 1, ie.e giving a probability distribution.

Training. For the training of a network, a **loss function** needs to be defined that will be minimized during the training process. For example, *binary cross-entropy* loss can be used for binary classification (formula 3.24), and *categorical cross-entropy* loss can be used for multi-class classification (formula 3.25); $\hat{y} = \hat{y}_1 ... \hat{y}_n$ denotes the classifier's output vector with size of n classes, and y denotes the one-hot encoded vector of the correct output class with size n which is the number of classes.

$$L_{binary\_cross\_entropy}(\hat{y}, y) = -y\ log(\hat{y}) - (1 - y)\ log(1 - \hat{y}) \tag{3.24}$$

$$L_{categorical\_cross\_entropy}(\hat{y}, y) = \sum_{i=1}^{n} y_i \, log(\hat{y}_i) \tag{3.25}$$

For training, i.e. learning how to update the weights such that the loss is minimized, **stochastic gradient descent (SGD)** algorithms are used. Goldberg (2017, p. 30) summarizes the function of the algorithms as follows: They iteratively estimate the loss over the training set, then compute the gradients of the weights with respect to this loss estimate, and then move the weights in the opposite direction of the gradient. Gradients are computed with back-propagation.

3.3.3. Word Embeddings: *word2vec* and *fastText*

In order to represent a word's meaning as vector in a high-dimensional space, count-based co-occurrence vector space models were long state-of-the-art. Broadly speaking, in these models a word is both represented as a dimension and as a vector. A word's vector entries are given by the counts of the word's context words in text. Thus, words occurring in similar contexts (and which potentially have similar meanings) should have similar vectors. Count-based co-occurrence vector space models were recently superseded by trained word vector space models, so-called word embeddings, but the underlying intuition remains the same. The most influential word embedding architectures so far were proposed by Mikolov et al. (2013a), commonly known as *word2vec*. Word2vec comprises two simple neural architectures, which train dense, low-dimensional word embeddings which capture many linguistic properties (e.g. gender, tense) and regularities (e.g. that $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman}$ results in the vector $\overrightarrow{queen}$ as nearest neighbor, Mikolov et al. 2013c).

***word2vec*: Skip-Gram with Negative Sampling.** The two *word2vec* architectures work in opposite ways for learning the word embeddings. *Skip-Gram with Negative Sampling* (SGNS) takes a target word as input and predicts the context words that surround it. The architecture of the model is shown in figure 3.5. In this example, the target word w_t is the input, and four context words, two on the left and two on the right side, are predicted. In reality, the prediction of each context word is a separate classification step. The neural network takes in the word and a potential context word, merges their embeddings with the dot product, and then predicts either a 1, if the word is a correct context word, or a 0 if it is not.

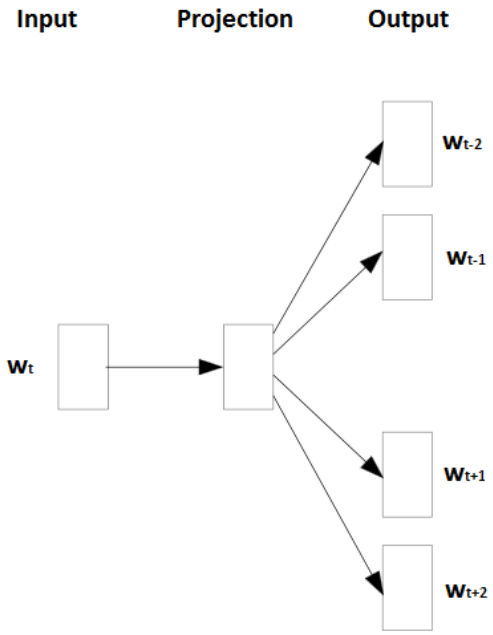


Figure 3.5.: The Skip-Gram architecture (q.v. Mikolov et al., 2013b).

To give a more formal definition, SGNS maximizes the objective

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-s\leq j\leq s, j\neq 0} log\ p(w_{t+j}|w_t), \tag{3.26}$$

which denotes the average log probability of a context word w_{t+j} given the target word w_t in a sequence of training words w_1 ... w_T. The parameter s denotes the size of the context window, e.g. if s=2 then the two words on the left and the two words on the right of the target word are considered as context words. $p(w_{t+j}|w_t)$ is computed by the softmax function. The softmax function is defined as:

$$p(w_c|w_t) = \frac{exp(\overrightarrow{v}_{w_c}{}^T \overrightarrow{v}_{w_t})}{\sum_{w_i\in V} exp(\overrightarrow{v}_{w_i}{}^T \overrightarrow{v}_{w_t})} \tag{3.27}$$

where w_c denotes the context word (the former w_{t+j}) and w_t denotes again the input target word, and $\overrightarrow{v}$ denotes the vector representation. The scalar product can be viewed as a scoring function, that scores the similarity between the target word w_t and its context word w_c.

$$score(w_t, w_c) = \overrightarrow{v}_{w_c}{}^T \overrightarrow{v}_{w_t} \tag{3.28}$$

However, the softmax is computationally inefficient, since it requires an iteration over the whole vocabulary in the denominator, because each word could be a possible context word. This is why *Negative Sampling* is applied, which means that the $log\ p(w_{t+j}|w_t)$ in formula 3.26 is instead computed as

$$log\ p(w_c|w_t) = log\ \sigma(\overrightarrow{v}_{w_c}{}^T \overrightarrow{v}_{w_t}) + \boxed{\sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)}[log\ \sigma(\overrightarrow{-v}_{w_i}{}^T \overrightarrow{v}_{w_t})]}, \quad \text{(3.29)}$$

negative samples

where $\sigma(x)$ denotes the sigmoid function as given in equation 3.20.

The sigmoid function simplifies the multiclass softmax classification from 3.26 to a binary classification task in formula 3.29.

Instead of considering the whole vocabulary, now the prediction probability is maximized for the positive sample (the correct context word) and k negative samples. Thus, fewer weight updates need to be done during training in each step. The negative samples are drawn from the noise distribution $P_n(w)$. One possible noise distribution is the unigram distribution. This means that k samples are randomly chosen, and the probability of each word to be chosen is equal to the probability of each word to occur in the corpus (i.e. the frequency of the word divided by the total frequency of all words). Mikolov et al. (2013b) tested several choices for the noise distribution $P_n(w)$, and found that the unigram distribution raised to the 3/4 power outperforms other distributions, such as the unigram distribution and the uniform distribution. Furthermore, their experiments indicate that k should be set between 5 and 20 for small training datasets, and between 2 and 5 for large datasets.

***word2vec*: Continuous Bag-of-Words.** The architecture of the Continuous Bag-of-Word model (CBOW) is based on a reverse principle as Skip-Gram, namely predicting a target word from its input words. The architecture of the model is shown in figure 3.6.

CBOW maximizes the objective

$$\frac{1}{T} \sum_{t=1}^{T} log\ p(w_t|C_t) \quad \text{(3.30)}$$

where C_t comprises the context words $w_{t-s}, w_{t-1}, .., w_{t+1}, w_{t+s}$, and the context window size is set as $2s$. The network takes the context words as inputs, averages their embeddings in the projection layer, and makes a classification using the softmax function (see formula 3.27).

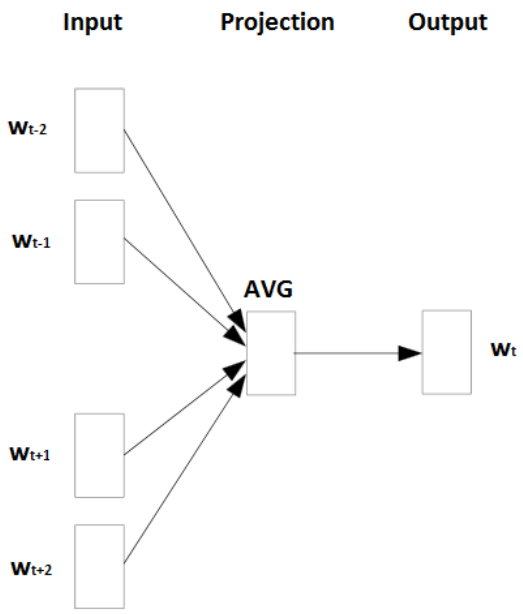


Figure 3.6.: The CBOW architecture (q.v. Mikolov et al., 2013b).

Thus, the scoring function is defined as

$$score(w_t, C_t) = \frac{1}{|C_t|} \sum_{w_c \in C_t} \overrightarrow{v}_{w_c}{}^T \overrightarrow{v}_{w_t} \tag{3.31}$$

fastText. Bojanowski et al. (2017) introduce *fastText*, which is based on Word2Vec, but works with character n-grams instead of words. This reflects the intuition that morphology is an important factor for word formation and word similarity. Furthermore, rare words, which would not have been assigned a word vector before, can now be represented by a combination of their subword units.

The fastText model generally relies on the Skip-gram model with negative sampling. In addition to learning word vectors, subword information is encoded in n-gram embeddings. Therefore, each word is represented as a set of n-grams, surrounded by word boundary symbols. Bojanowski et al. (2017) explains this by the following example because short words like 'her' are distinguished from parts of longer words such as 'where', the representations are '<her>' and '<wh, whe, her, ere, re>', using a fixed n-gram size of three (in practice, there is dynamic a n-gram size, ranging from 3 to 6). The vector representation of a word is then computed by the sum of the vector representations of its n-grams and the word embedding

itself:

$$\overrightarrow{v}_w + \frac{1}{|\mathcal{G}_w|} \sum_{g \in \mathcal{G}_w} \overrightarrow{v}_g, \tag{3.32}$$

with w denoting a word, and $\mathcal{G}_w$ denoting the set of n-grams forming that word. $\mathcal{G}_w$ is a subset of the whole set of occurring n-grams of size G, i.e. $\mathcal{G}_w \subset \{1, ..., G\}$.

Thus, the former Skip-Gram scoring function defined in 3.28

is substituted by the sum of all scalar products of an n-gram embedding with their contexts, which occur within one word:

$$score(w_t, w_c) = \sum_{g \in \mathcal{G}_{w_t}} \overrightarrow{v}_{w_c}{}^T \overrightarrow{v}_g \tag{3.33}$$

Bojanowski et al. (2017) compare fastText with word2vec. They find that using the subword information is especially beneficial for certain languages, among those being German. German contains grammatical declensions with four cases and many compound words, which results in more word types for the same or similar words. Because of the fastText subword information, a compound word like *Tischtennis* ('table tennis') is now partially represented by the same vectors as *Tennis* ('tennis'), and the words exhibit a certain amount of similarity. In comparison, for word2vec, both words are represented by completely different vectors, where the similarity is based on similar contexts and not on the word form itself. In our experiments, we use word2vec as the default vector representation. However, we lay a special focus on German closed compounds. In a first experiment (section 6.2) we split compounds and use constituent embeddings as additional information for classification. In a later experiment (section 6.3), we also use fastText since it has the above mentioned advantageous attributes regarding closed compounds, and compare it to word2vec.

3.4. Summary

The chapter dealt with concepts, methodologies, data and evaluation measures used for automatic term extraction. Methodologies for automatic term extraction include unsupervised as well as supervised methodologies, such as term extraction measures and machine learning approaches. Furthermore, we described semantic word representations and machine learning methodologies that are commonly used for natural language processing tasks, and which are important basics for the experiments in this thesis.

4. Human Annotation Studies to Investigate Term Definition

4.1. Introduction

Not only automatic term extraction, also **manually identifying terms** in text is a notoriously difficult task; Estopà (2001) shows that experts with different perspectives on terminology (e.g., terminologists, domain experts, translators and documentalists) vary significantly in their annotation of terms. We find a range of gold standard corpora for the evaluation of term extraction systems for English (Ohta et al., 2002; Bernier-Colborne and Drouin, 2014; QasemiZadeh and Schumann, 2016) and to a lesser extent also for German (Arcan et al., 2014; Arcan, 2017), but these benchmark datasets vary hugely in terms of granularity of term definition, topic and thematic focus. Further, not only term definition itself, also the nature of the **domain** can be a problem: For example, a domain can be characterized by a broad range of topics, and a heterogeneous text corpus can in addition cover several registers.

In this chapter, we examine the concept of terminology from new perspectives. We want to find out how people perceive terminology, and as a consequence how we can improve strategies for manual annotation that leads to an increase in agreement. 'Agreement' denotes the degree to which different test persons identify the same words or phrases as terms respective a domain. We choose our test persons from different groups of people, lay people and semi-experts. However, all test persons have in common that they are German native speakers and are given German texts or term candidates. Concerning the latter, text sources are diverse (handbooks, manuals, user-written texts, and at one time even forum texts), and we always choose several domains (cooking, DIY, hunting, chess). The domains have different peculiarities. For example, hunting is rather specific to the hunting community, while cooking has a huge overlap to the daily life. DIY is a thematically broad domain. All domains contain ambiguous terminology, but this is especially striking for the hunter's language, where a lot of general-language vocabulary has a different, domain-specific meaning. In sum, we aimed at selecting the text material in a way that we can investigate terminology broadly. We conduct

three studies to address that goal, which define the structure of this chapter:

In the first section, we examine the concept of terminology from scratch again. Since even experts disagree on the understanding of terminology, this raises the question whether there is a common, natural understanding of what constitutes a term, and to what extent this term is associated to a domain. We ask the question if there is a natural intuition of terminology or if it is an artificial, expert-defined construct. For this reason, we conduct a first user study with asking **lay people** for their understanding of what constitutes a term. We do not give direct instructions to identify terminology, but ask implicit questions to get an intuitive notion of terminology.

The second section then describes a **user study with semi-experts**. After letting lay people identify terms without explicit instructions, we now want to find out how we have to establish a set of rules to increase agreement for term annotation. Therefore we iteratively design a scheme for terminology annotation that clearly defines the annotation procedure. This time annotators have some knowledge about terminology and the domain in general. The task involves the special challenge that we focus on the DIY domain, which requires to deal with a lot of heterogeneous data.

Up to this point, we experimented with different groups of people and different instructions for identifying terminology. In the third section, we vary another parameter and experiment with term definition itself. One intuition we have is that term definition is too broad; we always identify terms, in other words we only have the two categories 'terms' and 'non-terms'. This might be problematic for annotators because they might make more fine-grained distinctions. Therefore, a **new fine-grained model for characterizing terminology** is introduced, that **overcomes the binary term understanding** (the separation into only two categories, 'term' and 'non-term'). Such a fine-grained model can be more flexibly used for multiple follow-up applications. Again, a lay annotation study tests the robustness of the model. Finally, we discuss strengths and weaknesses, and what the results of the study means for the following automatic term extraction experiments.

4.2. Lay People Study on Terminology Identification

In this study, we examine the concept of terminology from a new perspective[1]. Contrary to previous annotation studies, we investigate judgments of lay people, rather than experts, and focus on analyzing their (dis-)agreements on common assumptions and core issues in term identification: the word classes of terms, the identification of ambiguous terms, and the rela-

[1]The work in this section is published in Hätty and Schulte im Walde (2018b).

tions between complex terms and possibly included subterms. To ensure a broad understanding of term identification, we designed four different tasks to address the granularities of term concepts, and we performed all annotations across four different domains in German: DIY, cooking, hunting, chess. Finally, we compare the annotations to the output of an unsupervised hybrid term extraction system.

4.2.1. Material and Tasks

Domains. The data for term identification comprise German open-source texts from the websites *wikiHow*[2], *Wikibooks*[3] and *Wikipedia*. All texts have been pos-tagged with the *Tree Tagger* (Schmid, 1994); compound splitting was performed with *Compost* (Cap, 2014) and manually post-edited. In total, the text basis consists of 20 texts (five per domain) with ≈5 sentences each. All texts together contain 3,075 words, distributed over the following four domains:

- DIY: *"do it yourself"* (708 words)
- cooking (624 words)
- hunting (900 words)
- chess (843 words)

Term Identification Tasks. In order to investigate the effect of term definition on term identification, we specified the following four tasks:

- highlight domain-specific phrases (**DS**)
- create an index (**IND**)
- define unknown words for a translation lexicon (**TR**)
- create a glossary (**GL**)

We assumed the four tasks to provide different strengths of associating the terms with the domains: DS and IND were expected to demand a broad range of terms that characterize the domains. TR and GL were expected to have a focus on unknown and ambiguous terms. We can thus test the reactions onto these different kinds of tasks. Will people identify most terms

[2] `https://de.wikihow.com/`
[3] `https://www.wikibooks.org/`

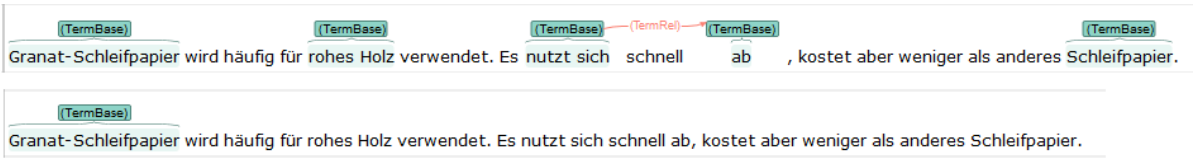


Figure 4.1.: example of WebAnno annotation for DS (top) and GL (bottom)

for DS and least for GL? How will people react to ambiguous terms, will they find or oversee them?

20 annotators were asked to perform only one of the identification tasks, which resulted in five annotations per task. In addition, we asked two annotators to perform all four tasks, to check whether the inter-annotator agreement differs in the two setups. Since the latter annotation setup did not exhibit systematic differences to the original setup, we merged the results of all seven annotations.

Annotation was done using *WebAnno* (Yimam et al., 2013), a general-purpose web-based annotation tool. We allowed annotations of single words, multi-words, and links between terms in case of nonadjacent term constituents. An example of two annotations is shown in figure 4.1. In addition to the actual annotation, annotators were asked to rate their knowledge about the respective domains. Overall, cooking was rated as best-known domain, with a mean of 6.86 on a scale from 1 (unknown) to 10 (well-known), followed by DIY (5.18), chess (4.05) and hunting (1.90).

4.2.2. Analyses of Term Identification

In the following, we analyze word forms annotated as terms, across tasks and across domains. As the central means in our analyses, we make use of the *agreement* between annotators. We rely on simple agreement (how many of the 7 annotators per task agreed?), the Jaccard index and the chance-corrected agreement measure Fleiss' κ (Fleiss, 1971). We start with various single-word type-based evaluations, and then explore multi-words afterwards.

4.2.3. Agreement across Tasks and Domains

Table 4.1 shows the number of type-based term annotations per task with the highest agreements, i.e. where all annotators (7) or most annotators (6 or 5) agreed.

In line with our intuition, the number of identified terms is highest for DS, and lowest for GL, with IND and TR in between.

task	DS	IND	TR	GL
agree = 7 (*all*)	203	66	94	27
agree $\geq$ 6	315	111	173	68
agree $\geq$ 5	400	148	247	117

Table 4.1.: Number of identified terms per task.

This trend is still obvious when including all annotated terms (i.e., all term types annotated by at least one annotator): Figure 4.2 shows the Jaccard index across tasks and domains, i.e., the intersection (agr.=7) of the annotations divided by their union (agr.$\geq$1). DS again receives the highest values, and GL the lowest. DS and GL thus seem to represent the extremes of the tasks, with DS providing the broadest and GL the narrowest definition of terminology.

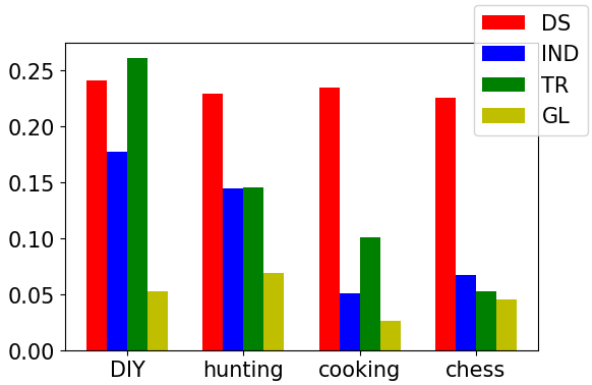


Figure 4.2.: Jaccard index across tasks and domains.

Across the tasks and different scopes of the terms, there is however a clear tendency for the same terms to receive high vs. low agreement. This effect is shown in figure 4.3, where all annotated term types are depicted in a four-dimensional space (x-, y- and z-axis plus the 4th dimension in colour). Each dimension represents one task, the value in each dimension represents the agreement on terms for this task (max. 7). We clearly observe an upward-moving tendency for term agreement across all dimensions, for example, across the four tasks, annotators (dis-)agreed on the same terms to a similar degree. We conclude that annotators have similar intuitions about a term's domain specificity regardless of the term identification task.

Figure 4.4 depicts the interaction between tasks and domains even more clearly: While Fleiss' κ for DS is in general very high across domains, and also IND and TR are well-agreed upon for DIY (and TR for hunting), the κ values for GL are particularly low, and so is IND

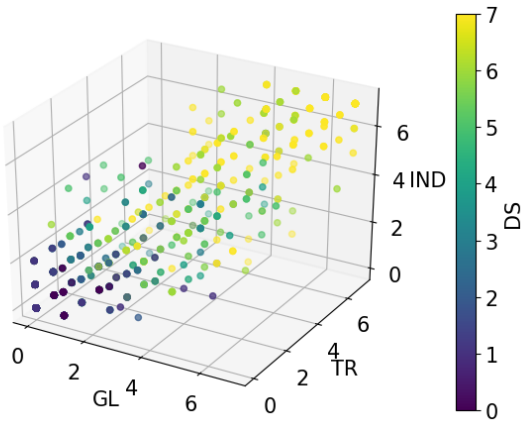


Figure 4.3.: Term agreement across tasks.

for cooking and chess, and TR for chess.

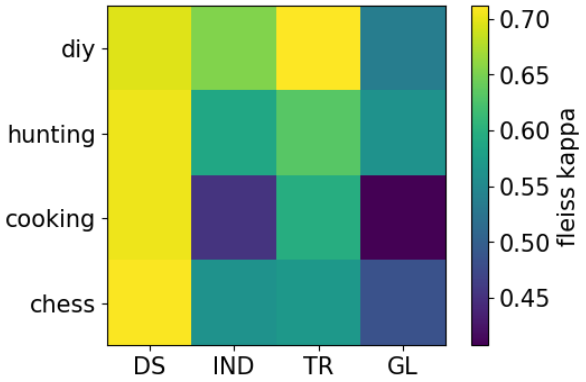


Figure 4.4.: Fleiss' κ across tasks and domains.

Term Identification across Word Classes

Traditionally, mostly nouns are perceived as terms (Bourigault, 1992; Justeson and Katz, 1995), and consequently annotation and extraction of terms is often restricted to noun phrases (Kim et al., 2003; Bernth et al., 2003). However, according to Estopà (2001) and others, terminology should not be restricted to noun phrases. Figure 4.5 shows that both views have a point. The figure shows the number of term type annotations for nouns, verbs and adjectives across the 28 annotated datasets (7 annotations $\times$ 4 domains). For example, roughly 300 noun types received a total of 5 term annotations across the four tasks DS, IND, TR and GL. We can see that in our dataset nouns are indeed preferred by our non-expert annotators. However,

with a fewer amount of annotations, the number of annotated verbs and adjectives rise. Looking into the data revealed that 70% and 58% of the annotated verbs and adjectives appear in multi-word terms (MWTs). One reason for this is their participation in annotated activities such as *großes Loch reparieren* ('repair a big hole') or *Eigelb schaumig schlagen* ('beat the egg yolk until fothy').

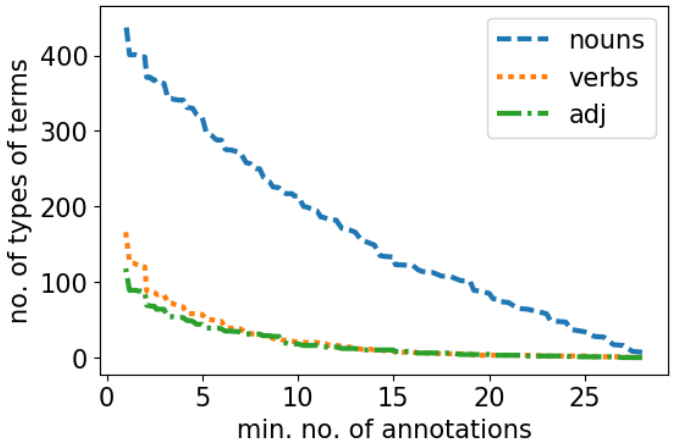


Figure 4.5.: Annotations per part-of-speech.

Complex Terms and Subterms

The fact that multi-word terms often contain subterms is a distinct attribute, frequently exploited by automatic term extraction methods relying on term constituent phrases for computing a termhood score (Frantzi et al., 1998; Nakagawa and Mori, 2003). In our study, 468 single-word terms, 138 closed compounds and 692 MWT types were annotated across annotators. Since German contains many closed compounds, treating them separately from MWTs (consisting of several separated words) is especially interesting: a compound term candidate is either annotated completely or not at all. Regarding MWTs, it is possible that only a subterm is annotated. For example, the compound *Rohholz* ('raw wood') cannot be separated, while annotators might mark only *Holz* as subterm of the MWT *rohes Holz*.

Table 4.2 shows aspects of multi-word and compound term types in relation to their number of annotations (7 annotators $\times$ 4 domains, i.e. a maximum of 28 annotations), across tasks and domains. We group the number of annotations into three categories: no concordance (<2), minimum concordance (≥ 2) and majority concordance (>14). For most MWTs (426), there is no concordance, and only a few MWTs were annotated by the majority (11). Compound terms show the opposite behaviour. Slightly more than half of the compounds (76) were annotated

no. of annotations	*<2*	*≥2*	*> 14*
no. of **MWTs**	426	266	11
% of subterms	49.23	57.40	45.83
∅ annot. on subterms	7.53	7.26	6.0
no. of **compounds**	6	132	76
% of subterms	16.67	31.76	40.37
∅ annot. on subterms	1.0	9.59	10.23

Table 4.2.: No. of annotations for compounds and MWTs. MWTs make up 53% and compounds make up 15% of the terms identified in total.

by the majority of annotators, while only 6 compounds were only annotated by one person. Thus, annotators are confident in identifying compound terms, but not MWTs.

We then analysed the annotation concordance of complex term constituents, and their likelihood to represent a subterm. We extracted all annotated single-word terms (SWTs) which were not annotated as part of a complex term. While for MWTs the proportion of subterms is relatively high across categories, there is a radical increase in number of compound subterms, when increasing the concordance.

Finally, the subterm concordance (i.e., their average annotations) for MWTs drops with an increasing concordance for MWTs. Compounds, again, behave in the opposite way. Thus, the less confidence there is for an MWT, the more confidence we find in its subterms. For the closed compounds, this effect cannot be perceived.

Ambiguity

A peculiarity of many terminologies are general-language words with a specialized meaning in one or more domains. For example, the English noun *solution* has a general-language sense as well as domain-specific senses in mathematics and chemistry (Baker, 1988). Ambiguous vocabulary is also present across our domains, e.g., *Fuchsschwanz* ('ripsaw'/'foxtail') in DIY and *ansprechen* ('identify game'/'address so.') in hunting.

In order to analyze the identification of ambiguous terms, we first looked up the general-language and domain-specific senses of all hunting and chess terms from our dataset in *Wiktionary*[4], *Duden*[5], and the *Wikipedia* disambiguation pages. We did this for hunting and chess, because only these domains are consistently specified in the sense definitions. We identified

[4] `http://www.wiktionary.org/`
[5] `https://www.duden.de/`

domain	DS	IND	TR	GL
hunting	5.32	3.74	4.12	3.44
chess	5.08	3.72	3.75	2.93

Table 4.3.: Average agreement on ambiguous words.

18 terms for hunting and 15 for chess (as a comparison: there are 425 word types for hunting, and 360 for chess).

Table 4.3 shows the average agreement on these ambiguous words, across tasks. For example, on average 5.32 annotators out of 7 agreed on the 18 hunting term types in the DS task. The table shows that the average agreement is higher for DS than for the other three tasks.

We conclude that when it comes to a stricter sense of termhood the domain-specific sense might be perceived by the annotators, but the general-language sense impedes them to accept the same strength of termhood for the ambiguous term as for other, more domain-specific terms.

4.2.4. Automatic Term Extraction

In a final step, we compared the identification of terms in our dataset against the identification done by state-of-the-art term extraction approaches. We used the hybrid term-candidate extractor for the DIY domain described in Schäfer et al. (2015) and Rösiger et al. (2016). After lemmatization and pos-tagging, the system extracts terms with predefined linguistic filters. For term candidate ranking, standard termhood measures are applied, cf. an overview in Schäfer et al. (2015).

Approximately half of our annotated terms were found by the term extractor (due to predefined linguistic patterns for extraction). Based on the measure scores, we applied the Spearman's ρ (Siegel and Castellan, 1988) to compare against a ranking based on annotator agreement. The best ρ values were 0.51 and 0.44 for two corpus-comparison extraction methods; these two are statistically significant ($p< 0.01$).

When inspecting the ranked list, we observed that the term extractors rank compounds and MWTs higher than the lay people do. Although the automatic extractors only use statistics over the whole word forms, ρ increases when adding subterm scores to compounds and MWTs. This again indicates the importance of subterms within complex terms for an annotator's decision.

4.2.5. Conclusion

This section presented a study about term identification by lay people, across four domains and four task definitions. We found that lay people generally share a common understanding of termhood and term association with domains, as reflected by inter-annotator agreement. With this result we have shown that terminology is an intuitive construct, and not purely expert-defined. Furthermore, (i) high inter-annotator variance for more specific tasks, (ii) little awareness of the degree of termhood of ambiguous terms, and (iii) low agreement on multi-word terms with high reliance on subterms showed that lay people's judgments deteriorate for specific and potentially unknown terms. Ultimately, although people show a common intuition about terminology, there is still much disagreement, especially for multi-word expressions. However, annotators agree to a much higher extent on closed compounds than on other form of multi-word expressions, which is most likely due to the clear phrase boundaries.

4.3. Semi-Expert Study on Terminology Identification

In the last section, we questioned the concept of terminology as a whole. We asked the question if terminology is intuitive, if people with no prior knowledge about terminology can still share a common intuition about it. We answered this question with yes. However, although this is an encouraging result, identifying terminology remains a difficult task for humans. This became apparent because there was still much disagreement among annotators. However, in the last study, annotators had nearly no instructions for how to annotate. Thus, in this study, we want to experiment with developing term annotation instructions. We aim at finding a set of instructions to guide annotators more concretely, but still want to remain general enough for that the guidelines can be transferred to other domains. These instructions will be refined in several annotation rounds, in order to get feedback from the annotators. We use again challenging text material as basis for the task: texts from the domain of do-it-yourself (DIY), for example instructions and reports, are chosen as basis. As described earlier, the DIY domain is characterized by a broad range of topics, and our text corpus covers several registers in addition. This results in the presence of term candidates with different status and poses a special challenge to the annotation approach. In the following, we describe the experiment as well as the decisions we made for creating the annotation constructions. Finally, we evaluate the overall annotation result[6].

[6]The work in this section is published in Hätty et al. (2017b).

corpora	ACL 1.0	ACL 2.0	B/C	Bitter	TTC	Genia	Craft	our approach
breadth	**	**	**/*	**	**	*	**	***
registers	*	*	*	*	**	*	*	***
token-based	-	+	+	+	-	+	+	+
guidelines	broad	broad	mid/strict	broad	mid	strict	strict	mid

Table 4.4.: Comparison of terminology gold standards.

4.3.1. Related Work

Existing Benchmark Datasets for Term Extraction. There exist a range of terminology benchmark datasets which vary in the specificity of their topic, their definition of termhood and writing styles. Well-known datasets are the **Genia** (Kim et al. (2003)), the **CRAFT corpus**, (Bada et al. (2012)) and **ACL RD-TEC** 1.0 (Zadeh and Handschuh (2014)) and 2.0 (QasemiZadeh and Schumann (2016)). These corpora are described in the background section (3.2.1).

There are further corpora and term annotations: Bernier-Colborne and Drouin (2014) (B/C) analysed three textbooks on automotive engineering. In addition to the annotation, they assign attributes to the terms (e.g. for acronyms or multiwords) and mark orthographic variants. Other reference sets consist of bilingual term lists to evaluate machine translation. In the **TTC project** (Loginova et al. (2012)), a list of term candidates is generated with a term extraction tool and then further evaluated by experts. In the **BitterCorpus** (Arcan et al. (2014)), terms are annotated in texts from KDE and GNOME documentation corpora. Both the TTC term list and the BitterCorpus contain German terms, among other languages. In the following, we compare the reference datasets with respect to the size of their domain, the registers represented and the underlying annotation approach (see also Table 1).

Domain. The reference datasets differ with respect to breadth of the topics covered. Genia's domain is very narrow, it is specialized to biological reactions concerning transcription factors in human blood cells. The texts are crawled on the basis of three seed terms. With Bernier-Colborne and Drouin, the topic is automotive engineering as presented in three textbooks for lay people. For CRAFT and ACL RD-TEC, journal and conference articles have been taken from a wide range of subtopics in their respective domains, and different research areas of the domains are included in the text basis. The same holds for the BitterCorpus: In the GNOME and KDE manuals, a range of topics, such as the user interface, settings, the internet

connection or information about hardware are addressed. All these corpora have clearly defined content since the extraction basis is hand-selected. This does not hold for the TTC texts, which are retrieved by a thematic web crawler; unexpected text can thus occur in the corpus. The topics of our own data are even more open: The DIY domain is broad in itself, and as the texts come from different sources, the variety of topics even increases. Several slightly off-topic texts are part of the text basis.

Register. Most of the gold standard corpora are homogeneous with respect to register. They either consist of scientific articles (Genia, CRAFT, ACL RD-TEC 1.0 and 2.0) or of instruction texts: The three expert-to-lay textbooks for automotive engineering used by Bernier-Colborne and Drouin (2014) might differ slightly from author to author, but nevertheless have the explanatory style of textbooks. Finally, the KDE and the GNOME documentation follow the style of online manuals. Different registers only occur in the crawled text of TTC. In our work, we deliberately choose texts from two putatively different registers; we distinguish them in terms of the intended public and sampled the text basis in a way that expert-to-expert writing and user generated content (= UGC) are both represented (60:40%).

Annotation Approach. The definition of termhood is widely divergent across the different gold standards. In Genia and CRAFT, the annotation is very strict, as specific syntactic patterns and semantic constraints are given. Both the work by Bernier-Colborne and Drouin (2014) and the TTC terms have a more liberal annotation scheme, partly following the rules proposed by L'Homme (2004). Bernier-Colborne and Drouin (2014) limit the annotation semantically to items denoting components of cars, and for TTC, term candidates were preselected by a term extraction tool. For the ACL RD-TEC gold standards and the BitterCorpus, the definition of termhood is particularly liberal, as termhood is rather loosely defined. They mainly rely on the association an annotator has with respect to a term or to a domain (e.g. by structuring terms in a mindmap) and provide theoretical background about terminology.

For our work, we aim at a compromise between generality of annotation and restriction of outgrowths. Because of the breadth and the stylistic variability of the DIY text basis, we do not set strict rules for the annotation, e.g. by limiting the syntactic or semantic shape of terms by predefined POS-patterns or predefined ontology elements onto which the terms would have to be mapped. However, we give positive and negative examples, and guiding rules elaborated after extensive discussion about the relation of DIY terms to their domain.

corpora	total	used	corpora	total	used	corpora	total	used
wiki	4.31 * 10^5	30,915	FAQs	4,805	347	project	2.16 * 10^6	2,701
expert projects	55,430	3,971	encyclopedia	6,059	449	forum	2.34 * 10^7	29,293
marketing texts	35,452	2,540	book	54,005	3,868			
tips and tricks	12,711	904	tool manuals	69,831	5,012			

Table 4.5.: Distribution of tokens by subcorpus: expert (two left-most) and user texts (right).

4.3.2. Corpus and Domain: from User-Generated to Standard Text

We use a corpus of German texts of the DIY domain, which is thematically about non-professional builds and repairs at home. There are different text sources available, containing texts produced by domain experts as well as by interested lay users. The latter mainly consists of forum posts collected from several online DIY-forums, e.g. from project descriptions or inquiries for instructions[7]. Expert texts include an online encyclopedia and a wiki for DIY work, tools and techniques[8]. The corpus used for the work described here contains ca. 11 M words in total, with 20% expert text vs. 80% user-generated data.

For the manually annotated part, we aim at a balanced extraction of text data from all the different sources. Thematically, we only excluded gardening activities, which we do not see as a part of the DIY domain. The corpus is balanced to include 40% user texts and 60% expert texts. In total, 80,000 tokens are extracted. Since we annotate terms in context (token-based), complete sentences are extracted. We thus sample subcorpora proportionally to their orginal size, to reach a total of 48,000 tokens of expert text plus 32,000 tokens of UGC (see Table 4.5). All sentences are shuffled.

4.3.3. Annotation

Procedure and Design of Annotations

General Procedure. The annotation guidelines and the actual data annotated were created in discussion rounds with 6 to 7 participants who have experience in terminology extraction. All are semi-experts of the domain, because they have been dealing with terminology extraction from the DIY domain for more than one year. The guidelines were influenced by terminology theory, peculiarities observed when analysing the text data and practical issues,

[7] e.g. `www.1-2-do.com/forum`
[8] e.g. `www.bosch-do-it.de/de/de/bosch-elektrowerkzeuge/wissen/lexikon/`

to ensure a consistent annotation. 40,000 tokens are annotated in total, by two of the above participants. The actual annotation is being produced by three (of the above) annotators.

Annotation Design. We use WebAnno (Yimam et al. (2013), de Castilho et al. (2016)) as an annotation tool again. For this annotation design, possible annotations are **spans** (for single- and multi-word terms) and **relations** (used here to link separated parts of a term). For the spans, several **values** can be chosen: *domain, domain-zusatz* and *ad-hoc*. While most terms are annotated with *domain*, we use *ad-hoc* for user-coined terms, and *domain-zusatz* (= domain-additional element) for elements that are themselves not terms, but are parts of multi-word expressions, e.g. the adverb *freihand* in *freihand sägen*.

Tiers of Terminology and Consequences for the Annotation Approach

The annotation of benchmark sets for terminology is typically implemented as a binary decision. However, it is widely acknowledged that the terminology of a domain is a rather inhomogeneous set. It can be divided into several tiers, e.g. with a distinction between terms which only occur in the very specific vocabulary of a small domain, as opposed to terms which occur with an extended or specialized meaning in one domain but also in other domains or in general language (e.g. Trimble (1985), Beck et al. (2002)). For example, the model by Roelcke (1999) consists of four layers (a description of the model can be found in background section 2.2.2).

Our annotation approach is liberal and our notion of 'term' comprises the first three layers of Roelcke's model (*intra-subject terminology*, *inter-subject terminology*, *extra-subject terminology*), and only Roelcke's *non-subject terminology* is not considered as terms. This means we acknowledge a certain heterogeneity of terms - even a scalar character of terms, which the lay people study showed is the natural intuition people have about terms - but still remain with the distinction 'term' and 'non-term' for the moment. We therefore keep the 'term' category broad, and capture differently strong terms in it.

Breadth of the Domain: Terminological Richness in the DIY-Domain

The DIY domain is influenced to a high degree by other domains. There is a quite obvious core set of terms which are prototypical (e.g., *drill, fretsaw, circular saw bench, ..*). In addition, there are many terms borrowed from other domains, e.g. from material science or construction techniques. In our annotation, we distinguish between **terminology *borrowed* from other domains** and **terminology from neighbouring domains**. While texts with intra-subject or

inter-subject terms tend to centrally belong to the DIY domain (and describe what we consider to be "typical" DIY activities), borrowing takes place from related domains, and knowledge about them is necessary for efficient communication in the DIY domain, such as some fields of physics, of material science, construction techniques, etc. That means borrowed terms are relevant for DIY, while terms from neighboring domains are not necessarily used in DIY, but are similar due to their origin. We consider fields as neighboring domains where DIY-realted activities are carried out professionally, such as sanitary, electrical or heating engineering. Sentences belonging to texts describing work of this kind are disregarded in our annotation.

Registers: User Language and Jargon

Apart from the broad domain, the wide range of registers is a challenge for annotation. In the user-generated texts, misspellings and user-coined terms (e.g. *Selberbauer, reinhämmern, Filterabrüttlung, mit Hobelmesser 'abgemessert'*) have to be addressed. We mark them with the special label *ad-hoc*, to show their terminological relevance but to distinguish them from accepted terms.

The way in which DIY-forum users talk about tools and materials shows their high degree of specialization, even in texts that exhibit signs of conceptual orality (in the sense of Koch and Oesterreicher, 1985). In the 40.0000 words, we identified 71 references to tools in which a highly specialized DIY knowledge is presupposed.

From the standard (expert) text in the domain, we observe that the official denomination of power tools mostly follows a rigid pattern. The names are composed of [BRAND][TYPE][MAIN DESIGNATION][SECONDARY DESIGNATION], for example *Metabo Kappsäge KGS 216 M* or *Bosch Tischkreissäge PTS 10 T*. An intuitive way of abbreviating these denominations would be by the type; instead we find highly specific references, close to in-group jargon:

- 16 times the tool was only referenced by its brand name (e.g. *meine Makita, Metabo,* ...);
- 24 times by its main designation (*IXO, PBS,* ...);
- three times by its secondary designation (*0633 ohne Motor, 900er*);
- and 28 times by a combination of main and secondary designation - of different granularity and written in different forms (*GKS 68 BC, PCM8S,* ...).

This special term use increases the number of term types and poses a challenge for automatic term extraction, as well as for coreference resolution in that domain. Furthermore, this way of referencing supports the claim that embedded terms need to be addressed in the manual annotation. Whether a term extraction tool which is sensitive to embedded terms can also

identify this kind of references is still an open question. There are less regular references as well, e.g. abbreviations by material (*ODF* instead of *ODF-Platte*), missing size units (*35er Scharnier*), or only sizes are mentioned (*K60-K220* instead of *Schleifpapier der Körnungen K60, K80, .., K220*)). Other special cases are jargon-like abbreviations (*TKS = Tischkreissäge*, *OF = Oberfräse*, *HKS = Handkreissäge*).

Another characteristic of user texts is the almost infinite number of domains from where terms can be borrowed: when being creative, everything can be used to do handicrafts with, everything can be (mis)used as a tool or material (*Frühstücksbrett in Fliesenoptik; Geschenkboxen aus Käseschachteln, gedrechselte Kirschen*). Items from these other domains fill areas in DIY which are prototypical, e.g. DIY project names, materials and tools. This makes it harder to decide whether these items are terms. That topics are spread more widely can be shown by the number of sentences annotated in the 40.000 words corpus. In the user-generated content (UGC) part, 45.36% of the sentences are annotated, in the expert texts 66.21%. Furthermore, the density of term annotation is higher in the expert texts: in the UGC texts, 9.15% of the tokens are annotated, in the expert texts 17.08%. We give preference (60/40) to the richer type of data.

Annotation Approach: Multiword Terms and Term Variants

A special focus of the annotation is on multi-word terms (MWTs). We aim to preserve as much of the terminological content in the data as possible. By allowing annotation of discontinuous multi-word terms, we enrich the term base.
Besides annotating adjacent MWTs, we also capture MWTs interrupted by terminologically irrelevant material. In *scharfes und gefährliches Messer* (sharp and dangerous knife) *und gefährliches* will not be annotated, while *scharfes Messer* is considered as a term. This annotation is realized by linking together the separate parts of the MWT. A similar case are MWTs which are interrupted by additional terminological material, e.g. *schwebender (schwibbender) Bogen*, from where two terms can be created by linking: *schwebender Bogen* and *schwibbender Bogen*.
Contrary, e.g. to TTC, we annotate all valid embedded terms. For example, for *freihand gebrochene gerade Kante*, the whole term, *gerade Kante* and *Kante* are annotated.

As we aim at covering all possibly terminologically relevant material, we do not a priori set restrictions as to the length or POS pattern of term candidates. Anyway, collocational verb-noun pairs (*Holz fräsen, mit Nägeln verbinden*) are not annotated as multi-word terms. We aim at distinguishing them from terms because otherwise there would be an outgrowth of

terms (while sentences could become terms then). However, this annotation decision leads to an inconsistency at the theoretical level, if the verb-noun pair occurs in its nominalized form (*Nagelverbindung*) and we consider it as a term then. As a consequence, we annotate the noun compound form and have this inconsistency; to attenuate this conflict, we also allow idiomatic verb-noun combinations to be annotated. For example in *auf Gehrung sägen*, *auf Gehrung* is annotated as **domain-zusatz** ('domain additional element') to *sägen*.
Our annotation keeps track of the variety and complexity of syntactic structures in which terms can appear in texts, including non-adjacent parts of multi-word expressions.

4.3.4. Evaluation

Inter-Annotator Agreement

Fleiss' kappa (Fleiss (1971)) is used to calculate the inter-annotator agreement. In our annotation, multi-word terms, parts of terms and different annotation labels have to be considered. In the 40,000 tokens annotated in total, 2514 single-word terms (SWTs) and 511 MWTs are identified by one annotator, 4269 SWTs and 1353 MWTs by the other one. An item can have several more than one labels. Thus, we introduce an IOB format for the terms (term-internal, out-of-domain, beginning of a (multi-word) term) and consider the annotation to have 7 labels: IB * labels *domain, ad-hoc, domain-zusatz* and O.

Fleiss' kappa is calculated for every label and the result is averaged. We achieve an inter-annotator agreement of 0.81 which is a substantial agreement according to Landis and Koch (1977).

Error Analysis: Consistent Differences in MWTs Annotation

Despite our strategy to encourage the annotation of MWTs as well as of their embedded terms, we still find consistent differences in this regard. Two kinds of structural inconsistencies are prevalent:

Adj N. In 151 out of 455 adjective-noun sequences annotated in total (by either of the annotators), one annotator annotated the whole phrase while the other one annotated only the noun. When analysing the relevant phrases, it is striking that in these cases the adjectives are evaluative (*handliche Fräse*), uninformative (*gängiger Handhobel*), underspecified dimension adjectives (*präziser Schnitt*) or related to the given situation (*vordere Schleifplatte*).

N Prep N. In 17 out of 86 cases a noun-preposition-noun phrase is annotated as one stretch by one annotator while the other annotator distinguishes between two single word terms. This set consists of nominalized verb-object pairs (*Schleifen von Kanten*), positional descriptions (*Querlöchern in Holzwerkstoffen*) and purpose constructions (*Sägeblätter für Porenbeton*).

We could refine the guidelines down to individual syntactico-semantic patterns (e.g. positional vs. purpose N Prep N groups), but this would not allow us to take the linguistic creativity of the forum authors into account. Similarly, the vagueness of underspecified dimension adjectives seems rather to be the typical property of the style of our texts. As a consequence, the terms extracted from the forum data can at best be partly organized in ontologies.

4.3.5. Conclusion

We presented work for an annotation study with semi-expert annotators. A special focus was laid on the design of the annotation guidelines and maintaining a liberal term definition. Challenges for annotation were the breadth of the domain and the register variety in our corpus. The corpus was characterized by its heterogeneity, as illustrated by a comparison of expert and user-generated text: User-generated text both has a lower density of terms than expert text (expectably) and jargon-like intra-community terminology. The domain as well as the text characteristics of UGC require specific provisions for the different tiers of terminology they contain (e.g. borrowed terms, neighbouring domains).
Our annotation approach was liberal, yet based on guidelines that were defined in a repeated discussion and annotation process. We paid special attention to the annotation of multi-word terms including discontinuous ones. We achieved a substantial inter-annotator agreement for the annotation. However, we find structural inconsistencies in the annotation results, and theoretical discrepancies for defining terms which contain verbs and nouns.

4.4. Investigating Term Characteristics

4.4.1. Defining Centrality and Specificity

From the previous studies, we gained important insights with respect to the question what constitutes a term. The lay people study shows that the property to be a term is implicitly perceived as scalar and not binary across all annotators. Agreement for term candidates slightly decreases, instead of that agreement divides term candidates into two clear-cut classes, a class

with high agreement term candidates (potential terms) and a class with low agreement ones (potential non-terms). Besides, even across term identification tasks, people agree on some term candidates more than on others. For the semi-expert study, we already acknowledged the heterogeneity of terms, and based the definition of terms on different tiers. However, the heterogeneity of terms became even more evident for the heterogeneous domain of DIY (as shown by the given term examples) and the annotation guidelines that were refined in an iterative process were quite detailed in the end. This effect was clearly due to the complicated nature of terms, but was at odds to some degree with the intuitive and liberal annotation approach we wanted to pursue.

In sum, the two studies let us come to the conclusion that we need to test for a gradual instead of a binary intuition about terms; that means we need to explore an **extended term definition**. Even more, the extended term definition might not necessarily constitute a one-dimensional scale, but two characterizing dimensions: from the previous annotation studies, we gained the impression that two different kinds of characteristics can be responsible for the scalar association of a term. We give two examples:

a) *Akkuschlagbohrer* 'cordless percussion drill', *Bohrer* 'drill bit' for the DIY domain

b) *Hubkolbenmotor* 'reciprocating piston engine' and *Abgasgesetzgebung* 'emissions legislation' for the automotive domain

For the first example, *Akkuschlagbohrer* is a more spezialized term for DIY than *Bohrer*. When asking a lay person, he would most probably have a clearer idea of the latter word than of the first one. For the second example, *Hubkolbenmotor* is specialized, as well as *Abgasgesetzgebung* - but the latter expression is thematically less related, and could also be associated to law instead of the automotive domain. In sum, we find two term characteristics here: specialization and thematical association. Interestingly, these are also the two characteristics that Pearson (1998) finds to be underlying the tier models (background section 2.2.4); Pearson (1998) noted that there are two features that influence the tier models (her 'pragmatic definitions of terms'): familiarity of a term, and subject-specificity of a term.

For all these reasons, we want to investigate these two term characteristics. More concretely, we define them as follows:

Specificity. Specificity describes the level of domain-specific information that a term carries, i.e. the level of **technicality** of a term. It is influenced by how strongly a term is represented in general language; a strongly specific term should only occur in its respective domain,

and not in general language. Symptomatically, specificity represents the level of **difficulty** or **familiarity** a lay person attributes to a term. I.e. how much expertise in the domain is necessary to understand a term.

Centrality. Centrality describes how strongly a term is prototypical or topically central to a domain. This concept is sometimes called subject-specificity and domain-specificity as well.

These two features can be perceived as forming a two-dimensional space, in which terms can be arranged. Example terms for the domain of soccer, arranged in the two-dimensional space, are given in figure 4.7.

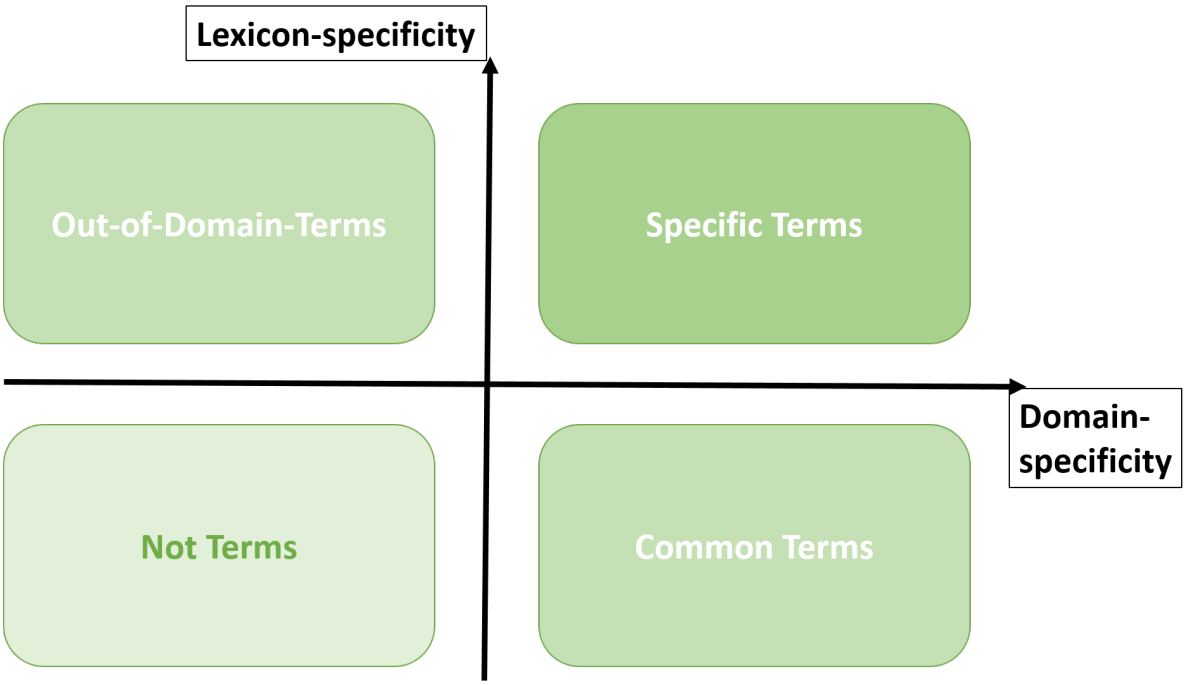


Figure 4.6.: Term model by Rigouts Terryn et al. (2018).

In parallel to the preparation of this thesis, a similar idea has been proposed by another group of researchers as well. Rigouts Terryn et al. (2018, 2019) define a two-dimensional space of similar two features: *Lexicon-specificity* denotes the degree to which a term belongs to either general language or to a lexicon of specialists, and *domain-specificity* shows how relevant the term is to the subject. They further define four categories in the two-dimensional space, as depicted in figure 4.6. They give examples from the medical domain for the three term categories: A specific term would be *ejection fraction*, a common term would be *heart*

and an out-of-domain-term would be *p-value*, since this is actually a term from statistics. The approach by Rigouts Terryn et al. (2018, 2019) is very similar to ours, in the way that a similar two-dimensional space is defined and that this constitutes an intuitive annotation framework for lay annotators. This confirms our intuition that centrality and specificity are fundamental characteristics of terms. The approach differs in the way that Rigouts Terryn et al. cut the space into four well-defined regions, with non-terms being located in the negative area. We do not define regions in space, and non-terms are set at the point of origin. In addition, Rigouts Terryn et al. perform a token-based annotation approach, while we perform a type-based approach. However, the similarities of our idea and the one by Rigouts Terryn et al. are remarkable. We believe that these two ideas came up in parallel shows even more that there was a need for this more fine-grained characterization of terms.

4.4.2. Specificity and Centrality: a User Study

In this section, we describe the annotation study in which we tested the previously described term characteristics. As a basis, we used a text collection consisting of 8,119 words for the cooking domain (they comprise cooking recipes), and a text collection comprising 21,410 words for the domain of DIY (4,908 words from forum texts and 16,502 words from expert-written DIY instructions). We got our term sample from a previous (not published) term annotation study that was conducted for that data set. For that previous annotation, term phrases should be identified in context and grouped into the following classes: *pattern, peripheral association, borrowed term, central concept of domain, specific and understandable term, specific and semi-understandable term, specific and non-understandable term*. Further, if a term candidate is ambiguous, it should be marked additionally. The token-based Fleiss' kappa agreement (Fleiss, 1971) was 0.81 for cooking, 0.67 for the DIY instructions and 0.73 for the DIY forum texts. Since the categories of this study already involved some kind of distinctions for centrality and specificity, we used these annotations as basis to get a rather balanced set of term candidates. We selected 200 items for each cooking and DIY. The samples were chosen by balancing over the term set according to the following criteria: Ambiguity (yes|no), complex term-constituent ratio (i.e. that some amount of terms have to be constituents of other selected terms), assigned class, agreement of annotators, and frequency of a term in text (frequency > 3). Then for each term, three context sentences were randomly drawn from the text basis and were given to the annotators as context for the target terms to annotate. 20 lay people were asked to annotate for specificity and centrality[9].The instructions asked to rate each term

[9]In the original annotation setting, we asked for labeling ambiguity as well, which was disregarded later on.

on a scale from 1 (low) to 6 (high) for the strength of centrality and specificity. Label 0 should be used if there was a problem and annotators could not rate a term. The annotators were given figure 4.7 as example, where we showed term examples from a different domain in order not to influence the annotators. We asked the annotators to rate centrality and specificity one after the other, not in parallel.

Figure 4.7.: Illustration of centrality and specificity for the domain of *soccer*.

Table 4.6 shows inter-annotator agreement, measured with the averaged Spearman's rank-order correlation coefficient ρ (Siegel and Castellan, 1988) for each annotator pair. The results show that annotators agree on specificity fairly well, but they do not agree on centrality. Further, annotators agree more on DIY than on cooking.

Centrality		**Specificity**	
DIY	Cooking	DIY	Cooking
0.32	0.26	0.66	0.54

Table 4.6.: Average Spearman's ρ correlations among annotators.

We further inspect the annotations for specificity and centrality in more detail. Boxplots for the annotations can be found in tables 4.8 to 4.11. The x-axis shows the 200 terms per task and domain, and the y-axis depicts the boxplots for the annotations on the scale from 1 to 6 (and 0 if the term could not be annotated). Annotations are sorted for the median.

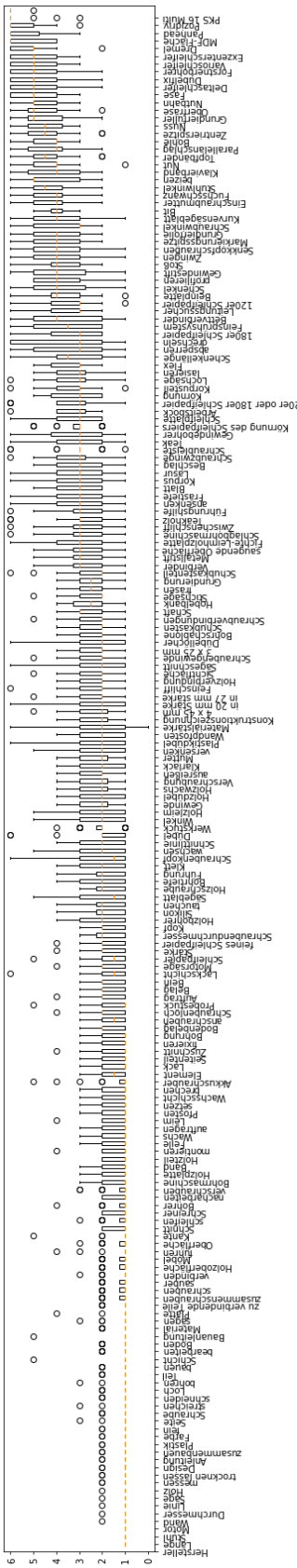


Figure 4.8.: Box plots for DIY specificity annotations.

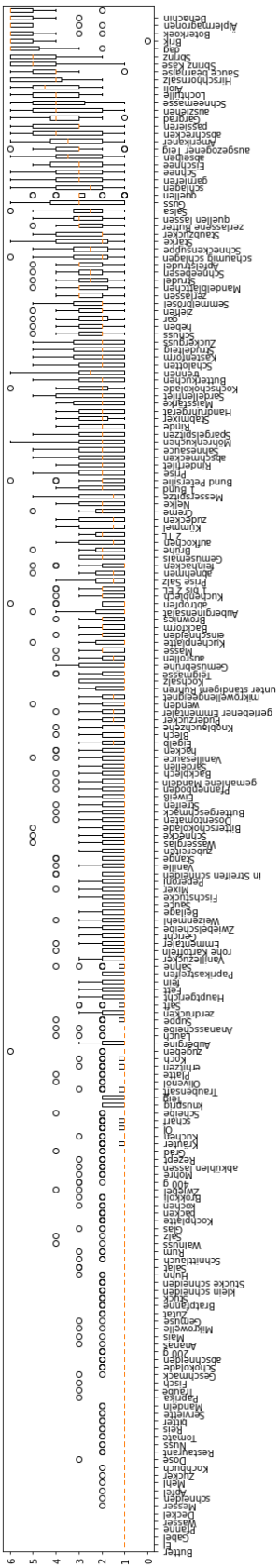


Figure 4.9.: Box plots for cooking specificity annotations.

Figure 4.11.: Box plots for cooking centrality annotations.

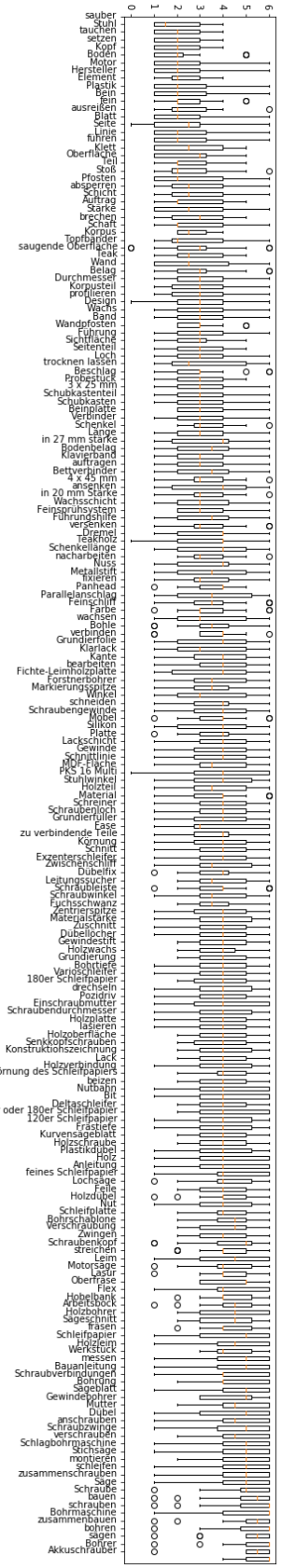


Figure 4.10.: Box plots for DIY centrality annotations.

The plots show that the divergences among annotators are high. However, both specificity and centrality annotations show an upward-moving tendency: if the median rises, the boxes (i.e. the range of distributions of the median 50% of the data) rise as well. This means that annotators have a broadly similar tendency for annotating, and that despite the divergences annotations of the attributes centrality and specificity are intuitive to some degree. While for centrality the divergence between annotators is high for all kinds of terms, we find that for specificity people especially agree well on non-specific terms. The reason for this might be that annotators can intuitively rely on how difficult they find a term candidate to understand when annotating specificity. They either completely understand a term and rate it with 1, or they find it to be at least partially difficult. For the latter case, it is harder to pin down the degree of difficulty. For example, one annotator could derive the meaning of the DIY term *Gewindebohrer* 'spiral fluted tap' by the constituents, while another annotator might deem it a difficult term because he has never seen the tool. Thus, although there are divergences for specificity as well, both the coherent annotations as well as divergences are rather intuitive to explain. In contrast to that, the reasons for the continuously high divergences for centrality are more complicated.

The problems for centrality could not be resolved in subsequent discussions. As a consequence of the discussions, we see a clash of different intuitions what constitutes centrality. One intuition could be that a term's broad usage in the domain could be an indicator for high centrality, as for example for *Rezept* 'recipe' for the domain of cooking. However, general and broad terms are then often used in other domains as well, or they are ambiguous. For example, *Rezept* can also be used in the context of pharmacy (engl.: 'prescription'). In these cases it is not clear which association wins, the high relevance within a domain or the lack of relevance in other domains[10]. Contrary to that, *zerlassen* 'melt, dissolve' has reverse attributes as *Rezept* in this respect: It is not highly used within the domain of cooking, but therefore it is predominantly used within it, e.g. in the phrase *Butter zerlassen* 'melt butter'[11]. Therefore, due to its nearly exclusive usage in the domain of cooking some annotators might see a high topical association to the domain, while others might see a low topical association due to its limited usage.

In sum, both specificity and centrality are rated with high divergences in this study, where we used scarce guidelines for annotation. However, annotations for specificity are better than

[10] In the process of refining the guidelines before undertaking the actual study, we even explicitly stated in the guidelines that one should concentrate on the domain-specific usage of a term candidate. Nevertheless, we still believe that the clash of intuitions remained.

[11] We searched the text collection we use as general-language corpus in this thesis, SdeWaC, for the word 'zerlassen', and we predominantly found it in the context of cooking.

for centrality on average, and the attribute is more intuitive, while for centrality there are partially clashing intuitions.

4.5. Summary

In this chapter, three annotation studies were described to analyze the nature of domain-specific terminology. Since manually identifying terminology is a difficult task, the first two annotation studies dealt with a first general exploration of how different groups of people annotate terms. Since we wanted to analyze the concept of terminology from scratch, the first study focused on lay annotators. The second study focused on semi-expert annotators. In the first study, we did not provide precise guidelines on what can be understood as term, lay annotators were asked for their intuitive notion of terminology instead. In the second study, we aimed at keeping a broad notion of terminology, but in addition, the study targeted at refining annotation guidelines in an iterative process, in order to improve inter-annotator agreement. Using this methodology, it could be evaluated which consistent divergences remained although effort was made to reduce divergences by rendering guidelines more precisely. Although the annotation procedures were different for both studies, they consonantly showed that it was highly difficult that people agreed on the linguistic form of a term, if dealing with multi-word expressions where the constituent words were separated by space. Especially the first study showed that annotators agreed to a higher extent on closed compounds. We conclude that for identifying closed compound terms, the advantages of both multi-word and single-word terms are combined: On the one hand, compounds come as one-word units and have clear delimitations, like single-word terms. On the other hand, single-word terms tend to be semantically more general than compound terms and other multi-word terms, which is why the latter kind of terms might be more easily identified as terms in this respect.

Besides problems with unclear delimitations of a term's linguistic form, regarding the understanding of terminology the lay people study showed that terminology is an intuitive concept. Even lay people exhibited some common understanding of terminology without being given precise guidelines, which shows that terminology is not a purely expert-defined construct. Furthermore, we found that annotators agreed more on certain terms than on others, even when being given different tasks for identifying terms. This stands in contrast to the conventional binary distinction into terms and non-terms, and would suggest a more fine-grained distinction of terms. In the course of the first two studies, we established the hypothesis that there are two main term characteristics that are responsible for terms being perceived as such to different extents: centrality, which represents the thematic relevance of a term candidate to a

domain, and specificity, which represents the degree of technicality. These two characteristics can be understood as forming a two-dimensional scale for describing terms. Our hypothesis was supported by a similar study coming up in parallel to ours Rigouts Terryn et al. (2018, 2019). We tested the robustness of the centrality-specificity framework by conducting a final annotation study with 20 lay test persons. We found that the notion of centrality is harder to pin down than the notion of specificity. Annotators agreed fairly well on specificity, and divergences were stronger for centrality. In addition, specificity was found to be the attribute for which divergences could be explained more easily, and thus it can be seen as the more intuitive attribute. We conclude that problems with term identification come to a higher degree from the thematic association of a term to a domain (centrality), than from the degree of technicality (specificity).

5. Automatic Term Extraction: Complex Terms, Meaning Variation

5.1. Introduction

The last chapter dealt with manually identifying terms while we shift now to automatically identifying terms, in other words, to automatic term extraction. This chapter covers experiments for "conventional" automatic term extraction, in the sense that we aim at distinguishing between terms and non-terms. The previous chapter's proposal for a more fine-grained term framework as a basis for automatic term extraction is dealt with in the next chapter. For automatic term extraction, we deal with two problems: a) identifying different forms of terms, i.e. complex and simple terms, and relying on the constituents of the first type, and b) meaning variation of term candidates.

Section 5.2 deals with recognizing **complex and simple terms**, and relying on complex terms' **constituents**. Exploiting additional information gained from nested terms or phrases is a traditional and popular approach for term extraction. Rather old models like Frantzi et al. (1998, 2000) and Nakagawa and Mori (2003) use nested terms and constituents of terms to more reliably identify terms. It is for this reason that we do a comparative study here: there is a vast amount of term extraction measures, and the first goal is to understand to what extent they add up onto each other and should be combined. That is why we collect and categorize term extraction measures. The second goal is to evaluate the influence of the constituents, by applying all measures to all possible constituent phrases, if applicable. Giving information to a classifier about which are the nested phrases *and* recursively computing the potential of the constituents to be terms with the same measures as for the term itself, is a new procedure to deal with embedded phrases. In both cases, the emphasis is given on the interpretability of the model, which is why we use decision trees as classifiers and carry out a detailed analysis of the outcomes of the classification.

Sections 5.3 to 5.5 then deal with **meaning variation** of terms: In section 5.3, we first demonstrate that the problem of meaning variation for term extraction has been underesti-

mated in the past, and how the meaning variation leads to a bias for the most relevant group of term extraction measures. Since no models exist for predicting and evaluating meaning variation in the area of automatic terminology extraction (with very few exceptions), we then rely on methodologies from the area of diachronic lexical semantic change. As a first step, we annotate a dataset (SURel - **S**ynchronic **U**sage **Rel**atedness) in parallel to an already existing dataset for lexical semantic change (DURel, Schlechtweg et al., 2018). Here, we map the problem setting and the annotation guidelines to our task. Section 5.4 covers a comparative study on computational models for predicting meaning variation, again mainly relying on models from the area of lexical semantic change, and one model from an area related to terminology extraction. Finally, we demonstrate the applicability of predicting meaning variation to term extraction: for section 5.5, the best model from section 5.4 is incorporated into a term extraction measure to improve automatic term extraction.

5.2. Modeling MWEs and their Constituents, and Simple Terms

This section deals with the recognition of multi-word (complex) terms and single-word (simple) terms[1]. To give examples, in the area of computational linguistics *parsing*, *machine translation* and *natural language generation* are candidates for single- and multi-word terms. By exploiting automatic term extraction (ATE) techniques, we identify such terms in domain-specific corpora. As described in the background section, a typical ATE system comprises two steps: First, term candidates are selected from text, e.g. by extracting sequences that match certain part-of-speech (POS) patterns in text. Secondly, term candidates are scored and ranked with regard to their unithood and termhood.

Unithood denotes to what degree a linguistic unit is a collocation. *Termhood* expresses to which extent an expression is a term, i.e to which extent it is related to domain-specific concepts (Kageura and Umino, 1996, for details, see section 3). Among a large number of measures, association measures like *Pointwise Mutual Information* (PMI) (Church and Hanks, 1990) are used to determine unithood whereas term-document measures like *tf-idf* (Salton and McGill, 1983) are used to determine termhood. Such measures use distinctive characteristics of terms on how they and their constituents are distributed within a domain or across domains.

We address term extraction as a machine learning classification problem (da Silva Conrado et al., 2013). Most importantly, we focus on the interpretability of a trained classifier

[1]The work in this section is published in Hätty et al. (2017a).

to understand the contributions of feature classes to the decision process. For this task, we use random forests to automatically detect the best features. These features are used to build simple decision tree classifiers.

For the classification, we use features based on numeric measures, which are computed from occurrences of term candidates, their constituents and derived symbolic information like POS tags. We call these *distributional features*. The advantage of relying on such features is that they are simple to compute and easy to compare. By combining machine learning with those features, we get a flexible system which only needs little further information to apply to different kinds of text. In this work, we investigate the contributions of the different features to term extraction and experimentally test with our system if these features are mutually supportive. We also point out the limit of a system solely relying on distributional features.

The section is organized as follows. Section 5.2.1 introduces related work. The data used for training and evaluation is presented in Section 5.2.2, followed by the feature selection and classification method. Our feature classes are motivated and defined in section 5.2.3. In section 5.2.4, we investigate the design of our models with a subsequent presentation of experiments and evaluation results in section 5.2.5. In section 5.2.6, we present a second experiment with term candidates that share a constituent to explore their contribution to termhood further.

5.2.1. Related Work

Several studies are investigating linguistic and numeric features, machine learning, or a combination of both to extract collocations or terms. Pecina and Schlesinger (2006) combined 82 association measures to extract Czech bigrams and tested various classifiers. The combination of measures was highly superior to using the best single measure. Ramisch et al. (2010) introduced the *mwetoolkit* which identifies multi-word expressions from different domains. The tool provides a candidate extraction step in advance, descriptive features (e.g. capitalization, prefixes) and association measures can be used to train a classifier. The latter ones are extended for multi-word expressions of indefinite length and only comprise measures that do not depend on a contingency table. Karan et al. (2012) extract bigram and trigram collocations for Croatian by relying on association measures, frequency counts, POS-tags and semantic similarities of all word pairs in an *n*-gram. They found that POS-tags, the semantic features and PMI work best. Concerning terms, Zhang et al. (2008) compare different measures (e.g. tf-idf) for both single- and multi-word term extraction and use a voting algorithm to predict the rank of a term. They emphasize the importance of considering unigram terms and the choice of the corpus. Foo and Merkel (2010) use RIPPER (Cohen, 1995), a rule induction learning system to extract unigram and bigram terms, by using both linguistic and numeric features.

They show that the design of the ratio of positive and negative examples while training governs the output rules. da Silva Conrado et al. (2013) investigate features for the classification of Brazilian Portuguese unigram terms. They use linguistic, statistical and hybrid features, where the context and the potential of a candidate representing a term are investigated. Regarding the features, they find tf-idf essential for all machine learning methods tested.

5.2.2. Data and Classification Method

Corpus and Gold Standard

ACL RD-TEC comprises the underlying corpus and the annotated term data set for the experiments (it is described in more detail in section 3). We take the whole set of *valid terms* as our gold standard term set. We clean the corpus by applying a language detection tool (*langdetect*[2]) to each sentence, in order to remove sentences which are too noisy. A drawback of the corpus is that about 42,000 sentences cannot be connected to a document. Thus, if no document is found for a particular term, its term-document measures are set to a default value outside of a feature's range, or to an extreme value.

Feature Reduction and Classification

Unigrams, bigrams and trigrams, which appear at least ten times in the text, are extracted from the corpus as term candidates. For all candidates, features are computed (see Section 5.2.3). As a preprocessing step, a **random forest classifier** (Breiman, 2001) with 100 estimators is used for feature reduction. To prevent overfitting, each of these decision trees is trained on a subset of the data, and a randomly chosen subset of features (here the square root of the number of features) is considered for splitting a node. Considering all internal decision trees, the contribution of the features to the classification is evaluated and averaged. In this way, we get good estimates of the importance of each feature and can use them for feature reduction: the classifier returns the importance scores for the features, and feature selection is performed by only taking those features whose score is greater than the mean. Subsequently, a **decision tree classifier** (Breiman et al., 1984) is trained with those features that provide a single representation for the decisions. The training set is balanced for terms and non-terms to prevent a bias in the classifier. In the first step, everything which is not marked as a term is treated as non-term. We only allow POS patterns also occurring in the term class and chose randomly to

[2]https://pypi.python.org/pypi/langdetect?

get a representative sample of non-terms. In the second step, we use the explicitly annotated non-term class.
Both classifiers produce binary decision trees and an optimized version of the CART algorithm[3] is used.
We use *information gain*[4] as split-criterium for the decision trees, and we only allow trees to evolve up to five levels, since they overfit otherwise. Besides, trees are very difficult to understand when getting deeper than five levels, and we explicitly choose decision trees because of their clear interpretability. For the following interpretation and evaluation, the construction of the final decision trees for each n-gram and their classification performances will be used.

5.2.3. Feature Classes

A salient attribute of terms is how they distribute in text. Our feature classes are motivated by three perspectives on that: a) measuring unithood involving the distribution of term candidates and their constituents, b) measuring termhood involving candidate term distributions in different texts, and c) recursively measuring unithood and termhood of term candidate constituents independently of each other. Concerning the classes defined in the following, point a) is covered by the *association measures*, b) by *term-document* and *contrastive measures* and c) by the *features of constituents*. In addition, we designed *count-based measures* and a *linguistic feature* to address unithood and termhood. However, we expect them to be weaker than the feature classes of a) and b) since they do not compare two kinds of distributions. They merely serve for filtering, ruling out doubtful term candidates.

Term-Document Measures (TD). The term-document measures deal with the distribution of term candidates in certain documents and contrast it to their distribution in the whole corpus. It is assumed that terms appear more frequently in only a few documents. We include a range of features dealing with that contrast: variants of tf-idf (Salton and McGill, 1983), i.e. *tf-idf* (without logarithm), *tf-logged-idf* for the document in which the term candidate occurs most often. Furthermore, *corpus maximum frequency* and *corpus maximum frequency & term average frequency (cmf-taf)* as defined in Tilley (2008), and *term variance* and *term variance quality* as described in Liu et al. (2005) are used. Da Silva Conrado et al. (2013) describe the latter features as useful for term extraction. In addition, we experimented with features

[3] `http://scikit-learn.org/stable/modules/tree.html#tree`
[4] Note that the parameter is called 'entropy' for the decision tree classifier of scikit-learn. Information gain is based on entropy, it is computed as the entropy of the parent node minus the weighted sum of entropy of the child nodes.

describing the relative occurrence of a term in a document or the corpus. For example, the percentiles of document or corpus frequencies are used as features, to which the frequency of the term under consideration can be assigned. Another example is the percentile of the document with the term candidate's first position. In the later experiments, these features are assigned little weight by the classifiers, which is why we will not go into further detail regarding them.

Contrastive Measures (CM). Contrastive measures treat the occurrence of a term in a general corpus and relate it to its occurrence in a domain-specific one. As domain-specific corpus, we simply chose the document with the most frequent occurrence of a term candidate. By doing that, the problem is omitted that the vocabulary of these corpora differs too drastically due to aspects of style. As features *weirdness ratio for domain specificity, corpora-comparing log-likelihood (corpComLL), term frequency inverse term frequency (TFITF)* and *contrastive selection of multi-word terms (CSmw)* are used (as defined in Schäfer et al., 2015).

Association Measures (AM). Association measures express how strongly words are associated in a complex expression, they measure unithood. 27 association measures defined in Evert (2005) were computed for bigrams, for example *Local Mutual Information (LocalMI)* and *Maximum Likelihood Estimation (MLE)*. For trigrams, we selected nine association measures (*MLE, PMI, Dice, T-score, Poisson-Stirling, Jaccard,* χ^2*, Simple Log Likelihood* and *true MI*) for which an extended usage for trigrams is described in Lyse and Andersen (2012), Ramisch et al. (2010) and the *nltk*-documentation [5].

Count-based Measures (Count). Wermter and Hahn (2006) compare co-occurrence frequencies and association measures and show that not association measures but only linguistically motivated features outperform frequency counts for collocation and terminology extraction. Therefore *frequencies* of the term candidates are included in the feature set. As described, we do not consider them as being as powerful as association measures (and they only play a minor role in our later models). The second count-based measure is *word length*.

Linguistic Feature (Ling). As a linguistic feature, *Part-Of-Speech*-tags (POS) of the candidates are used to represent distributions over POS patterns.

[5] www.nltk.org/_modules/nltk/metrics/association.html

Features of Constituents (Const). The constituents of a term phrase have frequently played a role in termhood extraction (Nakagawa and Mori, 2003; Zhang et al., 2012). Our approach differs from the previous ones by adding all feature information of the candidate term constituents to the candidates's feature set. I.e., for bigrams, the features of its unigrams, and for trigrams, the features of its uni- and bigrams are included. The features will be characterized with the following scheme: [POSITION IN TERM]-[CONSTITUENT IS A UNI- OR BIGRAM]-[FEATURE]. Examples would be *0-uni-CSmw* denoting the CSmw-feature for the first word X in bigram XY or *1-bi-CSmw* denoting the CSmw-feature for second bigram YZ in trigram XYZ. *1-bi-POS != NN NN* expresses that the second bigram YZ in trigram XYZ does not consist of nouns.

Class	1	2,3	Feature Examples
TD	+	+	tf-idf, cmf-taf, term variance
CM	+	+	weirdness ratio, corpComLL, TFITF
AM	-	+	PMI, LocalMI, Chi2
Count	+	+	frequency, word length
Ling	+	+	POS pattern
Const	-	+	0-uni-POS, 1-bi-tf-idf

Table 5.1.: Overview of Feature Classes

An overview of the classes is given in Table 5.1. The labels 1, 2 and 3 in the table denote uni- to trigrams, + and - express if a class can be applied or not. For unigram terms (SWT) not all feature classes can be applied.

5.2.4. Inspecting the Models

Combining all previously mentioned features with our classification method (i.e., unigrams, bigrams and trigrams) provides three decision trees. For ease of visualization and interpretation, only the first three decision levels are shown in the following figures (Figures 5.1 to 5.3). The tree is only allowed to evolve further if the distinction between terms and non-terms could not be made to that point. Furthermore, splitting a node is stopped if there are less than 10 elements in a leaf for one of the classes (even if the tree limit has not been reached yet).

Unigrams. The decision tree for unigram classification based on 1608 unigram terms and non-terms is shown in Figure 5.1. Term variance quality and term variance best classify terms;

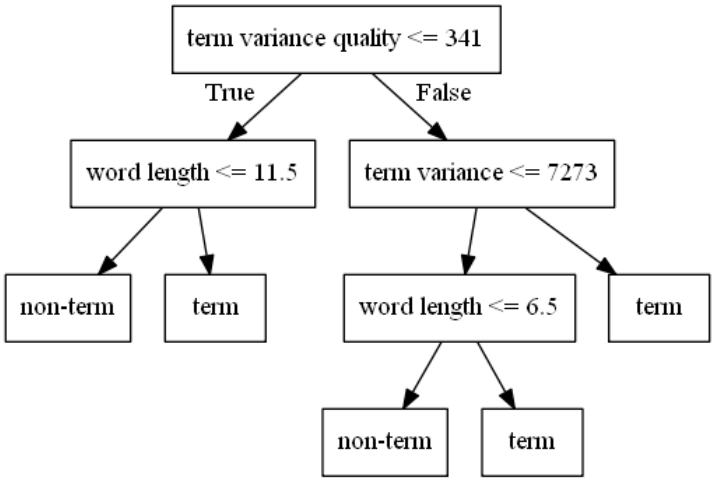


Figure 5.1.: Decision Tree for Unigrams.

In the resulting leaf node (rightmost node) 90% of the 324 elements are correct terms. When looking at the false positives in that node, it is striking that the few non-terms remaining in that class are unexpectedly "usual" (*'czech', 'newspaper', 'chain', 'travel', 'situation'*). The reason for this unexpected classification might result from the context in which the study is conducted: there might be papers that are limited to Czech data or only to newspaper texts.
The construction of the whole decision tree reveals that the classifier tries to identify clear-cut sets of terms using decision thresholds with extreme values. Following the path on the right-hand side, the subset of elements with the highest termhood scores is isolated. If the term-document measure values are not distinctive anymore (taking left branches), non-terms are singled out by filtering via word length. The less distinctive termhood measures are, the less word length is limited to filtering extremely short and therefore extremely unlikely term-candidates. This is an on-demand filtering step: term candidates are not only filtered in advance, but the threshold is adjusted to how significant the termhood measures are.

Bigrams. The decision tree for the 10,562 extracted bigram candidates is depicted in Figure 5.2. Features for the first constituent like 0-uni-CSmw are good indicators for termhood. When inspecting how the bigrams are distinguished by the root node, it seems that if the first word of a bigram is a general-language word, the whole bigram is unlikely to be a term. There are quite obvious examples like *this specification, the parser, a hurry* or *another expression* but also more interesting ones like *earlier paper, particular cluster* or *general scheme*. Nevertheless, in other term leaves there are still quite a few expressions whose first words are not

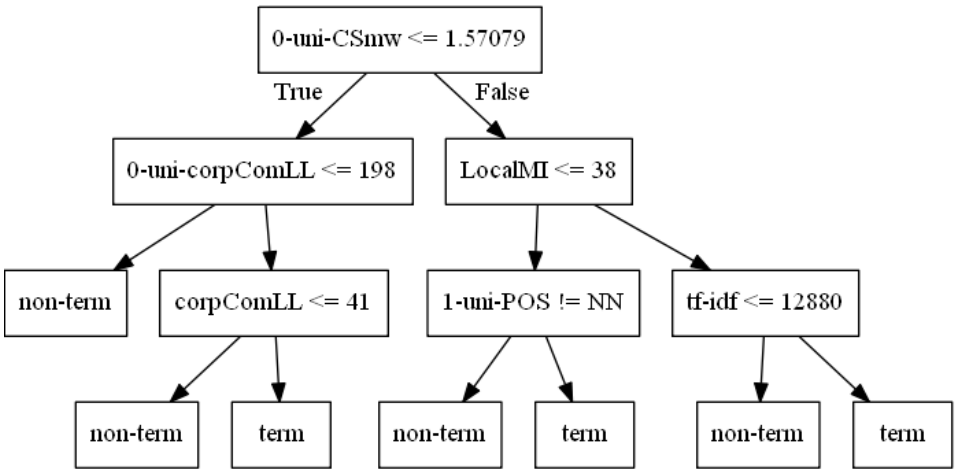


Figure 5.2.: Decision Tree for Bigrams.

terminological (e.g. *simple formalism, common description, good hypothesis*), so there is still room for improvement.

Trigrams. The decision tree for trigram classification of 1706 trigram candidates is shown in Figure 5.3. The association measure χ^2 (*Pearson's chi-squared test*; c.f. Evert, 2005) is by far the most important feature here and the sets are nearly completely distinguished by that feature. Thus, unithood nearly merges to termhood here. Besides that, it is again striking that expressions with non-terminological first constituents are ruled out correctly by the system, e.g., *possible syntactic category*, *other natural language*, *new grammar formalism*. There are also misclassifications (false negatives) like *first order logic*. The rightmost path produces the purest rightmost node compared to all previous ones for uni- and bigrams: 94% of the 636 elements are correct terms.

Comparison. Across the decision trees, different features dominate the tree, which shows that uni-, bi- and trigram terms behave differently and should be treated differently. Nevertheless, they have in common that the trees are dominated by termhood and unithood features and that features for filtering noise like POS patterns and word length occur lower in the tree. This supports the already mentioned claim that several filtering steps should be performed at different stages of the classification. As a second commonality, the trees combine features from various classes in their first decision steps. Especially in the rightmost path, in which terms are separated best in the experiments, term-document measures, association measures

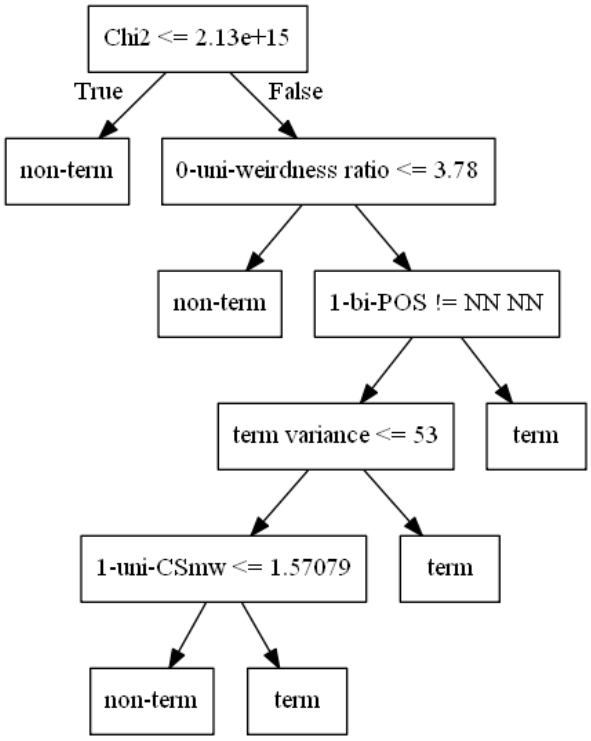


Figure 5.3.: Decision Tree for Trigrams.

and domain-specificity measures of constituents are combined. This shows that features from different feature classes interact for achieving a good result.

5.2.5. Experiments and Results

Our system is implemented in Python. For the classifications, we used the *RandomForestClassifier* and the *DecisionTreeClassifier* which are included in the Python module *sklearn* (Pedregosa et al., 2011).

Baselines. For each *n*-gram class, the best-working feature is chosen as a baseline. These are the root nodes of the decision trees for all features because they are chosen first, given that they make the best decision. The baselines are *term variance quality* for unigrams, *0-uni-CSmw* for bigrams and *Chi2* for trigrams.

Performance of Individual Feature Classes. As a first evaluation step, the different feature classes are compared. For that, decision trees are separately trained for each feature

class. We do 10-fold cross-validation with a balanced set of terms and non-terms in every step. The performances of the different classes for unigrams, bigrams and trigrams are shown in Table 5.2. When considering the overall results (F1-score), it is striking that for bigrams and trigrams the constituent features (Const) achieve the best score, middle-ranking groups are the count-based features (Count) and the linguistic feature (Ling), and the term-document (TD) and domain-specific features (CM) are in the lower area. This is quite a surprising result since these are the termhood features and therefore the ones to be expected to perform best. For unigrams, in contrast, term-document features and contrastive measures are good indicators for classification. However, when considering precision, the contrastive features lag behind. They do not seem to be competitive with term-document metrics in that respect. All in all, contrastive features do not reach the expected performance here. This is an interesting result because when the contrastive features are used for the constituents of an *n*-gram they appear in the upper part of the tree. We conclude that the contrastive features applied to constituents receive the unexpected application of downgrading the termhood of a term candidate if a constituent under consideration is unlikely to be terminological.

Feat. Class	**TD**	**CM**	**Assoc**	**Count**	**Ling**	**Const**
Unigrams						
Precision	**0.75**	0.67	-	0.73	0.63	-
Recall	0.71	0.73	-	0.66	0.81	-
F1-Score	**0.72**	0.70	-	0.69	0.70	-
Bigrams						
Precision	0.72	0.65	0.72	**0.73**	0.67	**0.73**
Recall	0.71	0.79	0.65	0.79	0.88	0.88
F1-Score	0.71	0.71	0.68	0.76	0.76	**0.80**
Trigrams						
Precision	0.67	0.59	0.85	0.75	0.80	**0.88**
Recall	0.72	0.72	0.96	0.82	1.0	0.97
F1-Score	0.69	0.65	0.90	0.78	0.89	**0.92**

Table 5.2.: Precision, Recall and F1-Scores for Feature Classes.

Evaluating All Features. As a last step, we evaluate if the combination of different features outperforms the best single feature. For that we do 10-fold cross-validation with a balanced set of terms and non-terms in every step. The results are shown in Table 5.3. All systems which combine features outperform the baselines. One can see that this is due to an increase in precision, while recall values are slightly lower than the baseline recall values. Besides, all systems also outperform the best systems which only use one feature class at a time (Table

5.2). All these improvements are significant[6], except for the comparison of the overall model for trigrams to the model of its best-working class (*features of constituents*). This shows that a combination is not only superior to a baseline but also information from several classes is needed. Term recognition works best for trigrams and is most difficult for unigrams.

Method	Precision	Recall	F-score
Baseline	0.62	0.85	0.70
Unigrams	0.75	0.79	**0.77**
Baseline	0.60	0.89	0.72
Bigrams	0.78	0.87	**0.81**
Baseline	0.84	0.97	0.90
Trigrams	0.89	0.96	**0.93**

Table 5.3.: Results.

5.2.6. The Relevance of the Constituent Class

In the previous experiments we investigated how terms can be distinguished from other expressions occurring in scientific texts which are restricted by POS but which are otherwise randomly chosen. For bigrams and trigrams, the constituent class performs best. Since the constituents of candidate terms seem to have a major influence on their termhood, we further investigate the constituents. For that, candidates are not chosen randomly anymore, but are taken from the class explicitly annotated as non-terms by Zadeh and Handschuh (2014). The reason for this is that the elements of the provided annotated term and non-term expressions have identical constituents in many cases. In this way, term candidates with constituents that are not uniquely terminological or non-terminological are used for training the classifier. Subsets of the classes are compared three times: Only those elements are allowed where either the first, the second or the third constituent (in case of trigrams) appears in both classes. The results are presented in Table 5.4.

The results indicate that a clearly terminological or non-terminological first constituent has more effect on the termhood of the whole expression than the last constituent has. If the first constituent is fixed and thus is not relevant for scoring termhood, results decrease.

This effect is also reflected in the decision trees: For identical heads, the most essential feature is the constituent feature of the first unigram and the first bigram. For identical modifiers,

[6] χ^2, $p<0.01$

	Bigrams			Trigrams		
Feature Class	**P**	**R**	**F1**	**P**	**R**	**F1**
last constituent	0.69	0.83	0.76	0.76	0.77	0.76
mid constituent	-	-	-	0.73	0.75	0.74
first constituent	0.66	0.70	0.68	0.73	0.71	0.72

Table 5.4.: Results for identical elements for different constituents in term- and non-term class.

no constituent feature is chosen as the most important feature.

5.2.7. Discussion

There are two main points why a system like ours only based on distributions reaches its limit. One aspect is the unexpected frequent occurrence of certain non-terms found especially for unigram term extraction. We found words being classified as terms because they often appear in the context of a particular experimental setting. Secondly, our results show that it is harder for such a system to distinguish term candidates with shared constituents than to distinguish terms from a representative part of the other in-domain text as done in the first experiment (Table 5.3 vs. Table 5.4).
However, an advantage of our model is that it is dynamic. Uni-, bi- and trigrams are quite different in nature which is reflected in the models. It filters improbable term candidates by making several decision steps adapted to the data seen in training. Thus, we might not need a preprocessing step to filter good candidates.

5.2.8. Conclusion

In this section, term extraction was approached as a classification problem using uni-, bi- and trigram term candidates. We used a decision tree classifier to model term recognition with focus on the distribution of terms and of their constituents in text. Different classifier setups were compared: classifiers for the single best feature, different feature classes and a combination of all features. In each of these steps classification improves. Neither a feature class nor a unique feature constantly dominates the classification in all models. The construction of the decision trees reveals that there is an interaction of features of different classes. Features from the most adequate classes to recognize terms, i.e. features which measure termhood and

unithood, interact to find the purest term class.
The decision trees resulting from the experiments indicate that there should not be a rigid pipeline of two steps, where candidate extraction and filtering noise comes first, and subsequently the terms are scored and ranked. Our results indicate that there should preferably be an on-demand filtering step, where filtering is performed successively during the classification and the threshold for ruling out improbable candidates is adjusted to the decisions made before.
The most exciting finding is that contrastive measures perform unexpectedly low for bigram and trigram recognition but when being applied to their unigram constituents they appear in the upper parts of the tree. When looking into the data, the reason for this seems to be that there is a downgrading of multi-word term candidate phrases (bigrams and trigrams) if a constituent (preferably the first) is too common to belong to a term. A second experiment, in which we compare term candidates with shared constituents confirms this finding. The constituents of terms are addressed in several studies (Erbs et al., 2015; Frantzi et al., 2000; Nakagawa and Mori, 2003; Zhang et al., 2012), but to our knowledge this aspect of termhood has not been considered yet.

5.3. A Dataset for Meaning Variation

When comparing domain-specific uses of terms to their usage in general language, we can find variations in meaning[7]. For example, the German noun *Schnee* predominantly means 'snow' in its general-language usage, and 'beaten egg whites' in the cooking domain. Terms with these characteristics are referred to as *sub-technical terms* and pose a problem for term extraction: Their nature makes it hard for humans to rank them along with unambiguous terms, and hard for computational models to classify them as terms, because of the strong bias towards their general-language meanings. In this section, we present SURel (**S**ynchronic **U**sage **R**elatedness), a novel dataset for meaning variation between general and domain-specific language, based on human annotations on the degrees of semantic relatedness between contexts of term candidates. We illustrate that SURel reflects the error that is commonly made by term extraction measures for sub-technical terms when relying on a general-language reference corpus.

[7]The work in this section is published in Hätty et al. (2019b).

5.3.1. The Relevance of Meaning Variation for ATE

The group of terms that is critical for automatic term extraction are sub-technical terms (background chapter 2) for which a variation in meaning between general and domain-specific language usages occurs. Despite the meaning variation, a sub-technical term can be as relevant for a particular domain as a monosemeous term. Their relevance must not be underestimated: Pérez (2016) provides empirical evidence that 50% of legal terminology is represented by sub-technical terms. Furthermore, the lay people study (4.2) shows that ambiguous terminology accounts for 4% of all word types in domain texts.

Sub-technical terms are a major problem for term extraction measures which usually operate on the word type rather than the word sense level.

One of the main strands of term extraction methodologies are contrastive techniques (described in 3.1.2), which compare a term candidate in a domain-specific and a general-language corpus. For these methods sub-technical terms are problematic, because irrelevant general-language senses are considered. An illustration is given in Figure 5.4. Contrastive term extraction measures are usually designed to identify terms with no meaning variation, in other words terms whose meaning in a domain-specific language is the same as its meaning in general language. We call this meaning *stability* in our illustration. Next to meaning stability, there are two relevant cases of ambiguity: First, when we consider all senses in general language, and then compare them with the sense(s) occurring in domain language, it can happen that not all senses still occur there. We call this a meaning *reduction*. This phenomenon is described by Baker (1988) (c.f. chapter 2.2.3) by giving the sub-technical terms example *effective*, which has the restricted meaning of 'to take effect' in biology. Secondly, when we consider general-language sense(s), and then compare them with the domain sense(s), it can happen that the meaning has completely changed. We call this a meaning *change*. Baker (1988) gives the example *bug* for this effect, whose meaning is completely different in computer science compared to general language. Senses that do not occur in the domain language should not be considered as term meanings; term hatchings in the figure show which meanings should not be considered as term meanings, and thus should not be considered by automatic term extraction methods.

However, with very few exceptions, sub-technical terms are not explicitly addressed by contrastive measures. Drouin (2004) mentions in his qualitative analysis that some ambiguous terms are not found by his extraction system. Menon and Mukundan (2010) and Pérez (2016) do explicitly tackle the extraction of sub-technical terms. Their systems rely on a term candidate's collocation frequencies in a domain and a general-reference corpus. But due to the lack of a gold standard, they only perform a qualitative analysis.

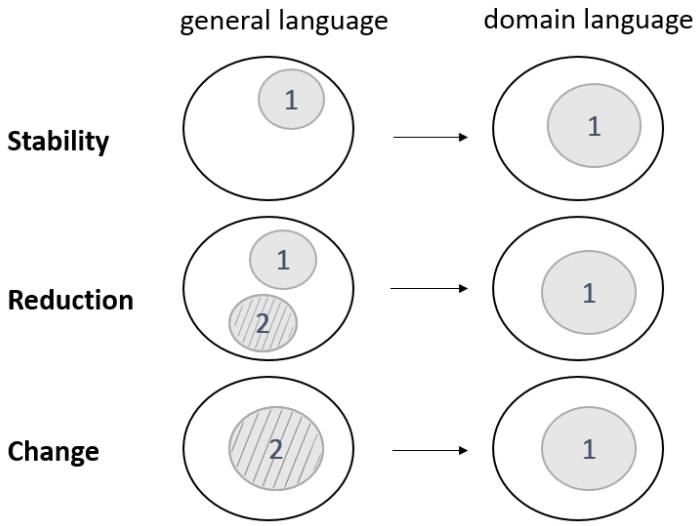


Figure 5.4.: Possible sense distributions of term candidates across languages.

This is where our work comes into play: sub-technical terms could be extracted in the same way as terms, if only the senses relevant for the domain were taken into account when comparing general language and domain-specific uses. Our novel dataset SURel captures meaning variation of term candidates and thus serves as a gold standard for the strength of the expected error produced by contrastive term extraction techniques when applied to sub-technical terms.

Finally, note that although we use expressions like *meaning reduction* or *meaning change*, we do not mean this in the sense of a temporal meaning shift. We do not make any assumptions if a general-language or a domain-specific sense was established first. However, disregarding the temporal aspect, procedurally the problem can be addressed in the same way as diachronic meaning change, as will be shown later on.

5.3.2. A Dataset for Meaning Variation: SURel

Relying on Strategies from Diachronic Lexical Semantic Change. No annotation scheme exists to annotate meaning variations of terms. However, the problem of diachronic lexical semantic change (LSC) is similar. In this area, meaning changes of words over time are analyzed. An example for diachronic LSC is the German noun *Vorwort* (Paul, 2002), which was mainly used in the meaning of 'preposition' before ≈1800. Then *Vorwort* rapidly acquired a new meaning 'preface', which after 1850 has nearly exclusively been used. When

the problem of diachronic semantic change is restricted on two time periods, the problem settings are nearly the same: For the diachronic task, meaning divergences from the first to the second time period are evaluated; for the terminology task (synchronic task), meaning divergences from general language to the domain language are evaluated. Schlechtweg et al. (2018) proposed an annotation scheme for LSC across two time periods, which we can then adapt to our problem. Thus, analogously to the annotated dataset DURel in Schlechtweg et al. (2018), we create our dataset SURel.

Corpus and Dataset Creation. We generated two corpora as a basis for the annotation in the following way:

- A **general-language corpus (GEN)**: we sub-sampled SDEWAC (Faaß and Eckart, 2013), a cleaned version of the web-crawled corpus DEWAC (Baroni et al., 2009). We reduced SdeWaC to ⅛th of its original size by selecting every 8th sentence for our general-language corpus. The reduced SdeWaC contains ≈126 million words.
- A **domain-specific corpus (SPEC)**: we crawled cooking-related texts from several categories (recipes, ingredients, cookware, cooking techniques) from the German cooking recipe websites *kochwiki.de* and *Wikibooks Kochbuch*[8]. SPEC contains ≈1.3 million words.

For the dataset to be manually annotated, we selected 22 target words which occurred in both GEN and SPEC, and which we expected to exhibit different degrees of meaning variation. For each target word we randomly sampled 20 use pairs, that means combinations of two context sentences from either GEN and SPEC, GEN and GEN, or SPEC and SPEC. Sampling the pairs from GEN, SPEC and across both, results in a total of 60 use pairs per word and 1,320 use pairs overall.

Annotation Strategy. For our novel dataset, annotators are asked to rate use pairs according to the relatedness scale given in 5.5 as illustrated in figure 5.5. It comprises a manual annotation of meaning relatedness between uses of target words in a general-language and a domain-specific corpus. The combined strength of relatedness between each pair of uses defines whether the meanings of a word are related or differ, thus indicating if a meaning variation exists and how strong the variation is.

Figure 5.5 shows the layout of the annotation. The use pairs consist of the sentences in column A on the left and column D on the right. The use pairs are either sampled both from

[8] de.wikibooks.org/wiki/Kochbuch

the general corpus (group GENMEAN), both from the domain corpus (group SPECMEAN), or one sentence comes from the general and the second sentence comes from the domain corpus (group COMPARE). Like that differences in meaning are measured across corpora, but also within the corpora.

4:	Identical
3:	Closely Related
2:	Distantly Related
1:	Unrelated
0:	Cannot decide

Table 5.5.: Relatedness scale, as given in Schlechtweg et al. (2018).

The annotator then rates the meaning relatedness of the use target word (bold) given in the use pairs, on the scale from 1 (unrelated meanings) to 4 (identical meanings). In figure 5.5, we give examples for each category: The two instances of *Ofen* in line 2 of figure 5.5 are judged identical in meaning (rating: 4), because both uses refer to an apparatus for heating. In contrast, the two uses of *reiben* in line 3 are judged closely related but not identical (rating: 3), because the meaning of *reiben* in target sentence 1 (column A in figure) is 'to rub', thus it is a more gentle kind of pressing and moving, while the meaning in target sentence 2 (column D in figure) is 'to comminute', thus also pressing and moving but with abrasion. In line 4, the two uses of the word *maskieren* are related, but more distantly (rating: 2): Both meanings are related by a figurative similarity, in the sense that they both denote a kind of covering or veiling. However, unlike *reiben* above, the two uses of *maskieren* in this example have different meanings. Target sentence 1 is about glazing a dish with a sauce, while target sentence 2 is about people masking themselves for carnival. A rating of 1 is assigned for two uses of a word that are completely unrelated in their meanings, as it is the case for *parieren* in line 5. Note that this pair of uses is semantically more distant than the two uses of *maskieren* above. The meaning in target sentence 1 is 'to obey', while the meaning in target sentence 2 denotes removing connective tissue from meat. Finally, there is also the option to provide the judgment 'Cannot decide' (rating: 0) when the annotator is unable to make a decision as to the degree of relatedness in meaning between the two bold words. One can provide comments in column C for why one cannot decide about this pair of uses.

After the annotation is finished, the annotation ratings for each word w can be computed to represent the degree of meaning variation. The most straightforward way is to compute a mean value for the annotation values for the sentence comparisons for GEN and SPEC.

	A	B	C	D
1	**Satz 1**	Bewertung	Kommentar	**Satz 2**
2	Aus dem **Ofen** nehmen und sofort mit Kristallzucker bestreuen oder die noch heißen Shortbreads in Kristallzucker wenden.	4		Hält man seine Hand in den **Ofen** fühlt sich die Luft im Backrohr warm an.
3	Er **rieb** seine schmerzende Wange.	3		Den Käse mit einer Reibe fein **reiben** und beiseite stellen.
4	Nun soviel von dem Gemisch aus Mayonnaise und Clotted Cream über die Eier geben, dass diese vollständig bedeckt (oder **"maskiert"**) sind.	2		Viele Künstler **maskieren** sich nicht nur zu Fasching, sondern das ganze Jahr über.
5	Wenn Rex dann nicht **parierte,** schlug man ihn.	1		Die Leber waschen, **parieren,** auf gewünschte Größe portionieren, zwei Stunden in Milch einlegen, damit die Leber ausbluten kann und gewisse Bitterstoffe entzogen werden.

Figure 5.5.: Layout of annotation, with examples for every kind of semantic relatedness.

$$\Delta\textsc{Compare}(w) = Mean_{\textsc{Compare}}(w)$$

This is the measure we will use for the gold standard creation.

Further, we have the corpus-internal comparisons for GEN and GEN, and SPEC and SPEC. One possibility is to use them to normalize ΔCOMPARE. For example, the degree of sense variation in one corpus (e.g. the general-language corpus) could be subtracted from ΔCOMPARE, in order to prevent high variability scores which only result from one corpus. The normalized compare measure, computed as $\Delta\textsc{Compare\_normalized}(w) = Mean_{\textsc{Compare}}(w) - Mean_{\textsc{GenMean}}(w)$, is a second measure proposed in Schlechtweg et al. (2018) (the formula is adapted to our use case). However, the measure only uses annotations for one of the corpora. Furthermore, if there is a high meaning variation within this corpus, the penalty factor might be too severe. For those reasons, we do not use the second measure. However, we still use the corpus-internal annotations for validation of our intuitions about meaning divergences.

Interannotator Agreement. Four native speakers annotated the use pairs on a scale from 1 (unrelated meanings) to 4 (identical meanings), reaching a strong mean pairwise agreement of ρ =0.88 (p <.01). The ranking of the 22 target words by their average strength of relatedness between general-language and domain-specific uses is shown in Figure 5.6. On the left are target words with highly related meanings in GEN and SPEC; on the right are words with strongly different meanings. Table 5.6 gives a detailed description of the dataset, with exemplary translations to illustrate the meaning variation.

lexeme	POS	translations	MRS	freq. GEN	freq. SPEC
Strudel	NN	whirlpool, strudel (a pastry)	1.05	232	46
Schnee	NN	snow, beaten egg whites	1.05	2,228	53
schlagen	VV	beat, whip (e.g. cream)	1.10	14,693	309
Gericht	NN	court, dish	1.15	13,263	1,071
Schuß	NN	shot (e.g. gunshot, shot of milk)	1.42	2,153	117
Hamburger	NN	citizen of Hamburg, hamburger	1.53	5,558	46
abschrecken	VV	discourage, chill (with cold water)	1.75	730	170
Form	NN	shape, (baking) mould	2.25	36,639	851
trennen	VV	separate	2.65	5771	170
Glas	NN	glass (e.g. material, drinking glass, jar)	2.70	3,830	863
Blech	NN	iron plate, baking tray	2.95	409	145
Prise	NN	pinch (e.g. of humour, tobacco, salt)	3.10	370	622
Paprika	NN	bell pepper, paprika	3.33	377	453
Messerspitze	NN	point of a knive, pinch (e.g. of salt)	3.43	39	49
Mandel	NN	tonsil, almond	3.45	402	274
Messer	NN	knife	3.50	1,774	925
Rum	NN	rum	3.55	244	181
Salz	NN	salt	3.74	3,087	5,806
Eiweiß	NN	protein, egg white	3.75	1,075	3,037
Schokolade	NN	chocolate	3.98	947	251
Schnittlauch	NN	chives	4.00	156	247
Gemüse	NN	vegetable	4.00	2,696	1,224

Table 5.6.: SURel dataset. MRS (mean relatedness score) denotes the compare rank as described in (Schlechtweg et al., 2018), where high values mean low change. The translations are illustrative for possible meaning variations, while further senses might exist.

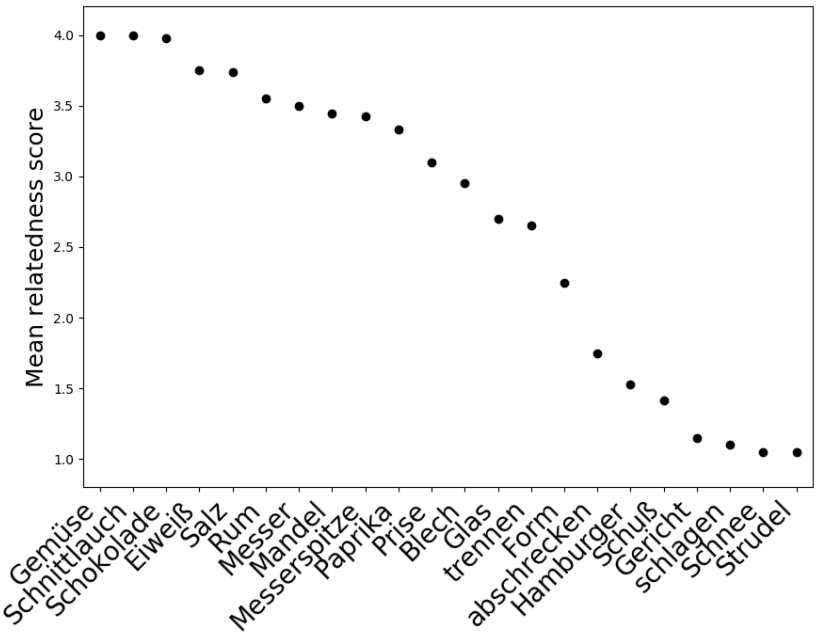


Figure 5.6.: Ranking of target words by average strength of meaning relatedness between GEN and SPEC.

Dataset Analysis. In the following, we analyse the meaning relatedness of use pairs within and across GEN and SPEC. Figure 5.7 shows examples of annotations that nicely correspond to cases of meaning *stability, reduction* and *change*, respectively. The y-axes shows how often the use pairs were rated as 1–4. The words are ordered for their final values and the values form a curve in the figure, covering the complete range of values. We had to select our target words based on intuition because there was no research available on the degree of meaning variation of German cooking terms (in comparison to the dataset of the original paper by Schlechtweg and Schulte im Walde, 2018). However, since the annotations are rather equally distributed, this confirms our intuition that we selected words in a way that the set is well-balanced by different degrees of meaning variation.

In Figure 5.7 on top we find *Schnittlauch* 'chive' with strongly related meanings within and across GEN and SPEC, thus indicating meaning stability. In the middle, we find *Messer* 'knife' with more related meanings in SPEC than in GEN, and even less strongly related meanings across GEN and SPEC, thus indicating meaning reduction. In Figure 5.7 at the bottom we find *Schnee* 'snow'/'beaten egg whites' with strongly related meanings within GEN and also within SPEC but very different meanings when comparing GEN and SPEC uses, thus indicating a complete meaning change. The three examples are taken from the two extremes and a mid

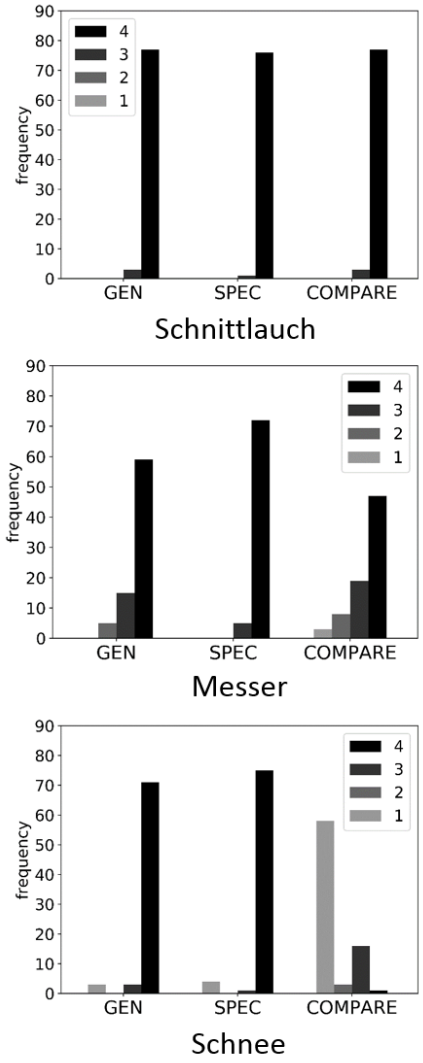


Figure 5.7.: Examples indicating meaning stability (top), meaning reduction (centre) and meaning change (bottom). COMPARE denotes cross-corpora relatedness (cf. Schlechtweg et al., 2018).

position in Figure 5.6. This shows that the annotation reflects our idea that complete meaning stability and a complete meaning change reflect the two extremes of the scale measuring the degree of meaning variation. A reduction or sense narrowing should be in between on the scale, depending on the strength of reduction. Note that a word can have several senses in each corpus, each of them occurring with different frequencies, which is reflected in the final meaning variation score.

5.4. Computational Modeling of Meaning Variation in a Comparative Study

Now that we have the gold standard dataset SURel for evaluation, computational models for predicting the strength of meaning variation will be evaluated and compared in this section[9]. As already described, not much work has been carried out for the prediction of meaning variation in the area of term extraction. Notably, only one model, later referred to as the *Word Injection* model (Ferrari et al., 2017), comes from the area of term extraction, but originally it had a slightly different focus and a stand-alone comparison of a few words with the highest meaning divergences was done. However, the area of diachronic lexical semantic change (LSC) detection, which deals with automatic detection of word sense changes over time, is a flourishing new field within NLP (Frermann and Lapata, 2016; Hamilton et al., 2016b; Schlechtweg et al., 2017, i.a.) and covers many different models. Yet, although many models have already been designed, the field lacks a systematic comparison of all the models and parameter settings because most models have been compared on different evaluation tasks and data sets up to now.

We thus kill two birds with one stone: With relying on DURel as a diachronic LSC gold standard dataset, we provide the first large-scale evaluation of a larger number of approaches for diachronic LSC detection. With relying on SURel as the meaning variation gold standard dataset, we can moreover test all the models on our terminology-related problem setting and can achieve a comparison of results for a different set of models, which have never been tested on that task before. On top, finding similarities in the results for both datasets strengthens the reliability of the results for each task.

To underline the similarity of the two tasks, from now on we will refer to the modeling of the meaning changes (or meaning variation) from general to domain language as **synchronic LSC detection**. Note that our case is actually just one example for synchronic LSC detection,

[9]The experiments in this section were joint work with Dominik Schlechtweg, Marco del Tredici and Sabine Schulte im Walde. The following descriptions are based on our publication Schlechtweg et al. (2019).

but the only one we deal with in the following study.

5.4.1. Related Work

Most of the studies, on whose methodologies we rely on for the synchronic and the diachronic LSC task, originate from the the field of diachronic LSC (Kim et al., 2014; Basile et al., 2015; Frermann and Lapata, 2016; Hamilton et al., 2016b; Schlechtweg et al., 2017). Some methodologies are used in other NLP areas as well, for example for bilingual lexicon induction (Artetxe et al., 2017, 2018a,b).

If we concentrate on research for the synchronic LSC task, the literal sense of this term would imply that we go into details for all kinds of synchronic lexical meaning variations, as for example meaning variation in online communities or for dialects (Del Tredici and Fernández, 2017; Hovy and Purschke, 2018, i.a.). However, as already described, we refer with this term to meaning variations from general to domain language, and since we regard such a study as an important step for terminology extraction, we go into details for research on ambiguous terminology. This area covers a broad range of meaning variances, not only meaning variation from general to domain language. In fact, it is mostly studied for within-domain ambiguities. A popular way to handle this problem is to apply domain-specific word sense disambiguation to all occurrences of a word within a domain (Maynard and Ananiadou, 1998; Chen and Al-Mubaid, 2006; Taghipour and Ng, 2015; Daille et al., 2016). This is a difficult task and returns a fine-grained in-domain ambiguity resolution. Another strand of research deals with term ambiguity detection, where a global prediction about a term's degree of ambiguity is made (Baldwin et al., 2013; Wang et al., 2013). Term ambiguity detection is methodologically closely related to our task, but also covers in-domain ambiguities. Ferrari et al. (2017) developed the only known term ambiguity detection approach that actually performs synchronic LSC detection, by measuring a term's meaning change across different domains. This method is referenced as *Word Injection* further on. It is based on a shared semantic vector space for different domains, but differentiating between vectors for target words. Cosine distance is used to measure meaning variation. For evaluation, the predictions for the top 100 most-frequent nouns in the corpora are analyzed. Note that this model is originally used to deal with meaning divergences across different domains, while we use it to measure meaning variation from general to domain-specific language.

Overall, we can see that detecting meaning variation either across domains or from general language to domain language is not intensively investigated. Thus, we provide the first systematic large-scale evaluation of an extensive number of existing and new modeling approaches for detecting meaning variation from general to domain language.

5.4.2. Task and Data

The experiments comprise two parallel studies:

- **synchronic LSC for specialized domains**: detecting a word's meaning variation between general language and a specific domain
- diachronic LSC: detecting a word's meaning change from one time period to another language

These tasks can be parallelized because on a meta-level two corpora are compared in each study (this is also the reason why such a study does not necessarily have to be limited to those two tasks). There is the following basic underlying task: given two corpora C_a and C_b, rank the target words in the datasets according to their degree of relatedness between the target word uses in C_a and C_b.

The focus for our application lies on finding the best model for predicting the synchronic LSC for specialized domains. There is a need to find the best-working method for predicting the meaning variations to integrate it into a term extraction procedure. Nevertheless, comparing to the diachronic case gives the results more robustness, especially since both underlying annotated datasets are rather small.

5.4.3. Corpora, Gold Standards and Evaluation Procedure

Corpora. For the synchronic LSC, we rely on the same datasets as for the SURel annotation (see subsection 5.3.2 for details on the processing of the corpora): the general-language corpus **GEN** and the domain-specific corpus **SPEC** (a corpus about cooking). For the diachronic task, we rely on DTA (Deutsches Textarchiv, 2017), which is a freely available lemmatized, POS-tagged and spelling-normalized diachronic corpus of German containing texts from the 16th to the 20th century. For all corpora, we removed words with a frequency threshold. For the smallest corpus we set the threshold to one, and set the other thresholds in the same proportion to the corpus size. We then created two versions:

- a version with minimal preprocessing, i.e. with punctuation removed and lemmatization (L_{ALL})
- a version limited on content words. After punctuation removal, lemmatization and POS-tagging, only nouns, verbs and adjectives are retained in the form lemma:POS (L/P).

Table 5.7 summarizes the corpus sizes after applying pre-processing.

	Domain		Time	
	GEN	**SPEC**	DTA18	DTA19
L_{ALL}	109M	1M	26M	40M
L/P	47M	0.6M	10M	16M

Table 5.7.: Corpora and their approximate sizes.

Gold Standard Annotated Datasets. For synchronic LSC, we rely on **SURel** (Hätty et al., 2019b), the newly created gold standard for synchronic LSC which was described in the last section (5.3). SURel is based on the same annotation framework as DURel (Schlechtweg et al., 2018), which we use as a basis for the diachronic task. DURel consists of 22 target words with varying degrees of LSC. Target words were chosen from a list of attested changes in a diachronic semantic dictionary (Paul, 2002), and for each target a random sample of use pairs from the DTA corpus was annotated for meaning relatedness of the uses on a scale from 1 (unrelated meanings) to 4 (identical meanings), both within and across the time periods 1750–1799 and 1850–1899. The annotation resulted in an average Spearman's $\rho = 0.66$ across five annotators and 1,320 use pairs. For our evaluation of diachronic meaning change we rely on the ranking of the target words according to their mean usage relatedness across the two time periods.

Evaluation. The gold LSC ranks in the SURel and DURel datasets are used to assess the correctness of model predictions by applying Spearman's rank-order correlation coefficient ρ as evaluation metric, as done in similar previous studies (Gulordava and Baroni, 2011; Schlechtweg et al., 2017; Schlechtweg and Schulte im Walde, 2018).

5.4.4. Meaning Representations

Our models are based on two families of distributional meaning representations: semantic vector spaces and topic distributions. All representations are bag-of-words-based, i.e. each word representation reflects a weighted bag of context words. The contexts of a target word w_i are the words surrounding it in an n-sized window: $w_{i-n}, ..., w_{i-1}, w_{i+1}, ..., w_{i+n}$.

Semantic Vector Spaces

A semantic vector space constructed from a corpus C with vocabulary V is a matrix M, where each row vector represents a word w in the vocabulary V reflecting its co-occurrence statistics

(Turney and Pantel, 2010). We compare two state-of-the-art approaches to learn these vectors from co-occurrence data, (i) counting and (ii) predicting, and construct vector spaces for each the general-language corpus and domain, and for each time period.

In a count-based semantic vector space the matrix M is high-dimensional and sparse. The value of each matrix cell $M_{i,j}$ represents the number of co-occurrences of the word w_i and the context c_j, $\#(w_i, c_j)$. In line with Hamilton et al. (2016b) we apply a number of transformations to these raw co-occurrence matrices, as previous work has shown that this improves results on different tasks (Bullinaria and Levy, 2012; Levy et al., 2015). In the following three kinds of count-based vector spaces are described (PPMI, SV and RI) and one predictive semantic vector space (SGNS).

Positive Pointwise Mutual Information (PPMI). In PPMI representations the co-occurrence counts in each matrix cell $M_{i,j}$ are weighted by the positive mutual information of target w_i and context c_j reflecting their degree of association. The values of the transformed matrix are

$$M_{i,j}^{\text{PPMI}} = \max\left\{\log\left(\frac{\#(w_i, c_j)\sum_c \#(c)^\alpha}{\#(w_i)\#(c_j)^\alpha}\right) - \log(k), 0\right\},$$

where $k > 1$ is a prior on the probability of observing an actual occurrence of (w_i, c_j) and $0 < \alpha < 1$ is a smoothing parameter reducing PPMI's bias towards rare words (Levy and Goldberg, 2014; Levy et al., 2015).

Singular Value Decomposition (SVD). Truncated SVD finds the optimal rank d factorization of matrix M with respect to L2 loss (Eckart and Young, 1936). We use truncated SVD to obtain low-dimensional approximations of the PPMI representations by factorizing M^{PPMI} into the product of the three matrices $U\Sigma V^\top$. We keep only the top d elements of Σ and obtain

$$M^{\text{SVD}} = U_d \Sigma_d^p,$$

where p is an eigenvalue weighting parameter Levy et al. (2015). The ith row of M^{SVD} corresponds to w_i's d-dimensional representation.

Random Indexing (RI). RI is a dimensionality reduction technique based on the Johnson-Lindenstrauss lemma according to which points in a vector space can be mapped into a randomly selected subspace under approximate preservation of the distances between points, if the subspace has a sufficiently high dimensionality (Johnson and Lindenstrauss, 1984;

Sahlgren, 2004). We reduce the dimensionality of a count-based matrix M by multiplying it with a random matrix R:

$$M^{\text{RI}} = MR^{|\mathcal{V}|\times d},$$

where the ith row of M^{RI} corresponds to w_i's d-dimensional semantic representation. The choice of the random vectors corresponding to the rows in R is important for RI. We follow previous work (Basile et al., 2015) and use sparse ternary random vectors with a small number s of randomly distributed -1s and $+1$s, all other elements set to 0, and we apply subsampling with a threshold t.

Skip-Gram with Negative Sampling (SGNS). SGNS is the only predictive vector space model we use, and falls within the category **word embeddings**, which we describe in the background section 3.3.3. This is the kind of vector space model we also use in the other experiments. SGNS differs from count-based techniques in that it directly represents each word $w \in V$ and each context $c \in V$ as a d-dimensional vector by implicitly factorizing $M = WC^\top$ when solving

$$\arg\max_\theta \sum_{(w,c)\in D} \log\sigma(v_c \cdot v_w) + \sum_{(w,c)\in D'} \log\sigma(-v_c \cdot v_w),$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$, D is the set of all observed word-context pairs and D' is the set of randomly generated negative samples (Mikolov et al., 2013a,b; Goldberg and Levy, 2014). The optimized parameters θ are $v_{c_i} = C_{i*}$ and $v_{w_i} = W_{i*}$ for $w, c \in V$, $i \in 1, ..., d$. D' is obtained by drawing k contexts from the empirical unigram distribution $P(c) = \frac{\#(c)}{|D|}$ for each observation of (w,c), cf. Levy et al. (2015). SGNS and PPMI representations are highly related in that the cells of the implicitly factorized matrix M are PPMI values shifted by the constant k (Levy and Goldberg, 2014). Hence, SGNS and PPMI share the hyper-parameter k. The final SGNS matrix is given by

$$M^{\text{SGNS}} = W,$$

where the ith row of M^{SGNS} corresponds to w_i's d-dimensional semantic representation. As in RI we apply subsampling with a threshold t. SGNS with particular parameter configurations has shown to outperform transformed count-based techniques on a variety of tasks (Baroni et al., 2014; Levy et al., 2015).

Alignment of Vector Spaces

We compute two separate vector spaces for each corpus pair, i.e. we compute one vector space for general language and one for the cooking domain. Vector spaces are not naturally aligned, which means that word vectors from different vector spaces cannot be directly compared. In the following, we explain the alignment measures we use.

Column Intersection (CI). In order to make the matrices A and B from general language a and domain b (or time periods $a < b$) comparable, they have to be aligned via a common coordinate axis. This is rather straightforward for count and PPMI representations, because their columns correspond to context words which often occur in both A and B (Hamilton et al., 2016b). In this case, the alignment for A and B is

$$A^{\text{CI}}_{*j} = A_{*j} \quad \text{for all } c_j \in V_a \cap V_b,$$
$$B^{\text{CI}}_{*j} = B_{*j} \quad \text{for all } c_j \in V_a \cap V_b,$$

where X_{*j} denotes the jth column of X.

Shared Random Vectors (SRV). RI offers an elegant way to align count-based vector spaces and reduce their dimensionality at the same time (Basile et al., 2015). Instead of multiplying count matrices A and B each by a separate random matrix R_A and R_B they may be multiplied both by the same random matrix R representing them in the same low-dimensional random space. Hence, A and B are aligned by

$$A^{\text{SVR}} = AR,$$
$$B^{\text{SVR}} = BR.$$

We follow Basile et al. and adopt a slight variation of this procedure: instead of multiplying both matrices by exactly the same random matrix (corresponding to an intersection of their columns) we first construct a shared random matrix and then multiply A and B by the respective sub-matrix.

Orthogonal Procrustes (OP). In the low-dimensional vector spaces produced by SVD, RI and SGNS the columns may represent different coordinate axes (orthogonal variants) and thus cannot directly be aligned to each other. Following Hamilton et al. (2016b) we apply OP analysis to solve this problem. We represent the dictionary as a binary matrix D, so that

$D_{i,j} = 1$ if $w_i \in V_b$ (the ith word in the vocabulary at domain b) corresponds to $w_j \in V_a$. The goal is then to find the optimal mapping matrix W^* such that the sum of squared Euclidean distances between B's mapping $B_{i*}W$ and A_{j*} for the dictionary entries $D_{i,j}$ is minimized:

$$W^* = \arg\min_W \sum_i \sum_j D_{i,j} \|B_{i*}W - A_{j*}\|^2.$$

Following standard practice we length-normalize and mean-center A and B in a pre-processing step (Artetxe et al., 2017), and constrain W to be orthogonal, which preserves distances within general language and within the domain, or within each time period. Under this constraint, minimizing the squared Euclidean distance becomes equivalent to maximizing the dot product when finding the optimal rotational alignment (Hamilton et al., 2016b; Artetxe et al., 2017). The optimal solution for this problem is then given by $W^* = UV^\top$, where $B^\top DA = U\Sigma V^\top$ is the SVD of $B^\top DA$. Hence, A and B are aligned by

$$A^{\text{OP}} = A,$$
$$B^{\text{OP}} = BW^*,$$

where A and B correspond to their preprocessed versions. We also experiment with two variants: OP$_-$ omits mean-centering (Hamilton et al., 2016b), which is potentially harmful as a better solution may be found after mean-centering. OP$_+$ corresponds to OP with additional pre- and post-processing steps and has been shown to improve performance in research on bilingual lexicon induction (Artetxe et al., 2018a,b). We apply all OP variants only to the low-dimensional matrices.

Vector Initialization (VI). In VI we first learn A^{VI} using standard SGNS and then initialize the SGNS model for learning B^{VI} on A^{VI} (Kim et al., 2014). The idea is that if a word is used in similar contexts in a and b, its vector will be updated only slightly, while more different contexts lead to a stronger update.

Word Injection (WI). Finally, we use the word injection approach by Ferrari et al. (2017) where target words are substituted by a placeholder in one corpus before learning semantic representations, and a single matrix M^{WI} is constructed for both corpora after mixing their sentences. The advantage of this approach is that all vector learning methods described above can be directly applied to the mixed corpus, and target vectors are constructed directly in the same space, so no post-hoc alignment is necessary.

Topic Distributions

Sense ChANge (SCAN). SCAN models LSC of word senses via smooth and gradual changes in associated topics (Frermann and Lapata, 2016). The semantic representation inferred for a target word w and time period t (and for our use case as well: general language and domain) consists of a K-dimensional distribution over word senses ϕ^t and a V-dimensional distribution over the vocabulary $\psi^{t,k}$ for each word sense k, where K is a predefined number of senses for target word w. SCAN places parametrized logistic normal priors on ϕ^t and $\psi^{t,k}$ in order to encourage a smooth change of parameters, where the extent of change is controlled through the precision parameter K^ϕ, which is learned during training.

Although $\psi^{t,k}$ may change over time for word sense k, senses are intended to remain thematically consistent as controlled by word precision parameter K^ψ. This allows comparison of the topic distribution across time periods. For each target word w we infer a SCAN model for general language and domain, or the two time periods a and b and take ϕ_w^a and ϕ_w^b as the respective semantic representations.

5.4.5. LSC Detection Measures

LSC detection measures predict a degree of LSC from two semantic representations of a word w. They either capture the contextual similarity or changes in the contextual dispersion of w's representations.[10]

Similarity Measures

Cosine Distance (CD). CD is based on cosine similarity which measures the cosine of the angle between two non-zero vectors $\vec{x}, \vec{y}$ with equal magnitudes (Salton and McGill, 1983):

$$cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\sqrt{\vec{x} \cdot \vec{x}}\sqrt{\vec{y} \cdot \vec{y}}}.$$

The cosine distance is then defined as

$$CD(\vec{x}, \vec{y}) = 1 - cos(\vec{x}, \vec{y}).$$

CD's prediction for a degree of LSC of w between time periods a and b is obtained by $CD(\vec{w}_a, \vec{w}_b)$.

[10] Find an overview of which measure was applied to which representation type in Appendix 5.4.6.

Local Neighborhood Distance (LND). LND computes a second-order similarity for two non-zero vectors $\vec{x}, \vec{y}$ (Hamilton et al., 2016a). It measures the extent to which $\vec{x}$ and $\vec{y}$'s distances to their shared nearest neighbors differ. First the cosine similarity of $\vec{x}, \vec{y}$ with each vector in the union of the sets of their k nearest neighbors $N_k(\vec{x})$ and $N_k(\vec{y})$ is computed and represented as a vector s whose entries are given by

$$s(j) = \cos(\vec{x}, \vec{z_j}) \quad \forall \vec{z_j} \in N_k(\vec{x}) \cup N_k(\vec{y}).$$

LND is then computed as cosine distance between the two vectors:

$$LND(\vec{x}, \vec{y}) = CD(\vec{s_x}, \vec{s_y}).$$

LND does not require matrix alignment, because it measures the distances to the nearest neighbors in each space separately. It was claimed to capture changes in paradigmatic rather than syntagmatic relations between words (Hamilton et al., 2016a).

Jensen-Shannon Distance (JSD). JSD computes the distance between two probability distributions ϕ_x, ϕ_y of words w_x, w_y (Lin, 1991; Donoso and Sanchez, 2017). It is the symmetrized square root of the Kullback-Leibler divergence:

$$JSD(\phi_x||\phi_y) = \sqrt{\frac{D_{KL}(\phi_x||M) + D_{KL}(\phi_y||M)}{2}},$$

where $M = (\phi_x + \phi_y)/2$. JSD is high if ϕ_x and ϕ_y assign different probabilities to the same events.

Dispersion Measures

Frequency Difference (FD). The log-transformed relative frequency of a word w for a corpus C is defined by

$$F(w, C) = \log \frac{|w \in C|}{|C|}$$

FD of two words x and y in two corpora X and Y is then defined by the absolute difference in F:

$$FD(x, X, y, Y) = |F(x, X) - F(y, Y)|$$

FD's prediction for w's degree of LSC between general language and domain or time periods a and b with corpora C_a and C_b is computed as $FD(w, C_a, w, C_b)$ (parallel below).

Type Difference (TD). TD is similar to FD, but based on word vectors $\vec{w}$ for words w. The normalized log-transformed number of context types of a vector $\vec{w}$ in corpus C is defined by

$$T(\vec{w}, C) = \log \frac{\sum_{i=1} 1 \quad \text{if } \vec{w}_i \neq 0}{|C_T|},$$

where $|C_T|$ is the number of types in corpus C. The TD of two vectors $\vec{x}$ and $\vec{y}$ in two corpora X and Y is the absolute difference in T:

$$TD(\vec{x}, X, \vec{y}, Y) = |T(\vec{x}, X) - T(\vec{y}, Y)|.$$

Entropy Difference (HD). HD relies on vector entropy as suggested by Santus et al. (2014). The entropy of a non-zero word vector $\vec{w}$ is defined by

$$VH(\vec{w}) = -\sum_{i=1} \frac{\vec{w}_i}{\sum_{j=1} \vec{w}_j} \log \frac{\vec{w}_i}{\sum_{j=1} \vec{w}_j}.$$

VH is based on Shannon's entropy (Shannon, 1948), which measures the unpredictability of w's co-occurrences (Schlechtweg et al., 2017). HD is defined as

$$HD(\vec{x}, \vec{y}) = |VH(\vec{x}) - VH(\vec{y})|.$$

We also experiment with differences in H between topic distributions ϕ^a_w, ϕ^b_w, which are computed in a similar fashion, and with normalizing VH by dividing it by $\log(VT(\vec{w}))$, its maximum value.

5.4.6. Pre-processing and Hyperparameter Details

Corpora. For all corpora, we removed words below a frequency threshold t. For the smallest corpus SPEC we set $t = 2$, and set the other thresholds in the same proportion to the corpus size. This led to $t = 25, 37, 97$ for DTA18, DTA19 and GEN respectively. (Note that we excluded three targets from the DURel dataset and one target from the SURel dataset because they were below the frequency threshold.) We then created the two previously described versions L_{ALL} and L/P. We used the TCF-version of DTA released September 1, 2017.[11]

Context Window. For all models we experimented with values $n = \{2, 5, 10\}$ as done in Levy et al. (2015). It is important to note that the extraction of context words differed

[11] http://www.deutschestextarchiv.de/download

between models, because of inherent parameter settings of the implementations. While our implementations of the count-based vectors have a stable window of size n, SGNS has a dynamic context window with maximal size n (cf. Levy et al., 2015) and SCAN has as stable window of size n, but ignores all occurrences of a target word where the number of context words on either side is smaller than n. This may affect the comparability of the different models, as especially the mechanism of SCAN can lead to very sparse representations on corpora with small sentence sizes, as e.g. the SPEC corpus. Hence, this variable should be controlled in future experiments.

Vector Spaces. We followed previous work in setting further hyper-parameters (Hamilton et al., 2016b; Levy et al., 2015). We set the number of dimensions d for SVD, RI and SGNS to 300. We trained all SGNS with 5 epochs. For PPMI we set $\alpha = .75$ and experimented with $k = \{1, 5\}$ for PPMI and SGNS. For RI and SGNS we experimented with $t = \{none, .001\}$. For SVD we set $p = 0$. In line with Basile et al. (2015) we set $s = 2$ for RI and SRV. Note though that we had a lower d than Basile et al., who set $d = 500$.

SCAN. We experimented with $K = \{4, 8\}$. For further parameters we followed the settings chosen by Frermann and Lapata (2016): $K^{\psi} = 10$ (a high value forcing senses to remain thematically consistent across time). We set K^{ϕ} = 4, and the Gamma parameters $a = 7$ and $b = 3$. We used $1,000$ iterations for the Gibbs sampler and set the minimum amount of contexts for a target word per time period $min = 0$ and the maximum amount to $max = 2000$.

Measures. For LND we set $k = 25$ as recommended by Hamilton et al. (2016a). The normalization constants for FD, HD and TD were calculated on the full corpus with the respective preprocessing (without deleting words below a frequency threshold).

5.4.7. Model Overview

Find an overview of all tested combinations of semantic representations, alignments and measures in Table 5.8.

5.4.8. Results and Discussions

First of all, we observe that nearly all model predictions have a strong positive correlation with the gold rank. Table 5.9 presents the overall best results across models and parameters. Note that for models with randomness we computed the average results of five iterations. With

Semantic Representation	Alignment					Measure					
	CI	SRV	OP	VI	WI	CD	LND	JSD	FD	TD	HD
raw count	x				x	x	x			x	x
PPMI	x				x	x	x				
SVD			x		x	x	x				
RI		x	x		x	x	x				
SGNS			x	x	x	x	x				
SCAN								x			(x)

Table 5.8.: Combinations of semantic representation, alignment types and measures. (FD has been computed directly from the corpus.)

Dataset	Preproc	Win	Space	Parameters	Align	Measure	Spearman m (h, l)
SURel	L/P	2	SGNS	k=1,t=0.001	OP	CD	**0.851** (0.851, 0.851)
	L/P	2	SGNS	k=5,t=None	OP	CD	0.850 (0.850, 0.850)
	L/P	2	SGNS	k=5,t=0.001	OP	CD	0.834 (0.838, 0.828)
	L/P	2	SGNS	k=5,t=0.001	OP_	CD	0.831 (0.836, 0.817)
	L/P	5	SGNS	k=5,t=0.001	OP	CD	0.829 (0.832, 0.823)
DURel	L_{ALL}	10	SGNS	k=1,t=None	OP	CD	**0.866** (0.914, 0.816)

Table 5.9.: Best results of ρ scores (Win=Window Size, Preproc=Preprocessing, Align=Alignment, k=negative sampling, t=subsampling, Spearman m(h,l): mean, highest and lowest results).

$\rho = 0.85$ **for synchronic LSC (SURel)** the model reaches an unexpectedly high performance. The even better $\rho = 0.87$ for diachronic LSC (DURel) shows that results are comparable on the two distinct datasets, which makes our results more robust. The overall best-performing model is Skip-Gram with orthogonal alignment and cosine distance (SGNS+OP+CD). The model is robust in that it performs best on both datasets and produces very similar, sometimes the same results across different iterations. It is notable that the results for SURel can compete with the results for DURel; the difference between general and domain language should be more intense than between two time periods, because one can assume that many more context changes due to the topical restriction of the domain. It might be possible that the global optimization process that is underlying the OP alignment might compensate for this. Furthermore, in next chapter's section 6.4 we find that the general-language corpus contains some domain-specific content as well, which might enforce this effect.

Pre-processing and Parameters. Regarding pre-processing, the results are less consistent for the two datasets: L/P (lemma:pos of content words) dominates in the synchronic task on SURel, while L_{ALL} (all lemmas) dominates in the diachronic task. In addition, L/P pre-processing, which is already limited on content words, prefers shorter windows, while L_{ALL} (pre-processing where the complete sentence structure is maintained) prefers longer windows. Regarding the preference of L/P for SURel, we blame unexpected text structures in the SPEC corpus, which contains many recipes listing ingredients and quantities with numerals and abbreviations, to presumably contribute little information about context words. For instance, SPEC contains 4.6% numerals, while DTA only contains 1.2% numerals. A second interesting result is that for SURel short windows are preferred, while for DURel longer windows lead to best results. As a starting point for interpretation why the synchronic task prefers short window sizes (besides the just mentioned fact that it therefore restricts on context words, i.e. L/P), we rely on insights from studies investigating the impact of window size. Lison and Kutuzov (2017) and Goldberg (2016) both find that shorter windows better model functional or synonym-like relationships, which lead to better results on similarity tests; longer windows better model topical relationships, which lead to better results on analogy tests. These insights might already answer the question. Modeling a topical association of the words would inevitably spawn a cooking association of the word vectors based on the cooking corpus, irrespective of their real association to cooking, due to the topical focus of the cooking corpus. This can be illustrated with an example from the cooking corpus: the word *Schnittlauch* occurs very often in the direct context of the words *Petersilie* or *frisch* in the cooking corpus. Additionally, it sometimes occurs within a longer range of ingredient listings: .. *Beifuß*

Bohnenkraut Dill Koriander Kresse alle Art Liebstöckel Löwenzahn Majoran Oregano Petersilie Schnittlauch Thai-Basilikum bis zum Thymian. These ingredients listings make contexts more diverse and can be seen as adding some noise to the vectors (because of driving them away from the clearer similarity to *Petersilie*), but also constitute the topical association to cooking.

Looking at the influence of subsampling, we find that it does not improve the mean performance for Skip-Gram (SGNS) (with $\rho = 0.506$, without $\rho = 0.517$), but clearly for Random Indexing (RI) (with $\rho = 0.413$, without $\rho = 0.285$). Levy et al. (2015) found that SGNS prefers numerous negative samples ($k > 1$), which is confirmed here: mean ρ with $k = 1$ is 0.487, and mean ρ with $k = 5$ is 0.535.[12] This finding is also indicated in Table 5.9, where $k = 5$ dominates the 5 best results on both datasets; yet, $k = 1$ provides the overall best result on both datasets.

Semantic Representations. Table 5.10 shows the best and mean results for different semantic representations. SGNS is clearly the best vector space model, even though its mean performance does not outperform other representations as clearly as its best performance. Regarding count models, PPMI and SVD show the best results.

SCAN performs poorly, and its mean results indicate that it is rather unstable. This may be explained by the particular way in which SCAN constructs context windows: it ignores asymmetric windows, thus reducing the number of training instances considerably, in particular for large window sizes. SCAN performs especially bad on SURel, which might be due to the underlying topic model. While topics might be equally diverse in two time corpora, for SURel diverse topics will be addressed as well in GEN, but in SPEC topics are restricted to domain-related topics. SCAN was not designed for such a topic reduction.

Alignments. The fact that our modification of Hamilton et al. (2016b) (SGNS+OP) performs best across datasets confirms our assumption that column-mean centering is an important preprocessing step for Orthogonal Procrustes analysis and should not be omitted.

Additionally, the mean performance in Table 5.11 shows that OP is generally more robust than its variants. OP_+ performs poorly on SURel (but well on DURel). Artetxe et al. (2018a) show that the additional pre- and post-processing steps of OP_+ can be harmful in certain conditions. We tested the influence of the different steps and identified the non-orthogonal whitening transformation as the main reason for a performance drop of ≈20% for SURel.

In order to see how important the alignment step is for the low-dimensional embeddings

[12]For PPMI we observe the opposite preference, mean ρ with $k = 1$ is 0.549 and mean ρ with $k = 5$ is 0.439.

Dataset	Representation	best	mean
SURel	raw count	0.599	0.120
	PPMI	0.791	0.500
	SVD	0.639	0.300
	RI	0.622	0.299
	SGNS	**0.851**	**0.520**
	SCAN	0.082	-0.244
DURel	raw count	0.639	0.395
	PPMI	0.670	0.489
	SVD	0.728	0.498
	RI	0.601	0.374
	SGNS	**0.866**	**0.502**
	SCAN	0.327	0.156

Table 5.10.: Best and mean ρ scores across similarity measures (CD, LND, JSD) on semantic representations.

(SVD/RI/SGNS), we also tested the performance without alignment ('None' in Table 5.11). As expected, the mean performance drops considerably. However, it remains positive, which suggests that the spaces learned in the models are not random but rather slightly rotated variants.

Especially interesting is the comparison of Word Injection (WI) where one common vector space is learned against the OP-models where two separately learned vector spaces are aligned. Although WI avoids (post-hoc) alignment altogether, it is consistently outperformed by OP, which is shown in Table 5.11 for low-dimensional embeddings.[13] We found that OP profits from mean-centering in the pre-processing step: applying mean-centering to WI matrices improves the performance by 3% on WI+SGNS+CD.

The results for Vector Initialization (VI) are unexpectedly low for both datasets, with a mean on SURel of $\rho = 0.082$ (on DURel: mean $\rho = -0.017$). An essential parameter choice for VI is the number of training epochs for the initialized model. We experimented with 20 epochs instead of 5, but could not improve the performance. This contradicts the results obtained by Hamilton et al. (2016b) who report a "negligible" impact of VI when compared to OP$_-$. We reckon that VI is strongly influenced by frequency. That is, the more frequent a word is in corpus C_b, the more its vector will be updated after initialization on C_a. Hence, VI predicts more change with higher frequency in C_b.

[13] We see the same tendency for WI against random indexing with a shared random space (SRV), but instead variable results for count and PPMI alignment (CI). This contradicts the findings in Dubossarsky et al. (2019), using, however, a different task and synthetic data.

Dataset	OP	OP_-	OP_+	WI	None
SURel	**0.590**	0.514	0.401	0.492	0.285
DURel	0.618	0.557	**0.621**	0.468	0.254

Table 5.11.: Mean ρ scores for CD across the alignments. Applies only to RI, SVD and SGNS.

Detection Measures. Cosine distance (CD) dominates Local Neighborhood Distance (LND) on all vector space and alignment types and hence should be generally preferred if alignment is possible. Otherwise LND or a variant of WI+CD should be used, as they show lower but robust results.[14] Dispersion measures in general exhibit a low performance, and previous positive results for them could not be reproduced (Schlechtweg et al., 2017). It is striking that, contrary to our expectation, dispersion measures on SURel show a strong negative correlation (max. $\rho = -0.79$). We suggest that this is due to frequency particularities of the dataset: SURel's gold LSC rank has a rather strong negative correlation with the targets' frequency rank in the SPEC corpus ($\rho = -0.51$). Moreover, because SPEC is magnitudes smaller than GEN the normalized values computed in most dispersion measures in SPEC are much higher. This gives them also a much higher weight in the final calculation of the absolute differences. Hence, the negative correlation in SPEC propagates to the final results. This is supported by the fact that the only measure not normalized by corpus size (HD) has a positive correlation. As these findings show, the dispersion measures are strongly influenced by frequency and very sensitive to different corpus sizes.

Control Condition. As we saw, dispersion measures are sensitive to frequency. Similar observations have been made for other LSC measures (Dubossarsky et al., 2017). In order to test for this influence within our datasets we follow Dubossarsky et al. (2017) in adding a control condition to the experiments for which sentences are randomly shuffled across corpora (since this was originally only done for time periods, we call it "time-shuffling" here). For each target word we merge all sentences from the two corpora C_a and C_b containing it, shuffle them, split them again into two sets while holding their frequencies from the original corpora approximately stable and merge them again with the original corpora. This reduces the target words' mean degree of LSC between C_a and C_b significantly. Accordingly, the mean degree of LSC predicted by the models should reduce significantly if the models measure LSC (and not some other controlled property of the dataset such as frequency). We find that the mean

[14] JSD was not included here, as it was only applied to SCAN and its performance thus strongly depends on the underlying meaning representation.

Dataset	ORG	SHF	+DWN
SURel	**0.767**	0.763	0.576
DURel	**0.816**	0.180	0.372

Table 5.12.: ρ for SGNS+OP+CD (L/P, win=2, k=1, t=None) before (ORG) and after time-shuffling (SHF) and downsampling them to the same frequency (+DWN).

prediction on a result sample (L/P, win=2) indeed reduces from from 0.53 to 0.44 on SURel (and from 0.5 to 0.36 on DURel). Moreover, shuffling should reduce the correlation of individual model predictions with the gold rank, as many items in the gold rank have a high degree of LSC, supposedly being canceled out by the shuffling and hence randomizing the ranking. Testing this on a result sample (SGNS+OP+CD, L/P, win=2, k=1, t=None), as shown in Table 5.12, we find that it holds for DURel with a drop from $\rho = 0.816$ (ORG) to 0.180 on the shuffled (SHF) corpora, but not for SURel where the correlation remains stable (0.767 vs. 0.763). We hypothesize that the latter may be due to SURel's frequency properties and find that downsampling all target words to approximately the same frequency in both corpora (≈ 50) reduces the correlation (+DWN). However, there is still a rather high correlation left (0.576). Presumably, other factors play a role: (i) Time-shuffling may not totally randomize the rankings because words with a high change still end up having slightly different meaning distributions in the two corpora than words with no change at all. Combined with the fact that the SURel rank is less uniformly distributed than DURel this may lead to a rough preservation of the SURel rank after shuffling. (ii) For words with a strong change the shuffling creates two equally ambiguous sets of word uses from two monosemous sets. The models may be sensitive to the different variances in these sets, and hence predict stronger change for more ambiguous sets of uses. Overall, our findings demonstrate that much more work has to be done to understand the effects of time-shuffling as well as sensitivity effects of LSC detection models to frequency and ambiguity.

5.4.9. Conclusion

We carried out the first systematic comparison of a wide range of LSC detection models on two datasets which were reliably annotated for sense divergences across corpora: A synchronic task, i.e. detecting meaning variations from general to domain language, and a diachronic task, which we use as comparison in order to test the robustness of the models. We base the synchronic LSC task on the gold standard SURel, which we introduced in the previous

section. The synchronic and diachronic evaluation tasks were solved with high performance and robustness. We introduced *Word Injection* to overcome the need of (post-hoc) alignment, but find that Orthogonal Procrustes yields a better performance across vector space types.

The overall best performing approach on both data suggests to learn vector representations with SGNS, to align them with an orthogonal mapping, and to measure change with cosine distance. We further improved the performance of the best approach with the application of mean-centering as an important pre-processing step for rotational vector space alignment.

Finally, this means that we have a best-performing method now, with which meaning variations for term candidates can be reliably predicted. We make use of these predicted variations in the next section, in order to correct automatic term extraction for terms with meaning variations.

5.5. Incorporating Meaning Variation into Automatic Term Extraction

After illustrating that the relatedness scores in SURel reflect degrees of meaning variation between general-language and domain-specific language usage in section 5.3, the current section demonstrates that (a) a standard measure for automatic term extraction does not capture meaning variation, and (b) we can utilize the meaning variation prediction from section 5.4 to modify existing measures to incorporate meaning variation into termhood prediction[15].

5.5.1. A Standard Term Extraction Measure

We selected one of the simplest standard contrastive term extraction measures, the *Weirdness Ratio* (WEIRD) (Ahmad et al., 1994), which is still commonly used or adapted (Moreno-Ortiz and Fernández-Cruz, 2015; Cram and Daille, 2016; Rösiger et al., 2016; Hätty et al., 2017a, i.a.). It encompasses just the basic ingredients for termhood prediction, a comparison of word frequencies in relation to corpus sizes:

$$\text{WEIRD}(x) = \frac{f_{spec}(x)/s_{spec}}{f_{gen}(x)/s_{gen}},$$

where f_{spec} and f_{gen} correspond to the frequencies of a term candidate x in a general and a domain-specific corpus, and s_{spec} and s_{gen} are the respective sizes of the corpora.[16]

[15] The work in this section is published in Hätty et al. (2019b).

[16] We use versions of our corpora which are limited to content words to be consistent with subsequent experiments.

The left panel in Figure 5.8 shows the ranking of the SURel target words after computing their WEIRD scores, with decreasing termhood scores for targets from left to right. The figure clearly illustrates that WEIRD ranks the target words with strongest meaning variation in SURel lowest, independently of their termhood: targets with high SURel scores are ranked as most terminological by WEIRD and occupy the first ranks (*Messerspitze, Eiweiß*, ...), and targets with low SURel scores are ranked as the least terminological ones and occupy the last ranks (..., *Form, schlagen*).

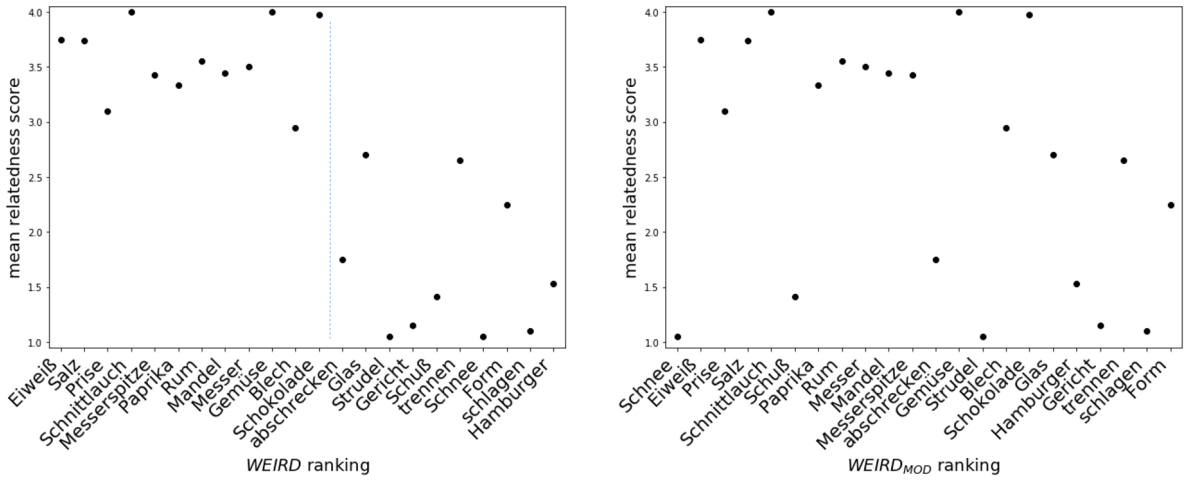


Figure 5.8.: SURel target words ranked by WEIRD (left panel) and WEIRD$_{MOD}$ (right panel), with termhood prediction strength decreasing from left to right; the y-axes show the SURel GEN–SPEC relatedness score.

To further investigate this bias, we looked up the SURel targets in (a) Wiktionary and Wikipedia, (b) the German dictionary Duden and (c) popular German translation dictionaries (Langenscheidt and PONS). If a word was assigned a cooking or gastronomy tag in any of these resources, we categorised it as a domain term. In this way, ten of our targets[17] were categorised as terms; seven of them are among the ten most non-terminologically ranked targets by WEIRD. This confirms that termhood predictions by WEIRD as a representative of contrastive termhood measures are strongly influenced by meaning variation of terms.

Although the influence of meaning variation might not be equally evident in other term extraction measures as in our simple example measure WEIRD, any other measure heavily relying on a general-language word frequency distribution will to some extent be negatively influenced by meaning variation of terms. Consequently, we need to correct the bias caused by meaning variation. In the following, we show that we can use SURel to assess factors that potentially reduce the bias.

[17] *Eiweiß, Messerspitze, Paprika, abschrecken, Strudel, Schuß, Schnee, Form, schlagen, Hamburger*

5.5.2. Correcting for the Meaning variation

For automatically predicting meaning variation we rely on the best model from the comparative study from section 5.4, the state-of-the-art model for diachronic meaning change by Hamilton et al. (2016b). We learn two separate word2vec SGNS vector spaces for GEN and SPEC. In order to compare the target vectors across spaces the spaces are aligned, i.e., the best rotation of one vector space onto the other is computed. This corresponds to the solution of the orthogonal Procrustes problem (Schönemann, 1966). If G and S are the matrices for the general and the specific vector spaces, then we rotate G by GW where $W = UV^T$, with U and V retrieved from the singular value decomposition $S^TG = U\Sigma V^T$. Following standard practice we then length-normalize and mean-center G and S in a pre-processing step (Artetxe et al., 2017). After the alignment, cosine similarity between the vectors of the same word in both spaces is computed. The cosine score of the two vectors of a word w predicts the strength of meaning change of w between GEN and SPEC, ranging from 0 (complete change) to 1 (stability).[18]

As input for the model, we use POS-tagged versions of our corpora, keeping only content words.

Evaluating the output of the model on the SURel dataset, we reach a Spearman's rank-order correlation coefficient of ρ=0.866 between the model's change predictions and SURel meaning-variation ranks. Inspecting the nearest neighbors (NNs) of our exemplary target words for meaning stability, reduction and change, *Schnittlauch*, *Messer* and *Schnee* (previously analyzed in Figure 5.7), confirms the ability of the model to predict strengths of meaning variation. For example, the NNs for *Schnee* change completely (from *mud, leaves, foggy* in the GEN space to *egg whites, foamy, beat* in the SPEC space), while for *Schnittlauch* all nearest neighbors in both spaces are cooking-related.

Finally, to correct WEIRD for the meaning-variation error, we incorporate the model's predictions of meaning change into the WEIRD formula, where $\alpha(x)$ corresponds to the model's predicted strength of meaning change for word x:

$$\text{WEIRD}_{MOD}(x) = \frac{f_{spec}(x)/s_{spec}}{(\alpha(x) \cdot f_{gen}(x))/s_{gen}}.$$

The right panel in Figure 5.8 shows the ranking of the SURel target words based on their WEIRD_{MOD} scores, again with decreasing termhood scores for targets from left to right. The plot clearly shows that WEIRD_{MOD} improves over WEIRD regarding the negative bias for

[18] Since *Messerspitze* occurred too few times in GEN, we did not compute a variation value and assumed variation.

word	predicted meaning variation value
Wikipedia	0.89
Artikel	0.89
Thema	0.78

Table 5.13.: Examples for wrongly predicted strong meaning variations.

meaning-shifted targets: now shifted target words do not gather in one part of the plot but occur across ranks. While WEIRD only reaches an average precision of 0.45, WEIRD$_{MOD}$ reaches an average precision of 0.59.

In the same way as we incorporated the Hamilton measure of semantic change into WEIRD, we could rely on other contrastive term extraction techniques and incorporate further measures of semantic change. SURel can be utilized to evaluate modifications and thus to optimize termhood prediction techniques regarding the sub-technical meaning variation bias.

5.5.3. Extension and Discussion

We presented a gold standard for meaning variation and how to use it for term extraction. Since our meaning variation prediction method works quite well with the however rather small dataset, we extend the target set and further compute the shifts for all nouns, verbs and adjectives in the cooking corpus with a frequency ≥ 50 in both SPEC and GEN. This results in meaning variation values for 1,125 words. In the following, we use the extended dataset for remarks on challenges for term extraction.

First, our dataset contains mostly words with at least some relevance to the cooking domain. The intuition behind this is, that for clearly non-terminological words (e.g. *anderes* 'different', *alternativ* 'alternative', *komplett* 'complete', *Ganze* 'whole') there should not be a meaning variation towards the domain. In practice, when applying our method, our system predicts a high degree of meaning variation for those words. Many of those words seem to be highly versatile in GEN and in SPEC. Additionally, especially problematic are words which occur without context in many cases (*Galerie* '[picture] gallery', *Inhaltsverzeichnis* 'table of contents'), or words with repeating similar context, as for example *Wikipedia*, *Artikel* 'article' and *Thema* 'topic', given in table 5.13. These three words reoccur in the sentence 'Wikipedia has one article to the topic ...' thus, the restriction on the two reoccurring context words for the respective target word in the domain has the same effect as a strong meaning reduction or meaning change towards these two words. After all, a sentence duplicate detection can

be applied for these cases. However, the first case is more difficult, and might need to be addressed as future work. Furthermore, due to homonymous surnames some words adopt a sub-technical behaviour, e.g. *Paul Auster* 'oyster', *Stefanie Kloß* 'dumpling' or *Paul Brie*. Although these cases are not very frequent, they are possibly hard to identify. Applying named entity recognition might be an option to address that problem. As a last effect, it seems that strictly monosemous terms accumulate at the other end of the scale, with especially low meaning variation values. We believe that this is due to the fact that a small percentage of cooking texts occurs in the general-language corpus as well, and words that only occur within cooking have similar contexts then.

To counteract against the errors for versatile words, we achieve some promising results with the following method: We compute a second meaning variation value, but here we shuffle the sentences across the corpora while preserving the target word's context sentence frequencies in each corpus. By that we obtain some kind of ground truth value for the word's context variance. The assumption here is that if a word already has strongly varying contexts throughout the corpora, then the high meaning variation across corpora is most likely a result from that. We finally substract the shuffling value from the meaning variation value. In the resulting ranked list, this method separates the non-terminological elements to the one end and many terms with meaning variation to the other end: *altbacken* 'dowdy/stale', *gedämpft* 'low voice/steamed', *Schnee*, *Fond* 'fund/stock', *Auflauf* 'crowd/casserole', *Form* 'shape/(baking) mould' together with other cooking-related words like *Spaghetti*, *Pfannkuchen* 'pancake', *Pommes* 'French fries', *Ananas* 'pineapple', where the latter words have a lower original meaning variation value. However, other sub-technical terms like *schlagen* 'beat/whip (cream)', *abschrecken* 'discourage/chill ', *binden* 'tie/thicken (sauce)' are still among the non-terminological elements, most likely because they have rather varying contexts in GEN as well. Nevertheless, for terms with meaning variation identified with the described method the original meaning variation value could be used to correct a termhood measure.

5.6. Summary

In this chapter, experiments were described which were designed to get insights about models for predicting meaning variation, and to realize improvements in conventional automatic term extraction. Basically, two problems were addressed: First, identifying various forms of terms, from simple terms to multi-word terms, with additionally having a special focus on constituents of the multi-word terms. Secondly, the problem of ambiguous terminology, i.e. terms with meaning variation between general language and a specific domain language was

addressed.

In a first experiment, we extracted term candidates from context sentences of a corpus of ACL publications. Methodologically, we used several classes of term extraction measures as features for decision tree and random forest classifiers, to get interpretable models. The most interesting finding was that constituents are important: the system can downgrade term candidates with clearly non-terminological constituents (e.g."new grammar formalism"), and we found that especially modifiers are responsible for downgrading. Secondly, we found that for longer multi-word terms, the system relies mainly on unithood, i.e. association measures. This might be due to the rigid publication language and might not be reproducible for more open contexts. In all, the results indicated that the interplay of complex terms and constituents is relevant for classifying terms, which is why we want to apply a model addressing this problem again for the more advanced termhood models in the next chapter.

As a second aspect, we dealt with meaning variation of terms. We first presented SURel, a German dataset for meaning variation annotations from general to domain-specific language, focusing on the language of cooking. Meaning variations are relevant for contrastive term extraction systems, because the affected terms are typically biased towards their general-language use and, consequently, might not be recognized as terms. SURel can be used as a gold standard for predicting meaning variation, and these predictions can be used to optimize term extraction measures. The experiments were divided into three steps: a) the conception and annotation of a meaning variation gold standard (SURel), b) computational modeling of meaning variation prediction and c) using the shifts to improve automatic term extraction systems which use contrastive approaches. We demonstrated the improvement for term extraction on the SURel dataset. However, this gold standard set is rather small, and when manually inspecting unlabeled data samples, there are still errors for very general words; for those, the context changes from general language to specific language in a similar way as it is the case for terms with meaning variation. As a conclusion, there is still room for improvement. We will address this problem again in the next chapter, when testing the more fine-grained termhood model. For that, we will still rely on the basic meaning variation architecture described in this chapter, but a classification system will be given the information about both the vector information and the vector differences, and it can then dynamically decide on which information it has to rely on in order to predict the correct termhood class.

6. Extended Automatic Term Extraction: Complex Terms, Meaning Variation Revisited

6.1. Introduction

This chapter covers experiments for automatic term extraction that are based on extended term definition frameworks. The term definition annotation studies in chapter 4 showed that a term's nature should be understood as being scalar and not binary, but otherwise it became clear that establishing an extended framework for term definition is rather difficult.

While experimenting with different term frameworks in human annotation studies in chapter 4, we also experimented with scalar representation of terms for automatic term extraction which is described in this chapter. In order to increase inter-annotator agreement to get reliable gold standards for evaluation, we took several measures: We aimed at avoiding the problem that annotators cannot agree on the linguistic form of a term. Therefore we restricted to items that occurred as a single orthographic word, which means that we restricted the set of term candidates to simple words and closed compounds. Since closed compounds are a dominating phenomenon in German, we laid a special focus on the latter in this chapter. Furthermore, we do not let annotators rate on a scale of values where we just define the minimum and the maximum value, but discrete categories are predefined, arranged on an ordinal scale. The goal is that by clearly defining the intermediate steps on a scale will guide annotators in a better way and will result in less diverging annotations.

How we designed the ordinal scale was still subject of experimentation throughout the studies. In the first experiment in section 6.2, we established a term scale that implicitly flattens the two-dimensional centrality-specificity framework described in section 4.4 to a one-dimensional scale. In the following two experiments, we then completely shift to some representation of **specificity**. In section 4.4, we defined specificity as the level of domain-specific information a term carries, and symptomatically then as the level of **difficulty** a lay person

attributes to a term.

We shift to specificity for the last two experiments in sections 6.3 and 6.4 because as chapter 4 showed, specificity is more intuitive than centrality, and received better inter-annotator agreement. Furthermore, the prediction of specificity has been investigated (predominantly as term 'familarity', the counterpart to 'difficulty') in several studies, for example by exploiting classical readability features such as term length or syllable count (Zeng et al., 2005; Zeng-Treitler et al., 2008; Vydiswaran et al., 2014; Grabar et al., 2014) or by relying on context-based approaches (Zeng-Treitler et al., 2008; Bouamor et al., 2016; Pérez, 2016). These studies mostly focus on the medical or bio-medical domain. We therefore see the need to extend the automatic prediction of specificity to other domains. Another important aspect is to understand the characteristics of specificity, and how we can make use of them to automatically predict specificity. For those features, we mainly rely on the ones we also used for classical automatic term extraction (chapter 5): hand-crafted term features and vector space models. Since we lay a special focus on closed compounds in this chapter, we further use compound features. The studies in this chapter can be seen as an expansion of the studies in the previous chapter: Instead of distinguishing between terms and non-terms, we aim at predicting a more fine-grained term model. We deal again with the two problems of the previous section for this extended task: a) complex terms and constituents, whereby complex terms are constrained to **German closed compounds** (sections 6.2 and 6.3), and b) **ambiguity** (section 6.4). Since we concentrate on compounds when dealing with complex terms, we provide an addendum that focuses on evaluating **compound splitting in specific domains** (section 6.5).

6.2. Fine-grained Term Prediction for Closed Compounds

In this section, we extend the conventional binary to a quaternary decision problem, introducing four classes representing distinct points on a ordinal term scale. We explain in detail the reasons for the selection of those classes. As basis for the annotation and prediction of termhood, 400 German closed compounds are taken as term candidates of the domain 'cooking'. Closed compounds are especially interesting for this task, since there is a comprehensible interplay between the termhood classes of the compound and its constituents. In addition, compounds pose a further challenge to term extraction: while determining unithood is not an issue (it is evident which constituents belong to the term), to evaluate termhood automatically is harder especially if compounds are infrequent.

For the prediction of the termhood classes, we take two steps: First, we apply compound splitting to obtain the constituents of the term candidates. Secondly, we design neural network architectures to predict the classes, in a way that information about constituents is adequately incorporated. We show that the interplay of the compound's and the constituents' termhood improves the prediction of the compound's termhood class, and that constituent information is especially important for this extended concept of termhood.

6.2.1. Closed Compounds and Term Extraction

In chapter 4 we concluded that closed compounds are a term type whose occurrence as single orthographic words makes it easier for annotators to agree on than on other multi-word expressions. The one-word form is an advantage for term extraction as well, since the unithood does not need to be evaluated. In addition, compounding is a frequent phenomenon: Baroni et al. (2002) found that 47% of the word types in a general-language corpus were compounds, and according to Clouet and Daille (2014) compounding is more productive in specialized domains. However, automatically identifying closed compounds as terms is challenging. Baroni et al. (2002) also find that 83% of the compounds have a maximal corpus frequency of 5. Frequency-based automatic term extraction methods might return wrong results then. As a solution, other compound attributes can be relevant for term extraction processes. One important attribute is **productivity**. Productivity can be defined as the number of compound types that share a constituent. This attribute is also known as **morphological family size** (De Jong et al., 2000), but we remain with the term used in earlier work (cf. Schulte im Walde et al., 2016). Concretely, we compute the productivity of the modifier and the head as following:

- **productivity of the modifier**: In how many compound types does a certain word type take the position of the modifier?
- **productivity of the head**: In how many compound types does a certain word type take the position of the head?

6.2.2. Related Work

Many of the major term extraction measures described in background section 3.1.2 rely on statistics about constituents. In our previous study described in 5.2, we combine several termhood measures with a random forest classifier to extract single and multi-word terms. For multi-word terms, the measures are recursively applied to the constituents.

Constituent information has also been used for the related task of keyphrase extraction: Erbs et al. (2015) split German compounds to enhance individual term frequencies, leading to an improved keyphrase extraction. Very influential to our work was the study by Zhang et al. (2016a), who use a joint-layer RNN for both classifying keywords and keyphrases; the information is cumulated for the keyphrase prediction. We adopt the idea of separate sub- and superstring objectives for our termhood class recognition system, but adapt the input data, the constituent objective and the general network architecture to our more specific needs.

6.2.3. Description of the Problem

While term extraction is often perceived as a binary task, we already introduced theories that terminology can be arranged into tiers (Trimble, 1985; Roelcke, 1999). Strength of association of a term to a domain is defined along its exclusive usage in one domain and its topical relation to it. Accordingly, we define the first three termhood classes: NONTERM, SIMTERM and TERM. The classes are described in table 6.1. There are the clear non-terms with no topical relation to the domain, but which can still appear in the domain context, like the example *Deutschland* in *Gericht aus Deutschland* "dish from Germany". The class SIMTERM includes all non-terms which are related to the domain to some extent, either by having received a special relevance for the domain (e.g. *Zimmertemparatur* "room temperature" as a temperature measure for dishes) or by semantic relatedness (e.g. *Tiernahrung* "pet food"). The class TERM represents the typical, clearly in-domain term. For this strong association to the domain, we can even introduce a fourth class: We distinguish in-domain terms between understandable and non-understandable or expert knowledge. The reason for this is that the more difficult or specialized a term is, the more distinctive it is from general language and therefore the more it is associated to a domain. If terms are both general and understandable, it is sometimes hard to distinguish them from general-language words. Thus, the more a term belongs to expert knowledge, the stronger it is associated to a domain. A description for the class SPECTERM is also shown in table 6.1. The concept of SPECTERM relates to our attribute of **specificity**, described in 4.4.
Finally, we have four classes which represent domain association strength, from NONTERM to SPECTERM in ascending order.

As a second issue of the termhood problem, we go into detail for the role of closed compounds. Closed compounds, as they often occur in German, can only be identified as a term as a whole (with all its parts) or not all. This distinguishes them from other multi-word expressions: On the one hand, an advantage of compounds is that it is clear which parts belong to the term. On the other hand, this implies that one constituent could be a term but another con-

Class	Description	Example
NONTERM	not a term	*Deutschland* "Germany"
SIMTERM	semantically related to the domain	*Vitaminbedarf* "requirement of vitamins"
TERM	understandable term	*Schweinebraten* "roast pork"
SPECTERM	non-understandable term	*Blausud [blue boiling]* "special kind of boiling fish by adding acid"

Table 6.1.: Termhood Classes.

stituent could inhibit the whole compound from being a term. For example, *Maismehl* "maize flour" is a cooking term, but *Maisanbau* "maize cultivation" is not. This distinction might be not so difficult for human annotators, but might pose a challenge to a termhood prediction system.

This aspect leads us to the relation of compounds to their constituents, and why constituents play an important role for our fine-grained termhood classification:

In the simplest case, the constituents exactly define the term category of the compound:

Tomate (TERM) + Püree (TERM) → Tomatenpüree (TERM)
tomato + *puree* → *tomato puree*

This is the ideal case where only knowledge about the constituent is necessary to predict the compound's term category.

In other cases, knowing the term category of the constituents does not necessarily allow us to infer the term category of the compound. Sometimes the term category of the constituents is the same, but the compound has a different term category :

Mittel (NONTERM) + Alter (NONTERM) → Mittelalter (NONTERM)
mean + *age* → *Middle Ages*

Schwade (NONTERM) + Gabe (NONTERM) → Schwadengabe (SPECTERM)
swath + *giving* → *steam injection*

In the opposite case, the compound class is the same but the constituents' term category changes:

Paprika (TERM) + Salat (TERM) → Paprikasalat (TERM)
sweet pepper + *salad* → *sweet pepper salad*

Paprika (TERM) + Hälften (NONTERM) → Paprikahälften (TERM)
sweet pepper + *halves* → *halves of sweet pepper*

Nevertheless, even in these cases constituent information is useful because certain term categories can be excluded with it. Information about the compound becomes less relevant. In addition, we expect that a prediction system will learn the interplay between the term categories of compounds and constituents, and even if the information about a compound is missing, it can be transferred from other compounds of the same kind. Since especially broad domains like cooking contain many neologisms (e.g. because of new recipe creations: *Parmesanchips* "crisps with Parmesan cheese"), and are anyway very productive, this effect is very advantageous. Many compounds will not be frequent enough to be evaluated for their term categories and thus, constituents need to be evaluated.

6.2.4. Data and Annotation Procedure

We started with a seed set of compound terms: 34 cooking recipes were selected randomly from *kochwiki*[1] and *wikibooks*[2]. From these texts, roughly 260 cooking compound terms were identified by three annotators. After briefly reviewing the data, the portion of SPECTERM elements seemed to be very low (about 10 very specific terms). We therefore manually extracted compound terms from cooking and cooking-related term lists from Wikipedia. For the classes NONTERM and SIMTERM, we retrieved unannotated compounds from the cooking recipes. We additionally added 15 compounds for the class SIMTERM which have some semantic similarity to the cooking domain (e.g. *Tiernahrung* "pet food", *Küchenzeile* "kitchenette").

For this experiment, the compounds should appear sufficiently frequently that a word embedding can always be created based on Wikipedia and the cooking domain texts. The reason for this is that we want to show that constituent information improves the prediction results; thus we want to have an optimal setting for compounds against which the new information can be compared and added to. For the same reason, there is no minimum frequency for the constituents, to have a realistic setting here. We crawl texts from wiki pages (e.g. *kochwiki,*

[1] https://www.kochwiki.org/
[2] https://de.wikibooks.org/wiki/Kochbuch

wikibooks and *wikihow*[3]) to obtain the minimum frequencies. In total, the collection contains 404 texts from the cooking domain, consisting of roughly 150,000 words. The text collection consists mostly of recipes, but contains descriptions of ingredients or cooking methods as well.

All in all, since a sufficient amount of occurrences could not be found for all compounds, the compound set finally comprises 400 elements.

The 400 compounds are prepared for the final annotation process: the compound terms are provided with either a definition (from *Wikipedia, kochwiki* or *Wiktionary*[4]) or, if a definition could not be found, a context sentence from the cooking recipes. Then 5 annotators were asked to decide for one of the four classes NONTERM, SIMTERM, TERM and SPECTERM.

The final classes are selected via majority vote of the annotators. For 46% of the compounds there was a complete agreement (5 out of 5), for 29% there was a 4:1 agreement. For 4 terms, there was no majority voting; these terms are excluded from the compound set, resulting in 396 terms overall.

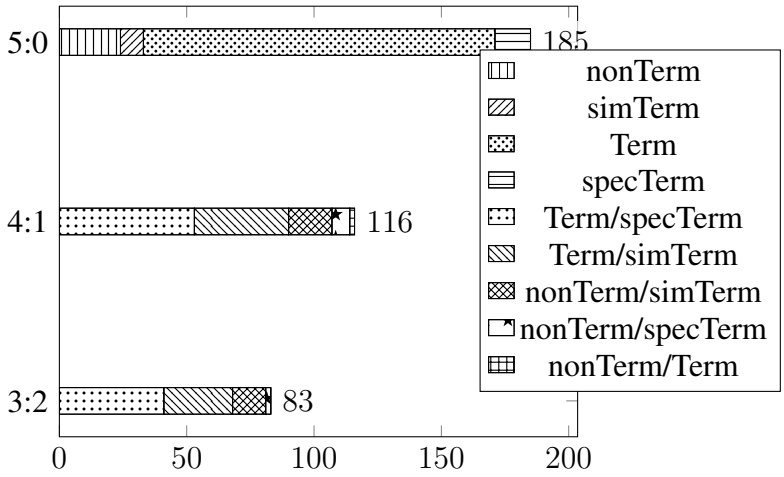


Figure 6.1.: Agreement and disagreement for term category annotation.

Figure 6.1 shows agreement (5:0) and disagreement (4:1 and 3:1) for term class annotations. The 5:0-bar shows the number of elements for which the decision was unanimous. These elements are distributed over all classes. For 4:1 and 3:2 disagreements, the number of elements are shown for the two classes between which the decision was made. Annotators were mostly disagreeing between TERM and SPECTERM. This is an expected result, since the knowledge level even of non-expert annotators differs to a certain degree, resulting

[3] https://de.wikihow.com/
[4] https://de.wiktionary.org/

in a different class selection. Furthermore, nearly all divergences are between neighboring classes (nonTerm/simTerm, Term/simTerm, Term/specTerm) while the divergences between other classes are negligible. This is a good indicator that the class design represents a valid scale of termhood.

After taking the majority vote we get a final distribution of elements per class, shown in table 6.2.

NONTERM	SIMTERM	TERM	SPECTERM
44	43	250	59

Table 6.2.: Number of annotated terms per class.

6.2.5. Compound Splitting

For compound splitting, we use three splitters:

- **CharSplit** (Tuggener, 2016): a n-gram-based splitter which is reported to have a good performance (95% accuracy for head detection on the Germanet compound test set)
- **CompoST** (Cap, 2014): a compound splitter which uses both the morphological resource SMOR (Schmid et al., 2004) and frequency information about constituents in corpus data for finding the optimal splitting points
- **Simple Compound Splitter** (Weller-Di Marco, 2017, further on: **SCS**): uses a frequency-based approach in combination with a set of hand-crafted rules.

CharSplit always splits, this is why we take it as the base splitter. Table 6.3 shows the correct compound splittings for the (combinations of) the three splitters. For our specific domain, without special training with only using CharSplit, we achieve 81.4% correct splits. There are 25 completely wrong splits (e.g. *Voll|lei* "whole egg" instead of *Voll|ei*), 9 elements have splits on the wrong side (e.g. *Marzipanroh|masse [marzipan raw mass]* "marzipan paste") instead of *Marzipan|rohmasse*).

To improve the results, we further use two morphologically informed splitters. We use CompoST, which splits conservatively, leading to a high splitting precision, but many unsplit compounds. For this reason, in the next step, we first apply CompoST, and for the rest of the compounds we apply CharSplit. This leads to an accuracy of 94.2%. Since there were

still problems with recognizing the compounding stem without plural -N (*Traube|Nsaft* "grape juice" instead of *Trauben|saft*), we then added the Simple Compound Splitter, which explicitly models this phenomenon. Weller-Di Marco (2017) compared it to a SMOR-based splitter, a variant of CompoST. SCS exhibits a higher recall and a higher F1-score. Since SCS needs a basis of POS, lemma and frequency information, we compute this information on the cooking dataset. Then we combine the three splitters in the way that we start with CompoST, then apply SCS and finally use CharSplit for the rest of the compounds. This results in the best split score of 95.7%.

We split on unlemmatized forms, since we expect that if the compound's head occurs as a single term in text, it will have the same unlemmatized form in many cases (e.g. *Walnüsse* and *Nüsse*). If the lemmatization goes wrong, otherwise the second constituent cannot be found anymore (a first inspection of the lemmatization by the Mate Tools (Bohnet, 2010) and the TreeTagger (Schmid, 1994) showed that compound lemmatization is rather erroneous). The modifier is lemmatized by CompoST and SCS. This is important, because the modifier often is a compounding stem form of the respective word (e.g. for verbs: *Marinier|zeit* "time for marinating" → *marinieren|Zeit*).

Splitters	**Wrong splits**	**Wrong side**	**% correct splits**
CharSplit	25	9	91.4 %
CompoST + CharSplit	14	9	94.2 %
CompoST + SCS + CharSplit	9	8	95.7 %

Table 6.3.: Splitting performances of the three compound splitters.

6.2.6. Termhood Prediction

After having created the gold standard dataset with annotation of the four term categories, the next step is to build classification systems that adequately predict the classes given the provided input information, i.e. features about compounds and constituents.

Word Embeddings

As basis for the experiments, we compute word embeddings for compounds and constituents. We use the Word2Vec CBOW model (Mikolov et al., 2013b) to generate 200-dimensional vectors. The vectors are first trained on Wikipedia, and then posttrained on the 404 cooking texts. 27 constituents could not be found, partly because they were infrequent, do not exist as

an independent word and mostly because they were falsely split or a lemmatization of the first constituent was missing. To those constituents, we assign a randomized word vector.

Basic Neural Network Architecture

Our basic neural network construction can take one or several words as input, for which the respective weights in the embedding layer are already set; the weights are initialized with the pre-trained word embedding weights. The final dense layer with a softmax activation predicts the four classes. The architecture is shown in figure 6.2.

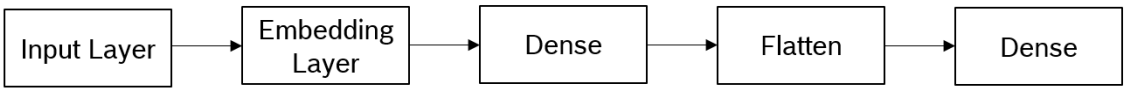


Figure 6.2.: basic neural network architecture.

Hyperparameters

The parameters of the neural networks are set to the same values for all network constructions and experiment implementations: The batch size is set to 32, epochs to 50. All dense layers have a dimensionality of 64. The word embeddings are used to initialize the word weight matrix. The matrix is not updated in the training process (since too many parameters would need to be trained). We do not focus on optimizing the model parameters, but aim at comparable architectures to show the effectiveness of the information gained by the input data.

Incorporating Compound and Constituent Information

In the following, we describe the construction of models which incorporate both compound and constituent information. The models build on each other, and we show how step by step additional information about the constituents leads to a better recognition of the compound class.

Compound and Constituent Vectors: CONCATVEC. Compounds and constituents are jointly taken as input to the basic neural network architecture. This is a first step to contrast the effect of concatenated compound and constituent word embeddings against only using the compound or constituent embeddings for prediction.

Productivity and Compound Frequency: VECPRODFREQ. When only taking word embedding information as basis for the prediction, the models only get cumulated information about both the more general domain Wikipedia and the specific cooking domain. However, for the constituents, additional information about their occurrence in the specific domain could be useful. There are two reasons why: On the one hand, very specific term parts of the class SPECTERM can be better distinguished from TERM, since these should tend to be less productive and less frequent. On the other hand, very general or ambiguous constituents can be tested for their relevance for the domain; for example, constituents like in *Teigränder "rim of pastry" or* Lorbeerblatt *"bay leaf"* are non-terms. The system could learn that they are nevertheless constituents which are accepted within a TERM or SPECTERM compound, if they appear in many other compounds terms as well. Thus, we introduce two further features here:

- Productivity of constituent
- Frequency of constituent (as word)

To include this information, we introduce an additional input layer into the network for the productivity and frequency features of the two constituents. The information is joined within the network and compound classes are again predicted with a softmax output layer.

Optimization for termhood of constituents: CONSTOPT. As a variant to join the compound and constituent embedding information more effectively, we finally use a multi-input multi-output shared-layer model, which is depicted in figure 6.3. The model takes five inputs, one for the compound and one for each of the constituents, and one for the productivity and frequency features for each of the constituents. All information for each of the constituents is joined within the network. For each of the three main chains (compound and each constituent) an auxiliary output layer is introduced. For the compound, the four established classes are predicted. The auxiliary output layer here is a mere regularization mechanism. For the constituents, optimization by the auxiliary outputs should sharpen their termhood, since we consider the constituent's termhood as strongly influential for the termhood of the compound (as explained in detail in section 6.2.3). We do not have information about the intrinsic termhood of the constituent, but we infer it from the compounds in which a constituent occurs. As basis, we take all the compounds in the training set and create four classes analogous to the compound classes. We make the following distinctions:

- *specific terms*: all constituents that only appear in SPECTERM compounds. Thus, the difference to non-specific terms gets sharpened.

- *similar terms*: shared constituents of SIMTERM and TERM. These SIMTERM constituents are especially critical, since then the other constituent decides the class (see section 6.2.3)
- *term*: all constituents which occur in TERM compounds
- *other*: the rest of the constituents

We set the loss weight for the main output to 1, and the weights for the auxiliary outputs to 0.2.

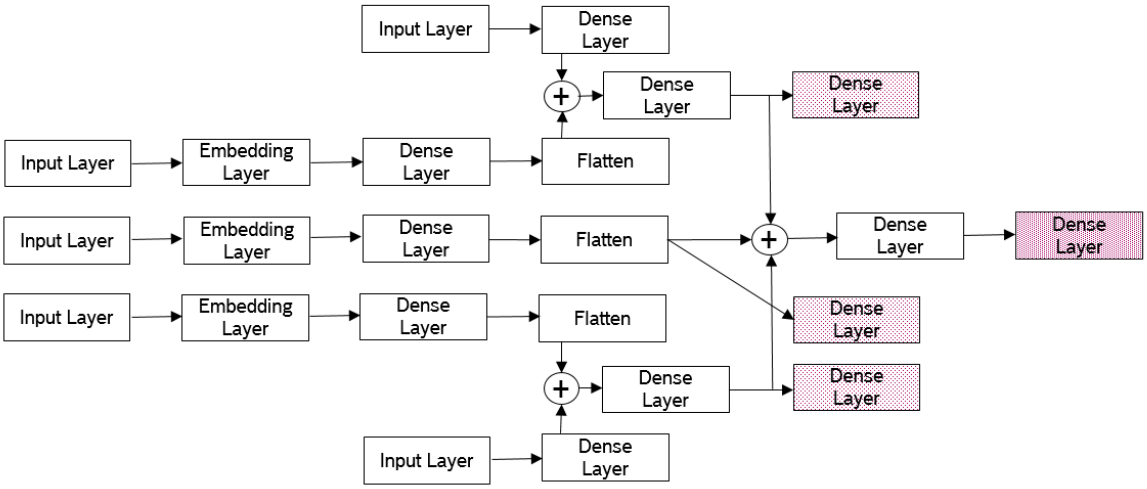


Figure 6.3.: CONSTOPT model.

6.2.7. Results and Analysis

In table 6.4 we provide the results for the different models. We applied 5-fold cross-validation and use Precision, Recall and F1-score as evaluation measures. In brackets next to the method name, the overall F1-score is given, weighted and averaged over all classes.

For the baselines, results show that taking only compounds as input provides a better result than only taking constituents as input. However, one should not forget that we provided ideal conditions for compounds, by enforcing a minimum frequency. When combining the

compound and constituent features, the overall F1-score continuously increases for every improvement we made to the system.

Focusing on the individual classes, NONTERM and TERM achieve a high recognition rate. We attribute this to the fact that they are the two prototypical opposing classes. SIMTERM and SPECTERM reach only decent results at first. However, when comparing the best model to the best baseline model, the results for the two classes experience a boost. The prediction for SIMTERM gains 17% F1-score, SPECTERM even gains 21%.

Method	**Non-Term**	**SimTerm**	**Term**	**SpecTerm**
Constituent Baseline [0.69]				
Precision	0.77	0.47	0.78	0.53
Recall	0.50	0.40	0.89	0.32
F1-score	0.6	0.41	0.83	0.37
Compound Baseline [0.72]				
Precision	0.86	0.61	0.79	0.47
Recall	0.77	0.42	0.86	0.37
F1-score	0.80	0.48	0.82	0.40
CONCATVEC [0.75]				
Precision	0.83	0.54	0.82	0.61
Recall	0.68	0.40	0.89	0.54
F1-score	0.74	0.44	0.85	0.57
VECPRODFREQ [0.77]				
Precision	0.85	0.66	0.82	0.62
Recall	0.73	0.53	0.88	0.54
F1-score	0.77	0.58	0.85	0.57
CONSTOPT [0.80]				
Precision	0.88	0.69	0.85	0.64
Recall	0.75	0.63	0.9	0.59
F1-score	0.79	0.65	0.88	0.61

Table 6.4.: Results for baselines and advanced models per class.

Finally, we analyze the predictions of the model to find reasons for misclassification. One aspect which seemed to be likely are wrong splits (e.g. *Ei|klar* "egg white", split to *Eik|lar*). However, only 4 out of 17 wrongly split compounds were predicted wrongly. Since no embeddings could be computed for wrong constituents, the model probably learned to rely on other features in this case. The most interesting classes for deeper analysis are SIMTERM and SPECTERM, since the other two classes are predicted well. For SIMTERM, we do not recognize differences between the correctly classified compounds, and the ones which are predicted

as TERM. For SPECTERM, we find at least two kinds of terms that seem to be classified mostly correctly: rather opaque terms (e.g. *Fetthenne [fat hen]* "sedum"), and many terms with one non-understandable, and one understandable constituent term (e.g. *Tonkabohnen* "tonka beans"). A lot of SPECTERM compounds are classified as TERM. One reason for this might be that a compound includes a very typical TERM constituent (e.g. *Blindbacken* "blind-baking").

6.2.8. Conclusion

We presented a term extraction approach for a fine-grained framework designed to classify different categories of terms and non-terms. The categories form an ordinal scale, and thus shall represent different degrees of termhood. The study concentrated on German closed compound term candidates. As data basis, 400 German compounds of the cooking domain were selected for annotation.

The focus of the study was placed on how features of the compounds' constituents influence their termhood, and how this can be applied in a classifying system. We first explained how the meaning and the term category of the constituents can influence the compound term category. On this basis, we designed several neural network models for the prediction of the defined term categories for compounds. We use word vectors to represent the meaning of compound and constituents, and productivity and frequency as additional constituent attributes. All these features improve the prediction result. As we already described, the term categories of the constituents have an influence on the term category of the compound. Thus, we design a further classifier that predicts the term categories of the constituents based on a heuristic in order to optimize the neural network in general and thus improve compound prediction at the same time. In the next section, we want to analyze the influence of compound and constituent features such as word vector, frequency and productivity on compound term prediction further.

6.3. Predicting Difficulty of Closed Compounds across Domains

This section describes experiments on the automatic prediction of the difficulty of domain-specific compounds for three domains: cooking, DIY and automotive.[5] We restrict these

[5]The analyses in this section are based on the results of Julia Bettinger's master thesis (Bettinger, 2019). Parts of the following descriptions are published in Bettinger et al. (2020). It is planned to publish other parts as

experiments to the treatment of closed German noun compounds to obtain complex term candidates with clear boundaries.[6] In addition, we make a second simplification step: in contrast to the previous paragraph, we do not separate terms from non-terms and only distinguish terms with respect to difficulty. We restrict the task to distinguishing all compounds in terms of their difficulty levels.[7] In exchange, we propose a more fine-grained analysis of domain-specific compound difficulty, and we investigate two groups of characteristics, which we assume to be possible sources of domain-specific compound difficulty: characteristics of terms and characteristics of compound formation. Consider the following examples:

- *Lieferwagen* (“delivery van", domain: automotive)
- *Collinsglas* (“Collins glass", domain: cooking)
- *Blindbacken* (“Blind-baking”, domain: cooking)
- *Paprikamarmelade* (“pepper marmelade”, domain: cooking)

For *Lieferwagen*, the compound and both its constituents occur frequently in general language, thus the compound is easily understandable. *Collinsglas* is compositional, but since *Collins* is a name (the name of the cocktail the glass is made for), domain-specific knowledge is needed to fully understand the concept. At first sight, *Blindbacken* has similar characteristics as *Lieferwagen*: both its constituents occur frequently in general language. However, *Blindbacken* needs domain knowledge to understand, the constituent *blind* seems to be rather awkward in the cooking domain context. One could argue that *Lieferwagen* as a whole most likely occurs frequently in general language, and *Blindbacken* does not; this would be an indicator for the assumption that the first compound is easier than the second one. Let us now consider *Paprikamarmelade*: it will not occur frequently in general language, and both constituents will occur frequently in general language - in this respect, it has the same attributes as *Blindbacken*. Nevertheless, it is easier to understand, because the meaning of the compound can be derived from the meaning of the constituents. In short: both the relationship of a compound to its constituents (i.e. compositionality) *and* their term attributes may influence the difficulty of a domain-specific compound. In order to investigate this hypothesis, the following study is divided in two parts: the creation of a gold standard dataset for domain-specific

well.

[6] According to Nakagawa and Mori (2003), the majority of domain-specific terms are compound nouns anyway.

[7] In a sense, we describe specificity as it is defined on the two-dimensional specificity-centrality-scale. Specificity on its own cannot distinguish easy terms from easy non-domain-related words, but the higher the specificity level, the more likely it is that a term candidate is actually perceived as term. In section 6.4, we thus add an explicit non-term class.

compound difficulty and the design of models which can automatically predict this difficulty. For the gold standard, a set of compounds is extracted from the domain-specific corpora, using a compound splitting tool. A subset of 1,030 compounds is finally selected, by balancing and filtering from the set of compounds. The set is subsequently rated manually for difficulty. For the modeling part, two kinds of systems are designed. We exploit different categories of termhood and compound features with using a decision tree classifier, which gives us the possibility to interpret how features interact. In a second step, we use neural networks with compound and constituent vectors, to gain insights on the influence of (semantic) vector information. For the binary classification 'easy' vs. 'difficult', we achieve a Micro-F1-Score of 0.75 with our decision tree classifier. This score is outperformed by the neural network approach reaching a Micro-F1-Score of 0.78.

The section is structured as follows: In 6.3.1 the data processing, and in 6.3.2 the creation of the gold standard is described. Subsection 6.3.3 deals with termhood and compound features, and 6.3.4 with the models. In subsection 6.3.5 the results are described and interpreted, which is again summarized in the conclusion.

6.3.1. Data Collection and Preprocessing

Corpora. In a first step, the domain-specific corpora are crawled and all compounds are extracted from text. We select the cooking domain because there was a large amount of text data available: Recipes, ingredient and technique descriptions, and more, crawled from `kochwiki.org`, `wikihow.de`, `wikibooks.de` and related Wikipedia articles. For DIY, we had a DIY corpus already available, containing online texts mostly crawled from the BOSCH empowered homepages `bosch-do-it.de` and `1-2-do.com`. The corpus consists of user-generated content as well as of expert texts (tool manuals, books on handicraft[8]). We further add material from `wikihow.de`. We finally choose the automotive domain because it contains a lot of technical terms. Texts are crawled again from Wikipedia and `wikihow.de` and the contents of an automotive handbook[9] are taken. For all domains, Wikipedia is crawled recursively by categories. The Wikipedia categories are manually filtered for categories which are contentwise too far away, as a further data cleaning step to maintain the topical focus of the corpora. Finally, all corpora are reduced to the size of the smallest corpus, which results in equally-sized corpora of 5.6 million tokens. The texts are tokenized, lemmatized and tagged with spaCy[10]. We applied lemma correction. For the exper-

[8] e.g. Holger H. Schweitzer. *Das große Heimwerkerbuch: Techniken, Geräte, Materialien*. Verlag Eugen Ulmer.
[9] Konrad Reif and Karl-Heinz Dietsche. *Kraftfahrtechnisches Taschenbuch*. Springer-Verlag.
[10] `https://spacy.io/`

iments later on, we further used SdeWaC (Faaß and Eckart, 2013) as a corpus that represents general language.

Extraction of closed compound nouns. We only extract closed compound nouns, which means the head and thus the whole compound is a noun. The following types can occur in text:

Kartoffelsalat$_{\text{Noun}}$ ("potato salad") = **modifier**: Kartoffel$_{\text{Noun}}$ + **head**: Salat$_{\text{Noun}}$

Kochtopf$_{\text{Noun}}$ ("cooking pot") = **modifier**: kochen$_{\text{Verb}}$ + **head**: Topf$_{\text{Noun}}$

Weißbrot$_{\text{Noun}}$ ("white bread") = **modifier**: weiß$_{\text{Adj}}$ + **head**: Brot$_{\text{Noun}}$

All compounds in the texts that were POS-tagged as nouns are extracted with the Simple Compound Splitter. We chose the SCS over other compound splitters because of its capabilities that were convenient for our task: All constituents are lemmatized and POS-tagged, and the splitter is capable of doing both binary and multiple splits. The SCS splitter was directly trained on the domain-specific corpora. The number of extracted compounds per domain is given in table 6.5. We mainly focused on two-part compounds, but due to the high number of longer compounds in the automotive domain (and the expectance that these are highly technical), we also extracted three-part and four-part compounds there. However, in the later processing steps, we still treat them as two-part compounds and only split them at the main split point.

domain	**constituents**	**frequency**
cooking	2	42,484
DIY	2	45,724
automotive	2	81,323
	3	73,675
	4	5681

Table 6.5.: Compounds extracted by the SCS splitter (Weller-Di Marco, 2017).

Table 6.5 shows that more two-part compounds are extracted for the automotive domain than for DIY and cooking. This is in line with our observation that automotive is the most technical of the three domains and the statement by Clouet and Daille (2014), that "*[compounding] is particularly productive in specialized domains because of the necessity to denote the domain concepts in a very concise and precise way*" (p. 11).

6.3.2. Creation of a Compound Term Difficulty Gold Standard

Since the set of retrieved compounds is too big to annotate it completely, we selected a balanced subset. The following characteristics are relevant for our task:

- **frequency** of compound and constituents
- **productivity** of modifier and head

Concretely, we then choose the following four criteria for balancing:

- frequency of compound
- productivity of the head
- productivity of the modifier
- frequency of the head

Before balancing, we exclude all terms with a frequency smaller than three, because the annotators will be given three context sentences. This results in a pool of 12,400 cooking compounds, 16,935 DIY compounds and 20,468 automotive compounds. The set is balanced by dividing into tertiles, i.e. dividing the set into groups of *low*, *mid* and *high* frequency and productivity. This results in $3^4 = 81$ classes. Then compounds are randomly selected from each class, and are checked by two annotators if the compounds are valid and split correctly. We further randomly inject a small amount of compounds which we find difficult to counteract against the presumed imbalance of the dataset for easy compounds. The final quantity of selected compounds for the gold standard is found in table 6.6.

domain	**constituents**	**frequency**
cooking	2	243
DIY	2	243
automotive	2	243
	3	162
	4	139
total		1030

Table 6.6.: Final gold standard set of compounds.

The final dataset is rated by 26 annotators in total, 16 women and 10 men. The annotators are shown the highlighted compound in context of three domain-specific sentences, and annotators are asked to rate each compound on the following Likert-like scale (Likert, 1932)[11]:

[11] Bettinger (2019, p.17). The original instructions were given in German.

1: The term does not require any specialized knowledge in order to be understood.
2: The term requires little specialized knowledge in order to be understood.
3: The term requires specialized knowledge.
Parts of its meaning can be inferred from its context.
4: The term requires specialized knowledge.
Its meaning cannot be inferred from its context.

After the annotation process, we selected the 20 annotations where annotators agreed most. Table 6.7 shows the Fleiss' κ (Fleiss, 1971) and the Spearman's ρ (Siegel and Castellan, 1988) agreements. One can see that the results are rather low for Fleiss kappa; Spearman's rho, which measures not only the values but also the ranking, is higher, with an overall agreement of 0.614.

domain	**Spearman's ρ**	**Fleiss' κ**
cooking	0.585	0.312
DIY	0.607	0.305
automotive	0.623	0.264
total	0.614	0.291

Table 6.7.: Inter-annotator agreement for 20 annotators (for annotations from 1 to 4).

6.3.3. Features for Classification

One important research question for the following experiments is to what degree attributes which are common to all closed compounds influence the prediction, in contrast to attributes that are related to termhood. We compute features tailored to represent these attributes:

1. Compound features:

- **Frequency** of
  - compound
  - head
  - modifier
- **Productivity** of
  - head

  - modifier

Note that for all but one of these features, we have a balanced set of compounds as a basis, because of the way how we selected the gold standard. For the termhood features, this is not the case, and imbalance might influence the classification processes:

2. Termhood features:

- contrastive measures:
  - **Weirdness ratio** (Ahmad et al., 1994)
  - **TFITF** – Term Frequency Inverse Term Frequency (Bonin et al., 2010b)
  - **CSvH** – Contrastive Selection via Heads (Basili et al., 2001)
- termhood measures that combine compound and termhood attributes:
  - **FGM-Score** (Nakagawa and Mori, 2003)

Finally, for the experiments with neural networks, we do not use hand-crafted features, but word embeddings. The word embeddings are trained for each domain, by using a concatenation of SdeWaC and the respective domain data as input data. We use the following two word embedding architectures:

4. Word embedding models:

- **word2vec** (Mikolov et al., 2013a)
- **fastText** (Bojanowski et al., 2017)

We use the word2vec model, because it is a standard model for natural language processing applications. The fastText model, which we introduced in background section 3.3.3 as well, works on character n-grams and not on words. As we already described, Bojanowski et al. (2017) states that it performs especially well on (German) closed compounds. This model is thus very interesting for us, because like that a compound embedding is learned partially from the same n-grams as its constituents. Thus, we implicitly have a representation of the constituents in the compound embedding, which should be beneficial for our classification task. Inspecting some words and their nearest neighbors for the two models confirms our intuition. When choosing *kochen* (“cooking”) we can obtain the following six most similar words by word2vec: *sieden* (“to boil”), *garen* (“to refine”), *brutzeln* (“to sizzle”), *braten* (“to

fry"), *grillen* ("to barbecue") and *zubereiten* ("to prepare"). For fastText, we find the following nearest neighbors: *erkochen* ("to reach by cooking"), *garkochen* ("to cook sth. well"), *teekochen* ("to make tea", we cite words in their original lowercased version as used in the model), *reiskochen* ("to cook rice"), *eierkochen* ("to cook eggs") and *bekochen* ("to cook for someone"). We find that the similarity in word2vec is more on the semantic level in contrast to fastText, where the words are highly similar on a surface level and thus usually morphologically related.

As a last feature, we compute **cosine distance between the modifier and the head**. The intuition behind that is that if constituents are more dissimilar, the compound is harder to understand (i.e. because the compound is more likely to be non-compositional then). An example for an easy compound with semantically related constituents would be *Kochtopf* ("cooking pot"), where both *kochen* ("cooking") and *Topf* ("pot") are cooking-related. In contrast to that, *Blindbacken* ("blind baking") consists of two constituents from different semantic areas and the compound requires more knowledge to be understood. We decided against a direct computation of compositionality, which can be computed by comparing compound and constituent vectors (Reddy et al., 2011; Schulte im Walde et al., 2013, 2016). The reason for this is that we balanced our dataset for frequency, which means that we will deal with a lot of infrequent compounds. For these compounds, we will not be able to train satisfactory word embeddings, and compositionality measures will be imprecise.

6.3.4. Predictive Models and Evaluation Procedure

We base the models on two specifications of the gold standard:

- **binary**: We simplify the annotation: We break down the four graded classes, which were labeled by the annotators, into two broader classes: *easy* and *difficult*. We decide to cluster classes 2,3 and 4 into a new class 2 'difficult' and keep class 1 as 'easy' for the following reasons: Annotators agree most for class 1 and it is the biggest class. A grouping like this would a) balance the class sizes more equally and b) we believe that annotators can easily recognize when they find a compound to be easy (because they fully understand it, which is why we get such a good agreement), but when it comes to difficulty, they have more problems to express to what degree they do not understand the compound (due to the fact that they cannot know how much they do not understand). This is why we find it a good simplification to distinguish between 'totally easy' and 'somehow difficult'.[12]

[12]Note that this binary distinction is a simplification in order to increase the sizes of the classes to predict.

- **four-class**: We decided for taking the median instead of the mean of the annotators' ratings, to be less sensitive to outliers. In case of being between values, we decided for the upper median (if the value is .5, it is rounded).

For classification, we use two models: a) **decision tree classifiers**, with relying on Gini impurity as the branch splitting criterium, and b) **feed-forward neural networks**. The neural architectures consist of an input layer, an embedding layer, an optional hidden layer and an output layer. Hidden layer size is 32, we train with 50 epochs and a batch size of 32.

As a comparison to all models, we use a most-frequent class **baseline**.

When testing for **significance**, we use the McNemar's significance test (McNemar, 1947), a paired non-parametric statistical hypothesis test.

For **evaluation**, we use 5-fold cross validation and the Micro- and Macro-F1 score.

6.3.5. Results and Evaluation

Decision Trees

All features. As a first step, we find the optimal tree depth of the decision trees, because the models can easily overfit without pruning. For that, decision trees are computed with constantly enlarging the depth, until results decrease. Like that, we find an optimal depth of 3 for the decision tree for the binary task, and an optimal depth of 5 for the decision tree for the four-class task for all features[13].

Table 6.8 shows the results for the decision tree classification with all features. The results for 'All', denoting the whole dataset, is marked in bold. The model significantly improves over the baseline, both for the binary and the four-class task. The picture is different for the individual domains: While the classification is better for Cooking and DIY than for automotive in the four-class task, it is not significantly better than the baseline. The better results thus might be due to a higher imbalance of classes for cooking and DIY. For the binary task, all models perform significantly better as the baseline, where automotive reaches the best results across the domains.

For compounds, there should exist at least three difficulty levels; next to 'completely easy' and 'completely difficult' there should be 'partially easy/difficult' compounds, where the compound consists of an easy and a difficult constituent.

[13]The optimal depth varies for the domain subsets, cooking, DIY and automotive.

GS	Binary		Four-class	
Measure	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Baseline Cooking	0.519	0.342	0.498	0.166
Baseline DIY	0.584	0.369	0.407	0.145
Baseline Automotive	0.667	0.400	0.325	0.123
Baseline All	0.604	0.377	0.376	0.137
Cooking	0.646	0.631	0.543*	0.312*
DIY	0.712	0.684	0.519*	0.406*
Automotive	0.750	0.720	0.471	0.286
All	**0.732**	**0.707**	**0.492**	**0.405**

Table 6.8.: Results for classification using all features. Results marked with an asterisk do not significantly improve over the baseline.

Feature groups. For the next experiment, we distinguish the features by groups with the following characteristics[14]:

1. Domain-specific corpus-related features:
   - Frequencies of compound, head and modifier, productivity of the head and modifier, FGM-Score
2. General-language corpus-related features:
   - Frequencies of compound, head and modifier, productivity of the head and modifier, FGM-Score
3. Contrastive features:
   - Weirdness score of compound, head and modifier, TFITF of the compound, head and modifier, CSvH
4. Cosine distance features:
   - Cosine similarity of word2vec-vectors, cosine similarity of fastText-vectors
5. Compound features:
   - Frequency in domain-specific corpus, frequency in general-language corpus, number of constituents, weirdness score of compound, TFITF of the compound

[14]This list of characteristics can be found in Bettinger (2019, 47f.)

6. Constituent features:
    - Frequencies of the head in the domain-specific corpus and the general-language corpus, Frequencies of the modifier in the domain-specific corpus and the general-language corpus, productivity of the head in the domain-specific corpus and the general-language corpus, productivity of the modifier in the domain-specific corpus and the general-language corpus, weirdness score of the head and modifier, TFITF of the head and modifier, CSvH
7. Head features:
    - Frequencies of the head in the domain-specific corpus and the general-language corpus, productivity of the head in the domain-specific corpus and the general-language corpus, weirdness score of the head, TFITF of the head, CSvH
8. Modifier features:
    - Frequencies of the modifier in the domain-specific corpus and the general-language corpus, productivity of the modifier in the domain-specific corpus and the general-language corpus, weirdness score of the modifier, TFITF of the modifier

Feature Group	Micro-F1	Macro-F1
Baseline	0.604	0.377
Cosine	0.594*	0.391*
Head	0.608*	0.568*
Domain	0.635*	0.593*
Modifier	0.656	0.627
Constituent	0.661	0.648
Contrastive	0.713	0.690
All	**0.732**	**0.707**
General	0.735	0.703
Compound	0.736	0.713

Table 6.9.: Binary.

Feature Group	Micro-F1	Macro-F1
Baseline	0.376	0.137
Cosine	0.400*	0.258*
Domain	0.405*	0.300*
Head	0.418	0.287
Constituent	0.455	0.364
Modifier	0.457	0.370
General	0.458	0.359
Compound	0.480	0.342
All	**0.492**	**0.405**
Contrastive	0.510	0.408

Table 6.10.: Four-class.

significant improvement

For tables 6.9 and 6.10, the results for 'All' are marked in bold again. The tables show that results are worse with using all features, in contrast to using a subgroup of features; 'All' is not the the best class. The categories 'Cosine', 'Domain' and 'Head' perform worst, and do in most cases not even significantly improve over the baseline. The modifier features give a better result than the head features - this is in line with the results from Hätty et al. (2017a) (section 5.2), where the modifier features are more important for detecting termhood than head

features. For both the binary and the four-class task, the groups 'general', 'compound' and 'contrastive' perform best. For the binary task, the compound features perform best, while for the four-class task the contrastive features perform best.

Individual features. Tables 6.11 and 6.12 show results for all individual features which perform significantly better than the baseline (the results are sorted). For the four-class task, there are three more features that perform significantly better than the baseline than for the binary task; these features are marked in bold. One can see that the best individual features are the same for both tasks (except for the three additional features for the four-class task). The individual features are ranked in the same way, only the results for the general-language frequency of the compound ($freq\_gen$) and the general-language FGM-score (fgm_gen) are inverted. The first three ranks are allocated to features which address distinct attributes of a compound term candidate: a compound's general-language frequency, a termhood measure involving constituents (fgm_gen), and a contrastive termhood measure ($comp\_weird$).

Feature	Micro-F1	Macro-F1
Baseline	0.604	0.377
Comp_tfitf	0.637	0.566
Freq_head_gen	0.642	0.571
Freq_mod_gen	0.645	0.619
Prod_mod_gen	0.653	0.616
Comp_weird	0.709	0.690
Fgm_gen	0.713	0.696
Freq_gen	0.732	0.706

Table 6.11.: Binary: Single features, which are significantly better than the baseline.

Feature	Micro-F1	Macro-F1
Baseline	0.376	0.137
Comp_tfitf	0.412	0.238
Freq_mod_dom	0.415	0.280
Num_comp	0.417	0.248
Prod_head_gen	0.426	0.306
Freq_head_gen	0.435	0.290
Freq_mod_gen	0.454	0.322
Prod_mod_gen	0.455	0.298
Comp_weird	0.462	0.330
Freq_gen	0.464	0.343
Fgm_gen	0.467	0.339

Table 6.12.: Four-class: Single features, which are significantly better than the baseline.

Best feature combination. Next, we analyze the results of how features work together in a decision tree. For that, we perform feature selection: We start with the best performing individual feature for each task (see previous paragraph). In each iteration, we add the feature that gives us the best Micro-F1-Score together with the previously added features. We stop when results stagnate or decrease. The results are shown in tables 6.13 and 6.14. The best performing individual feature is marked in bold. Combining four features already gives us

Chosen Feature	Micro-F1	Macro-F1
+Freq_gen	**0.732**	**0.706**
+Prod_mod_dom	0.739	0.720
+Prod_mod_gen	0.744	0.725
+Mod_weird	0.746	0.727
+Freq_dom	0.746	0.727

Table 6.13.: Binary: Add-one-feature (depth 3).

Chosen Feature	Micro-F1	Macro-F1
+Fgm_gen	**0.467**	**0.339**
+Head_tfitf	0.487	0.350
+Prod_mod_gen	0.493	0.362
+Prod_head_gen	0.511	0.370
+Num_comp	0.511	0.370

Table 6.14.: Four-class: Add-one-feature (depth 5).

the best results for each task. Both for the binary and the four-class task, we find that a feature addressing attributes of the whole compound is complemented with features addressing constituent attributes.

Analyzing 'low' and 'high'. For creating the gold standard compound dataset, we distinguished between tertiles for every feature, for example the tertiles 'low', 'mid' and 'high' modifier productivity. The tertiles are computed by sorting all elements for one feature, and cutting the data into three equally-sized portions. The resulting frequency ranges are shown in table 6.15. In this paragraph, we compare the classifier results for the two extreme tertiles, 'low' and 'high', for three important features: frequency of the compound and productivity of modifier and head (note: some of the features are the ones that the dataset was balanced for). Each feature is evaluated once for the general-language corpus and once for the domain. The results are shown in table 6.16. For every line, the higher value of 'high' and 'low' is marked in bold. It is obvious that better results are achieved in the 'low'-category, across all features. The gap between the results for 'high' and 'low' is especially high for the productivity of modifier and head. Thus low productivity represents a rather clear indicator for a compound to be either easy or difficult (given that the model achieves better results in the prediction), while high productivity is an attribute of harder to distinguish easy and difficult terms. In order to investigate this effect further, we inspect the gold label distribution in the 'low' and 'high'-categories. We find a dominance of difficult compounds in the 'low'-categories, while there is a higher balance between easy and difficult compounds in the 'high'-categories. This shows that low productivity and frequency are indicators of difficulty, while high productivity and frequency are less distinctive.

Feature	High	Mid	Low
Comp. freq. dom	8 - 444	4 - 8	3 - 4
Comp. freq. gen	17 - 53569	0 - 17	0
Prod. head. dom	62 - 1157	14 - 61	1 - 14
Prod. head. gen	786 - 8293	119 - 786	0 - 119
Prod. mod. dom	55 - 665	14 - 55	1 -14
Prod. mod. gen	590 - 4976	103 - 588	0 - 101

Table 6.15.: Feature ranges (Bettinger, 2019, p.57).

feature	**Micro-F1**	
domain	**low**	**high**
Freq_gen	**0.779**	0.722
Freq_dom	**0.773**	0.722
Prod_mod_gen	**0.884**	0.661
Prod_mod_dom	**0.863**	0.658
Prod_head_gen	**0.812**	0.693
Prod_head_dom	**0.802**	0.652

Table 6.16.: Results for *low* and *high* tertiles (binary case).

Feedforward Neural Networks

In a last experiment, we use pre-trained word embeddings as features for feed-forward neural networks, in order to see if we can replicate or overtrump previous results without using hand-crafted features. We use two kinds of network architectures:

- a *logistic regression* (LR), i.e. a network structure with only input and output layer, without a hidden layer
- a *multilayer perceptron* (MLP), i.e. a network with each one input, hidden and output layer.

For the binary classification task, the networks use a sigmoid activation in the output layer, for the four-class task the networks use softmax activation. For the multilayer perceptron, again a sigmoid activation is used for the hidden layer. The results are shown in tables 6.17 and 6.18, and best results across columns are marked in bold. We compare three different input settings for the classification tasks: The first model only takes the compound word embedding as input (denoted 'compound' in tables 6.17 and 6.18). For all settings, we distinguish between

NN with ...	network	word2vec		fastText	
		Micro-F1	Macro-F1	Micro-F1	Macro-F1
compound	LR	0.760	0.722	**0.746**	**0.724**
	MLP	0.761	0.729	0.738	0.720
comp+const	LR	**0.771**	**0.758**	0.734	0.715
	MLP	0.749	0.735	0.732	0.716
only const	LR	0.701	0.685	0.703	0.679
	MLP	0.714	0.697	0.713	0.696

Table 6.17.: Binary: neural network.

NN with ...	network	word2vec		fastText	
		Micro-F1	Macro-F1	Micro-F1	Macro-F1
compound	LR	0.514	0.385	0.459	0.338
	MLP	0.518	0.383	0.469	0.341
comp+const	LR	0.515	0.429	0.465	0.355
	MLP	**0.525**	**0.431**	**0.477**	**0.369**
only const	LR	0.460	0.362	0.447	0.355
	MLP	0.493	0.389	0.469	0.365

Table 6.18.: Four-class: neural network.

two differently trained word embeddings: the word-based word2vec and the character-based fastText word embedding models. The second model ('comp+const') takes the concatenated embeddings of the compound and of its constituents (binary split, i.e. two constituents) as input, to evaluate the impact of the constituents in contrast to the first model. The last model ('only const') only uses the concatenated constituent vectors, to evaluate if this information is competitive against the first model.

The results for the classifications are shown in tables 6.17 and 6.18. One can see that for the binary task, we reach the best results (marked in bold) using a combination of compound and constituent information with word2vec, and only using the compound embedding with fastText. This is an expected result. Since fastText embeddings are character-based, the constituents are implicitly covered as well. Using only the constituent information does not give us as good results as using compound information. This result is in line with the results of the previous section (section 6.2). The distribution of the results of the four-class task is similar to the binary task, except for that even for fastText the combination of compound and constituent information works best. This could result from the higher difficulty of the task. This is also indicated by the fact that for the four-class task, the MLP with the additional hidden layer pro-

duces the best results, while for the binary task the simpler model (LR) obtains best results. Another interesting result is that the models mostly perform better than models using fastText, although fastText implicitly contains constituent information. We trace this back to the fact that 171 infrequent compound vectors are missing for word2vec, because there is a minimum frequency threshold for word vectors to be trained. These 171 compounds are thus assigned to the same random vector. We hypothesize that infrequency is a good indicator for difficulty, and that the model can learn from the missing vectors which compounds are infrequent.

Although models using both compound and constituent information seem to be superior to models using only compound information, these results can only be treated as a tendency. For word2vec and both the binary and the four-class task, models using both compound and constituent embeddings are not significantly better than models using only compound embeddings. However, although models using compound embeddings perform significantly better than models using only constituent embeddings (which is intuitive), the latter models perform significantly better than baseline. This shows that constituent embeddings carry informative characteristics for classifying compounds for difficulty.

6.3.6. Conclusion

This section described experiments for predicting the difficulty of German closed compounds occurring in specialized domains. We dealt mainly with two questions: First, how do constituents influence the difficulty of a compound. Secondly, how do termhood and domain attributes on the one hand, and compound formation attributes on the other hand influence this difficulty. In order to train predictive models, we first designed an annotation study on the basis of a dataset with a total of 1030 compounds for three domains. We reached a Spearman's rho correlation of 0.616 on average for 20 annotators. The subsequent classification experiments were based on two settings of the gold standard dataset; a binary version, where the annotations were broadly separated into 'easy' and 'difficult'; secondly, a four-class version with using the median of the actual annotations. We conducted two kinds of classification experiments: A decision tree classification using manually designed features characterizing termhood and compound formation, and neural networks using word embeddings. In summary, we found the following:

Compounds and Constituents. The binary task, as the presumably simpler task, reached better results with simpler means: General language frequency of the compound was a good indicator (2% better than the second-best feature for Micro-F1); in addition, there is a 5% gap between compound and constituent features (table 6.11), which shows that compound

features are sufficient for this task. For the four-class task, the distances between features are smoother, and the best results include compound and constituent information (table 6.12). However for both tasks we can see: a combination of compound and constituent features leads to best results (tables 6.13 and 6.14). The experiments with using neural networks show the same tendency: Using both compound and constituent vectors ('comp+const') gives best results, for the binary task in half of the cases the compound vector suffices (however, the improvement of 'comp+const' against only 'comp' is not significant).

Termhood Features. Contrastive features (i.e. termhood features) are more important for the four-class task than for the binary task (tables 6.9 and 6.10). For the four-class task, the contrastive features perform significantly better than the general-language features. This is especially notable since the 'freq_gen' feature is dominant for the binary task. In conclusion, for a broad difficulty distinction as for the binary task, general-language information might be enough, for the four-class task contrastive termhood features become more relevant.

Domains. We do not see any difference in the predictive power of the models for different domains. Presumably prediction differences lie in the sizes of the different classes. *Automotive* exhibits best Micro-F1 results for the binary task (table 6.8), but the baseline performs better as for the other domains as well. *Automotive* is the only domain with significant improvements over the baseline for the four-class task.

Low versus high productivity and frequency. We get clearest results for contrasting the lower and upper tertile value ranges for compound frequency and constituent productivity. Low productivity and frequency are clearly the better indicators for the level of difficulty. This seems counterintuitive: e.g. high frequency could be a reliable indicator for simplicity of a compound, while low frequency could indicate difficulty, but low frequency could also indicate that concepts are newly coined (which does not mean that they are difficult), or because of spelling or inflection errors. The dataset was cleaned for the latter, but the first case could happen. Concerning the productivity, the gap between 'high' and 'low' is even more extreme. We hypothesize that this could be due to that one difficult constituent would make a compound difficult, but an easy compound would need all constituents to be easy. For example, *Wankel* in *Wankelmotor* 'Wankel engine' (automotive) is the name of the engine (or of its inventor) and is little productive in comparison to the well understood and highly productive *Motor* 'engine'. *Wankel* makes the compound difficult to some degree[15]. This is why single easy constituents

[15] Be reminded that for the binary classification, which we performed for the tertile experiment, we separated the annotations into 'completely easy' and 'somehow difficult'.

might be no good indicators - it depends on the other constituent to be easy or difficult for the compound to be easy or difficult.

6.4. Predicting Degrees of Specificity across Domains

Assessing the specificity of the extracted terms is still a niche within the area of automatic term extraction. So far, studies for term specificity are mostly restricted to medical terminology and relate to the communication between doctors and patients. But clearly, the communication between experts and lay people is relevant across specialized domains, and term specificity prediction is important for a range of tasks such as automatic thesaurus creation, assessing text specialization, and domain knowledge acquisition. Above all, predicting specificity can be considered a more fine-grained and expressive form of terminology extraction.

In this section, we first semi-automatically collect German specialized domain corpora to create a gold standard of term specificity across four domains: automotive, cooking, hunting and DIY ("do-it-yourself"). Based on a qualitative analysis of terminological phenomena and variants of ambiguity across domain-specific and general-language corpora, we then suggest two methods to explicitly integrate not only vector spaces of term occurrences but also vector space comparisons. In a first approach, we enrich the combined general-language and domain-specific word embeddings with a difference vector as input for a classification system. In a second approach we design a multi-channel feed-forward neural network with a Siamese network constituent to represent the vector comparison internally[16].

6.4.1. Specificity and Ambiguity

Ambiguity is not only relevant for the conventional automatic term extraction, but also plays an important role for term specificity prediction. This becomes clear by the overview that Ha and Hyland (2017) give about specificity (which they call technicality). According to Ha and Hyland (2017), there is no consensus among researchers about what exactly characterizes specificity. However, they observe two main categories. On the one hand, technical terms might have a narrow range of senses specific to the domain. They are only understood by a limited set of people, because they require knowledge about the domain. On the other hand, there are terms which are highly frequent in general-language usage as well. These terms are ambiguous, i.e., they carry specialized meanings within a particular domain, which are different from the general-language meanings.

[16]The work in this section is published in Hätty et al. (2020).

Corpus sizes	Preprocessed	Lemma:POS
Cooking	4.3 M	2.5 M
Automotive	4.9 M	2.3 M
DIY	4.0 M	2.1 M
Hunting	0.7 M	0.3 M
SdeWaC	778 M	326 M

Table 6.19.: Sizes of corpora. *Preprocessed* refers to the lemmatized corpus without punctuation, *Lemma:POS* to the version reduced to content words.

6.4.2. Data and Gold Standard Creation

Data. We use again the domain-specific corpus collections for automotive, cooking and DIY from the previous studies described in 6.3, but refined the domain-related filtering process for Wikipedia articles. We additionally create a corpus collection for the domain of hunting. We dealt with the domain of hunting previously in 4.2 and found it especially interesting because of its ambiguous terminology. We collect the data for this corpus from a hunting handbook[17] and hunting-related Wikipedia articles. As general-language reference corpus, we use again SdeWaC Faaß and Eckart (2013). In order that the preprocessing is consistent for all corpora, all corpora are lemmatized and POS-tagged with the TreeTagger (Schmid, 1995), and reduced to content words (nouns, verbs and adjectives). The corpus sizes are shown in Table 6.19.

Gold Standard. We select all words as term candidates with a minimum frequency of 10 in both the domain corpus and SdeWaC. Instead of relying on labor-intensive human annotations, we determine the specificity labels semi-automatically. First, we collect domain-specific glossaries for each domain, i.e. textual glosses and specialized terms with their meanings[18]. These glossaries contain terms which require domain knowledge (especially if they are ambiguous) and thus need to be explained to a lay person, i.e. they contain technical terms. Secondly, we collect thematic basic vocabulary lists (from thematic base vocabulary books, thematic vocabulary training lists for foreign apprentices, etc.). These lists contain the basic terminology of a domain, with a low level of specificity. Finally, we collect indices and tables of contents of domain-specific handbooks, which include all kinds of terminological vocabulary. We label the data as follows: As Ha and Hyland (2017), we see specificity as a continuum, but we adopt

[17] Ferdinand von Raesfeld. *Das deutsche Waidwerk. Lehr- und Handbuch der Jagd*. Paul Parey Verlag.
[18] c.f. `https://www.merriam-webster.com/dictionary/glossary`

	Cook.	**Hunt.**	**Auto.**	**DIY**
Tech. Terms	384	250	706	350
Basic Terms	853	186	236	250
Non-Terms	853	1176	5010	2962
Total	3045	1612	5952	3562

Table 6.20.: Size of Gold Standard.

a simplified handling and distinguish between three broad classes of specificity: specific terms (or technical terms), basic terms and non-terms.

1. **specific/technical term**: a word is contained in a glossary, but not in a basic vocabulary list

2. **basic term**: a word is contained in a basic vocabulary list, but not in a glossary

3. **non-term**: all other words, which do not overlap more than 4 characters with any term in the glossaries, the basic vocabulary lists, the indices and the table of contents

The resulting sizes of the gold standards per domain are presented in Table 6.20. Overall, our semi-automatic labeling method leads to 1,690 technical terms, 1,525 terms and 10,956 non-terms, a total of 14,171 term candidates.

To evaluate the quality of the gold standard, we randomly extract 30 words per domain and per system-assigned label (which leads to $30 \times 4 \times 3 = 360$ words in total). Together with three random context sentences, three annotators (including one of the authors) rated the labeling. We obtain an average Cohen's κ inter-annotator agreement of 0.50 and an average agreement with the gold standard of 0.47. This corresponds to "moderate" agreement, which we judge as sufficient for our gold standard, given that agreement in term annotation is considered a difficult task (Rigouts Terryn et al., 2019).

Note that although the gold standard contains only single-word terms, it does not only contain simplex terms. Since German is a compounding language, the gold standard also includes closed compounds.

Qualitative Analysis. We performed an in-depth analysis of our four domain corpora to identify the range of terminological phenomena and variants of ambiguity within and across general- and domain-specific data, in order to motivate and apply an appropriate model.

General language corpus	**Domain-specific corpus**
Die Knollen in Salzwasser fünf Minuten, die Deckel eine Minute **blanchieren**, dann warmstellen.	Den Lauch in einem Topf mit gesalzenem Wasser 15 Minuten **blanchieren** lassen.
Rindfleisch kurz in kochendem Wasser **blanchieren**.	Die Zwiebeln fein hobeln, kurz im kochenden Wasser **blanchieren** und kurz unter Kaltwasser abschrecken.
Die Würfelchen in Salzwasser weich **blanchieren** und anschließend warm stellen.	ca. 10 min in Salzwasser **blanchieren**.
Die Wirsingwürfel in kochendem Salzwasser mit etwas Natron **blanchieren**, abgießen und in Eiswasser abschrecken.	Den Spinat in das kochende Wasser geben und für ca. 2 min **blanchieren** lassen.

Table 6.21.: Example context sentences for *blanchieren* (cooking).

The automotive domain contains many compounds (such as *Antriebsschlupfregelung* 'traction slip control') and English words (*Frontairbags*). In the cooking and DIY corpora we find many complex verbs (such as *entgraten* 'deburr' for DIY and *abbinden* 'thicken (a sauce)' for cooking). Ambiguous terminology is an outstanding characteristic of the hunting domain, which contains many ambiguous expressions completely unknown by lay people, such as *Licht* 'light' as term for the eyes of game. With all those variations, it seems likely that surface form features will not be useful in a prediction task. Furthermore, frequency-based features might not be useful due to the high amount of ambiguity.

Regarding the two strands of technicality that Ha and Hyland (2017) observed, we further analyze the context sentences of annotated specific/technical terms in the general-language and the domain-specific corpus. We find examples that reflect the first category of technicality in Ha and Hyland (2017): there are specific/technical terms that seem to be rather monosemous and have a very restricted usage, like *Antriebsschlupfregelung* 'traction slip control' for automotive or *blanchieren* 'blanch' for cooking. For example, *Antriebsschlupfregelung* often co-occurs with *Antiblockiersystem* 'anti-skid system' and *blanchieren* with *Salzwasser* 'salted water' or 'water' in their domain-specific context sentences (see table 6.21 for example context sentences for *blanchieren*). Surprisingly, we find very similar domain-specific contexts in the general-language corpus, where we would not expect them. Since the general-language

General language corpus	Domain-specific corpus
Ich denke, mit **Zauberstab** kann man leichter zaubern.	1 Schneebesen, **Zauberstab** (Pürierstab) oder Handrührer mit Rührbesen.
Nicht vergessen soll er bitte seinen **Zauberstab** und es bleibt ihm freigestellt, ob er eine Eule, eine Katze oder eine Kröte mitbringt.	1 Mixgerät, Handrührer mit Mixstab oder **Zauberstab** mit Schüssel
Auf dem Planeten gibt es so genannte magische Höhlen, in denen sich die Upgrades für Fox ' **Zauberstab** befinden.	1 **Zauberstab** mit hohem Mixbecher oder Küchenmixer
Das Mädchen aber , wie es die Alte daherschreiten sah , verwandelte mit dem **Zauberstab** seinen Liebsten Roland in einen See, sich selbst aber in eine Ente, die mitten auf dem See schwamm.	Die Sauce abermals erhitzen, die Butter mit der Stopfleber zugeben und die Sauce mit einem **Zauberstab** schaumig aufmixen.
Mit Betten, **Licht** und einem Tisch.	**Lichter** ist die Bezeichnung für die Augen, die Ohren werden auch Lauscher genannt..
Trotzdem zögert Nikita, ob sie grünes **Licht** für den Mord an Karyn geben soll.	Auch bei schwachem **Licht** können sie noch sehr gut sehen.
Ich verließ die Bank und wanderte mit dem Blick gebannt auf den Mond, taumelnd, wie hypnotisiert, dem **Licht** entgegen.	Denn der Jäger hat nur dieses eine im Auge zu behalten, während er es [...] mit einer ganzen Anzahl von aufmerksamen **Lichtern** und Lauschern zu tun hat.
Darf man, kann man die intime Frage nach dem Heiligen ins grelle **Licht** der Laborlampen zerren?	Es gibt einige Vogelarten, bei denen sich die Geschlechter im für uns sichtbaren Licht nicht unterscheiden, wohl aber im UV-Licht.

Table 6.22.: Example context sentences for the ambiguous terms *Zauberstab* (cooking, upper table) and *Licht* (hunting, lower table). Sentences with a lime green background contain the target term in its general-language sense.

corpus is web-crawled, it however contains a certain amount of domain-specific texts as well; especially if a highly technical term is not ambiguous, it only contains such contexts. Consequently, we assume the general-language and domain-specific contexts to be maximally

similar in these cases.

The picture is different for ambiguous terminology, where sense distributions vary across corpora. For example, for the hunting term *Licht* 'light/eyes of game' we find both general and domain-specific meanings in the domain corpus (table 6.22, lower part); for the cooking term *Zauberstab* 'wand/hand blender' senses seem to be disjunctive across the corpora (table 6.22, upper part). These example reflect the second category of technicality described in Ha and Hyland (2017), the ambiguous terms that carry specialized meanings within a particular domain.

Based on our observations, we suggest an approach by Amjadian et al. (2016, 2018) as basis to detect degrees of specificity, since both general-language word embeddings and domain-specific word embeddings will encode termhood attributes. On top of that, we hypothesize that a comparison of the word vectors represents valuable information for a prediction system.

6.4.3. Models

Baselines. Existing studies on specificity typically rely on classical readability features such as frequency, term length, syllable count, the Dale-Chall readability formula and affixes (Zeng et al., 2005; Zeng-Treitler et al., 2008; Vydiswaran et al., 2014; Grabar et al., 2014). Therefore, we use a decision tree classifier (DT) as a baseline, with three standard features used for term familiarity prediction: frequency (corpus size normalized), word length and character n-grams. As a second baseline, we implement the approach by Amjadian et al. (2016, 2018) using a Multilayer Perceptron (MLP) and the concatenation of general-language word embeddings (GEN) and domain-specific word embeddings (SPEC) of a term candidate as input (MLP, GEN⊕SPEC), in comparison to using only one of the embeddings. We learn two separate word2vec SGNS vector spaces (Mikolov et al., 2013b) for GEN and SPEC.

Centering and Batch Normalization. Across neural models we apply batch normalization (Ioffe and Szegedy, 2015), which normalizes the output of a preceding activation layer by subtracting the batch mean and then dividing by the batch standard deviation. This reduces the effect of inhomogeneous input data, in our case the different domain corpora. We further length-normalize and apply element-wise column mean-centering to the embeddings, which is suggested as pre-processing step for a rotational alignment of vector spaces by Artetxe et al. (2016). The intuition behind centering is that it moves randomly similar embeddings further apart. Thus, we we consider it to be a beneficial pre-processing step for all embeddings in our task.

Comparative Embeddings and Multi-Channel Model. Simple vector concatenation does not incorporate any kind of comparison of the embeddings. We thus suggest two novel models to exploit general- vs. domain-specific comparisons: *Comparative Embeddings* (MLP, CON⊕DIFF) use a MLP classifier and add a difference vector to the input vector concatenation GEN⊕SPEC. Since the word embeddings were trained separately on different corpora, this model requires an alignment of the vector spaces. We use a state-of-the-art alignment method (Artetxe et al., 2016; Hazem and Morin, 2017), where the best rotation GW of a vector space G onto a vector space S is determined, with the rotation matrix W. W is computed as $W = UV^T$, with U and V retrieved from Singular Value Decomposition $S^TG = U\Sigma V^T$ (Schönemann, 1966). After the alignment, unit length is applied again (since the vectors do not have unit length anymore after alignment) and the absolute difference vector (DIFF) is computed. The concatenation vector GEN $\oplus$ SPEC $\oplus$ DIFF is then taken as input to the model. The model architecture is shown in Figure 6.4a).

As our second model, we use a *Multi-Channel Feed-Forward Neural Network* (MULTI-CHANNEL). The network takes the unaligned GEN and SPEC vectors as input, and processes each GEN and SPEC in a different channel. The third channel is a variant of a *Siamese network* (Chopra et al., 2005), which also represents a dual-channel network with shared weights. Both GEN and SPEC are processed through the shared weight layer, to map them onto the same space. Then the element-wise absolute difference is computed, and the output of all three channels in concatenated. The network is defined as:

$$h_1 = \sigma_1(W_1 * E(x_1) + b_1)$$
$$h_2 = \sigma_2(W_2 * E(x_2) + b_2)$$
$$h_{3a} = \sigma_3(W_3 * E(x_1) + b_3)$$
$$h_{3b} = \sigma_3(W_3 * E(x_2) + b_3)$$
$$d = |h_{3a} - h_{3b}|,\ d \in R^l$$
$$c = h_1||h_2||d,\ c \in R^{3l}$$
$$p = \textit{softmax}(c)$$

where x is a term candidate, and E(x) is the embedding layer, a function $E : x_i \rightarrow z_i$ that maps the word x_i onto its corresponding 300-dimensional vector z_i. W denotes the weight matrices, b the bias, σ the activation functions, and l denotes the sizes of the hidden layers. The model architecture is shown in Figure 6.4b).

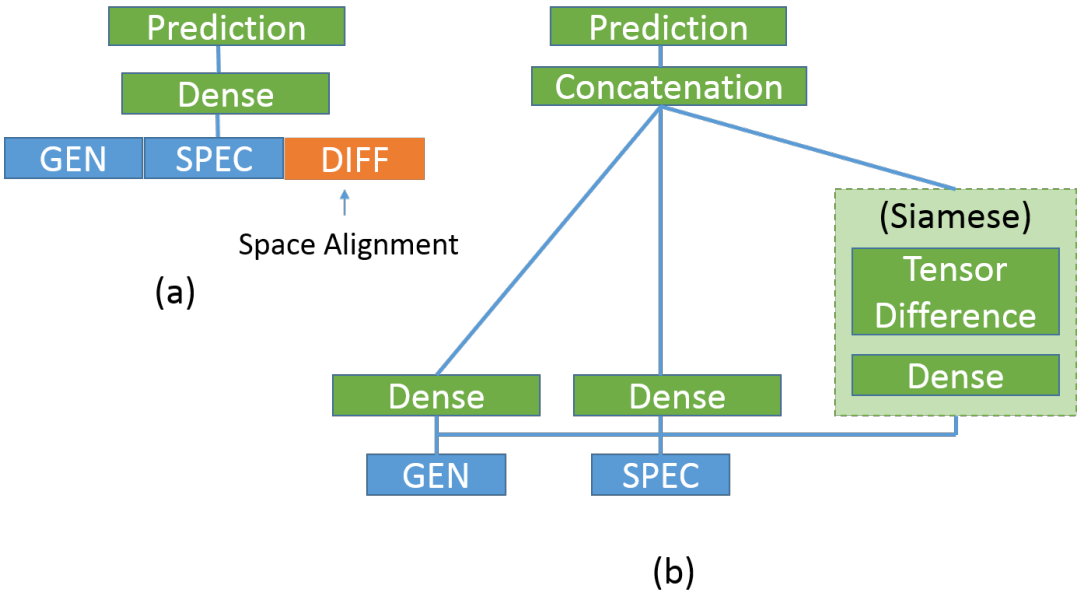


Figure 6.4.: Neural Architectures; a) MLP, CON⊕DIFF and b) MULTI-CHANNEL.

Training. We use SMOTE subsampling (Chawla et al., 2002) and train our network to minimize the cross-entropy loss, using back-propagation with stochastic gradient descent. We perform a randomized search for hyperparameter optimization for each model, i.e. subsampling parameter combinations. We test with the following parameters: hidden layers, epochs and batch size with values between 16 and 64, learning rate between 0.001 and 0.3, momentum between 0.0 and 0.9, and *tanh* or *rectified linear unit (ReLU)* as activation functions. To initialize the weights of the embedding layer, we use word2vec SGNS trained with a window size of 2, negative sampling with k=1 and subsampling with a threshold of $t = 0.001$. These parameter settings we chose here are based on the best-performing parameter settings obtained in the study on meaning variation between general-language and domain-specific language described in section 5.4. We do not train embedding layer parameters to maintain the original word meaning.

6.4.4. Results

We use Macro-Precision, Recall and F1-Score for evaluation, to put more weight on the correctness of the smaller classes *Base Term* and *Technical Term*. The results of our experiments are shown in Table 6.23. Using only a general-language vector GEN for classification performs

Method	P	R	F1
DT, basic features	0.56	0.58	0.57 (–)
MLP, SPEC	0.65	0.79	0.69 (0.62)
MLP, GEN	0.68	0.82	0.73 (0.72)
MLP, GEN⊕SPEC	0.76	0.89	0.81 (0.76)
MLP, CON⊕DIFF	0.84	0.94	0.88 (0.88)
MULTI-CHANNEL	**0.86**	**0.94**	**0.89 (0.85)**

Table 6.23.: Macro-Precision (P), Recall (R) and F1-Score results. All results use centering and batch normalization; results without centering are in brackets.

Method	Automotive	Cooking	DIY	Hunting
spec-vec	0.65	0.74	0.65	0.63
gen-vec	0.73	0.72	0.67	0.73
concat	0.80	0.82	0.76	0.80
diff-vec	0.88	0.90	0.85	0.85
siamese	0.90	0.90	0.87	0.86

Table 6.24.: Macro F1-Score for individual domains.

better than only using a domain-specific vector SPEC. This is most likely due to more training data and having both domain-specific and general-language parts in the general-language corpus. The models integrating a notion of vector comparison outperform the baselines, with the multi-channel networks performing best. Centering improves all results but MLP, CON⊕DIFF. This confirms that centering has an overall beneficial effect for our task.

The results for each domain separately are shown in table 6.24. We see an equal performance increase for each individual domain for each model improvement.

6.4.5. Conclusion

We semi-automatically created the first large-scale gold standard for specificity prediction and proposed two novel neural network models to fine-tune automatic terminology extraction by distinguishing between degrees of specificity. The models integrate general- vs. domain-specific comparative word embedding information in different ways; an adapted Siamese multi-channel network model performs best, and centering has an overall beneficial effect on pre-processing the vector spaces. Using both general- and domain-specific information is in

line with the motivation of contrastive term extraction approaches. Moreover, using word vectors gives us additional information about the meaning of a word. This is especially relevant for that the difference between monosemous and ambiguous terms can be more adequately represented.

6.5. Addendum: Evaluating Compound Splitting on Specific Domains

For the term specificity prediction experiments in this chapter closed German compounds were of great importance. This is due to the fact that German is a highly compounding language. We showed that it was beneficial for the experiments to have access to the constituents of a compound, i.e. sub-compounds or simplex words that the compound is composed of. Up to this point, we used compound splitting tools and even combined them to obtain better compound splits, but we did not tune the compound splitting tools on available domain data. If a pre-trained version of a compound splitting tool was available, or the original training data were attached to the tool, we relied on that. In the other case, we trained the compound splitter on our data. However, post-training a compound splitter on domain-specific data can possibly improve splitting results for compounds which predominantly occur in a specific domain. We give an example: the compound *Holzverbinder* ('wood connector') is likely to occur in DIY texts, and less in (randomly crawled) general-language texts. With the word statistics of general language, it is likely that the compound will be splitted into *Holz•Verb•Inder* (literally: 'wood'+'verb'+'Indian') instead of the correct split *Holz•Verbinder*, because the words 'verb' and 'Indian' occur more often in general-language texts than 'connector'. Thus, although improving compound splitting for specific domains is not a central aim of this thesis, it can be a useful preparatory step for experiments on compound termhood or compound term specificity.

In the light of the above, we present the following additional study on evaluating compound splitting for specific domains, in this case for DIY ('do it yourself'). We describe a post-training experiment, where we post-train the German compound splitting tool CompoST (Cap, 2014) on a DIY text corpus. We then split all noun compounds in the DIY corpus using the original and the modified splitter, and compare the results[19].

It is important to note here that the compound splitting evaluation study turned out to be rather complex, which is why we restricted it to only relying on CompoST. Thus, this study should be regarded as pointing the way to more research in this area, rather than a holistic

[19]The work in this section is published in Hätty et al. (2019a).

study on compound splitter evaluation for specific domains.

6.5.1. Related Work

There exist a variety of compound splitters, which rely on different methodologies. There are linguistically motivated splitters, that rely on word frequencies (Koehn and Knight, 2003; Cap, 2014; Weller-Di Marco, 2017). We already introduced CharSplit and SCS. Furthermore, a recent trend is to exploit distributional semantics to find the correct constituents (Ziering et al., 2016; Riedl and Biemann, 2016). Similarly, another splitter relies on semantic analogies (Daiber et al., 2015). Next to using different methodologies, the splitters return different splittings. For example, SCS can return a binary or an n-ary split, lemmatize and POS-tag the constituents. CharSplit, however, does only a binary splitting. The output might depend on the application the splitter was designed for; for example, CharSplit was designed to find the compound heads in order to facilitate co-reference resolution.
While a number of well-known and some upcoming tools for splitting German compounds exist, we are not aware of recent activities towards the comparative evaluation of the output quality of such tools; Escartín (2014) conducts a small comparative study with two compound splitters. In addition, there is little work on domain adaptation of compound splitters. Macken and Tezcan (2018) perform Dutch compound splitting, and adapt the splitter to the automotive and the medical domain. They find that only using general language leads to better results than only using domain-specific data, but a combination of both leads to the best results.

6.5.2. Post-training CompoST with Domain-specific Text Data

Adapting a compound splitter to a certain domain of interest, as DIY in our case, might improve the compound splitting for two reasons: First, the domain-specific constituents of a compound might be infrequent in general language, and that is why the correct split or base form of the constituent cannot be found. For example, the compound *Eloxierverfahren* ("anodizing procedure") should be splitted and lemmatized to *eloxieren•Verfahren* ("to anodize•procedure"). Secondly, splitting probabilities might be skewed because a certain split is more likely in general language, while another one is more likely within the domain (as illustrated with the *Holzverbinder* example above).

However, post-training of a compound splitter on a domain-specific corpus is not always possible. It depends on the design of the tool and if the original training data are available for updating.

We adapt the splitter CompoST. CompoST relies on frequency counts derived from a corpus, in the default case a general-language corpus. CompoST allows different parameters and therefore different versions of output (for instance, it can split a word even when frequency scores suggest that the word cannot or should not be split (forced splitting), or it can split only nouns). In our experiment, we do not apply forced splitting. One of the drawbacks of the tool is that words unknown to SMOR cannot be split, as well as disambiguation of possible splits happens entirely frequency based, and this might lead to inconsistencies on a non-lemmatized word list. To adapt the splitter to the DIY domain, we compute all the frequency counts for a DIY text corpus. Then we either add the frequencies to existing token entries, or create new ones. We use a domain-specific DIY corpus with 5.6 million words. The texts were collected from different sources, but all of them are DIY-related. There are texts produced by domain experts as well as by interested lay users, such as encyclopedia texts, DIY-instructions and manuals. Preprocessing has been done with SpaCy[20] (Honnibal and Johnson, 2015). Working with the German language model of SpaCy, we make use of the tokenizer, the POS-tagger and the lemmatizer. While the tagging itself is based on a convolutional neural network, the lemmatizer still works with a conservative look-up table. We use the POS-tags to select noun compounds as candidates for compound splitting.

6.5.3. Results and Evaluation

For the evaluation of post-training CompoST, we take all word types from the DIY corpus as candidates for compound splitting, which are tagged as nouns. We both run the original CompoST (ORIG) and the version of CompoST adapted to the DIY domain (MOD). The results are shown in table 6.25. Overall, the modified version of CompoST finds more compounds than original CompoST does (first two rows of table). However, the difference is not big (259 compounds). Furthermore, for the majority of the cases, both splitter versions split identically (row 3), i.e. roughly 95% of the compounds split by MOD are split in the same way by ORIG. Rows 4 to 9 show the cases where the splitters do not agree, which is further analyzed below.

Only MOD splits vs. only ORIG splits. MOD splits more compounds than ORIG. In return, it misses compounds which were originally split ("only ORIG splits"). This makes up roughly 2/3 of the size of the compounds only split by MOD. It seems likely that the missed compounds originate from general language, and the newly split ones are domain-specific. However, when analyzing the compounds, this is not the case; clear DIY-compounds like

[20] https://spacy.io/

feature	#
all ORIG splits	59.936
all MOD splits	60.195
same split	57.145
only MOD splits	640
only ORIG splits	411
MOD more splits	232
ORIG more splits	227
different split points	127
lower/upper difference	1.793

Table 6.25.: Comparison of the splitting results for the original CompoST (ORIG) and CompoST post-trained on a DIY corpus (MOD).

Akkuschrauber ("screwdriver')', *Stichsäge* ("padsaw") or *Heimwerker* ("DIYer") are not split by MOD.

Secondly, we want to analyze the impact of hyphenated compound candidates. An example would be *Douglasien-Bodendielen* ("douglas fir-floor boards"), where the split point is obvious because of the hyphen. There are rare cases where such a split would be wrong, e.g. *3-in-1* or *200-er*. We throw out all compounds where the split point is set at the hyphen and show the result in table 6.26 (columns "only X splits"). Obviously, most compounds that MOD missed were hyphenated compounds; for closed compounds, MOD shows a superior performance for both binary and ternary compounds.

	only X splits		**X more splits**	
	ORIG	MOD	ORIG	MOD
binary	43	600	-	-
ternary	0	50	137	22
nary	-	-	9	0

Table 6.26.: Difference of splitting results for the original CompoST (ORIG) and post-trained CompoST (MOD) with disregarding all compounds with splits at hyphens.

MOD more splits vs. ORIG more splits. In these cases, both splitters split the same compound but the number of splits is different. While for the overall results (table 6.25) this part seems to be rather equally sized for the splitters, focusing on the closed, not hyphen-

ated compounds again (table 6.26, columns "X more splits") the picture is quite different. MOD produces fewer splits, i.e. contracts constituents within a compound. For example, ORIG splits *Schraubendreherklingen* ("screwdriver blades") as *Schraube•Dreher•Klingen* ("screw•driver•blades"), while MOD splits *Schraubendreher•Klingen* ("screwdriver•blades"). We conclude that MOD finds some compounds to occur frequently and thus does not split them anymore. This intuition also coincides with the results from the previous paragraph, that DIY compounds like *Akkuschrauber* ("screwdriver") are not split anymore by MOD.

Different split points. In these cases, both splitters split the same compound and return the same number of splits, but the split points are differently set. When analyzing the compounds, we find that in most cases the results are different because the modifier is either lemmatized as noun or verb, e.g. *Putz/putzen* ("plastering/to clean"), or the lemma is different: *Dosen* → *Dose/Dosis*. Some errors result from the Fugen-s (*Prozessor•Steuerung* "processor controlling" vs. ??*Prozessor•Teuerung*, lit.: "processor increase in prices"), or a completely wrong split. MOD performs superiorly to ORIG because it always selects the more likely lemma in the domain (e.g. *Putz* instead of *putzen*). We randomly select 30 compounds of this category and compare the splitting results; MOD splits 18 times correctly, ORIG only 8 times (in the other cases, both splits were incorrect).

Lower/upper difference. In these cases, both splitters split the same compound, return the same number of splits and find the same split points. Only upper- and the lowercasing is different. When analyzing the respective compound splits, one can see that it is mostly again the modifier which is different. Sometimes this is a discrepancy between verb and nominalized verb (e.g. *Sägetisch* "sawing table" is either split as *sägen•Tisch* "to saw•table" or *Sägen•Tisch* "sawing•table"), or upper- or lowercasing is just wrong (e.g. *Nahtkontrolle* is split as *naht•Kontrolle* "joint examination"). It is unclear where this effect comes from. When again extracting 30 compounds randomly, MOD lemmatizes 15 times correctly, and ORIG lemmatizes 14 times correctly. To conclude, no splitter shows superior performance here.

6.5.4. Discussion

For this study, we relied only on one compound splitting tool instead of comparing several splitting tools. In general, it is rather difficult to compare and evaluate the performance of different compound splitters. They return diverse splittings, e.g. they either return binary or n-ary splits, lemmatize the results or additionally POS-tag them. For some splitters, there even are several settings available (as for example, restricting either to a binary split or allowing

an n-ary split). Thus, sometimes a comparison can be hard. For example, do we prefer a splitter that does not lemmatize against a splitter that lemmatizes, but sometimes returns wrong lemmas? Finally, the follow-up task for the compound splitting might decide which splitter we will use.

6.5.5. Conclusion and Outlook

We presented a small study to evaluate the performance of a German compound splitting tool, CompoST, when being adapted to a specific domain. For that, we post-trained CompoST on domain-specific DIY data, and compared the results for splitting domain-specific compounds. We found that for roughly 95% of the compound candidates, the original and the modified splitter return identical splits. For the rest of the compounds, we performed a detailed evaluation with respect to several features, like the number of splits or a difference of the exact split points. We find that in these cases the adapted CompoST mostly outperforms the original one, especially for binary and ternary closed compounds. This qualitative improvement is quantitatively watered down by the fact that the original CompoST more often splits hyphenated compound candidates than the post-trained version. The modified version more often contracts constituents within an n-ary compound, presumably due to the increased number of occurrences of a complex constituent (e.g. *Heimwerker*) in the data used for post-training.

Overall, a possible comparison of compound splitters proved to be more difficult than one would expect, as the tools come with widely diverging features: some tools only provide one split-point, others do not come with training data, yet others include lemmatization of the output, which in some cases can be a source of further errors. Against this background, we see a need for further detailed methodological work on the topic.

6.6. Summary

This chapter dealt with experiments for extended automatic term models, all including some notion of term specificity or difficulty. Extended automatic term models can be seen as an extension to 'conventional' binary automatic term extraction. The two central problems of the previous chapter - complex terms and ambiguity - were treated again with regard to this problem. Insights and modeling solutions from the previous experiments were exploited for designing the experiments with extended term frameworks.

In section 6.2, an extended termhood model is designed where the termhood is defined by four ordered classes. We exploit the information given by constituent embeddings for

predicting compound termhood. We find that constituents, and especially the constituents' estimated termhood, are highly influential on the termhood prediction of the compound.

In section 6.3, the difficulty of compounds in specific domains is investigated. Automotive, DIY and cooking are selected as domains. A central question of the study is to investigate both the influence of termhood features and the influence of compound features for difficulty prediction. We find that pure general-language frequency is a strong indicator for domain-specific difficulty, but both contrastive and compound features become more influential the more granularly we distinguish between levels of difficulty.

In section 6.4, specificity is predicted for simple terms and compounds. We use a contrastive approach for the prediction, by exploiting both general-language and domain-specific word embeddings. We add a constituent to our neural network architecture, that aligns the vectors and computes a difference vector, in order to cope with possible meaning variation of the term candidates between general language and the domain.

In the addendum section 6.5, we evaluate the effect of post-training a compound splitter on specific domain texts, to improve the performance for splitting domain-specific compounds. In our experiment with CompoST, 95% of the splits remain identical, what the post-trained splitter performs superiorly according to our qualitative analysis.

7. Conclusion and Future Work

This concluding chapter is divided into three parts: In the first section, we summarize the thesis and elaborate on the main findings. This final summary allows us to compare findings across chapters so that we can emphasize the strongest findings that were consistent across experiments and experiment groups. In the second section, we conclude and briefly highlight the main contributions of this thesis. In the last section, we give an outlook on possible future work, which is based on the insights we gained while conducting the experiments.

7.1. Summary

This thesis was concerned with domain-specific terminology, i.e., vocabulary that can be associated with a specific domain such as automotive, cooking, DIY or hunting. The thesis dealt with theoretical and computational challenges for domain-specific terminology and automatic term extraction. The starting point for the theoretical considerations, which led to a refinement of the definition of 'term', was the need to create a human-annotated gold standard for the experiments on automatic term extraction. It soon became apparent that this was a notoriously difficult task, both because we wanted to work without a concrete follow-up task in mind, and we wanted to collect terms as holistically as possible. This led to a number of human annotation studies: a lay people study for the intuitive understanding of what constitutes a term, a semi-expert study with an annotation guideline refinement process, and finally, the evaluation of two term characteristics, centrality and specificity. In addition to deepening our understanding of what constitutes a term, the lay annotator study confirmed the reliability of lay people as participants; we often relied on the lay perspective for annotation, and the study showed that lay people have an intuitive understanding of terms.

By conducting the first two annotation studies, where annotators had to identify terms in context, we wanted to investigate two challenges annotators have to face: deciding on the correct linguistic form of a term candidate, and deciding if it is a term or not. Concerning the linguistic form, we found that annotators agree to a greater extent on closed compound terms than on other forms of multi-word terms. We concluded that for closed compounds, the advan-

tages of both multi-word and single-word terms are combined: they have clear demarcations but are less general than single-word terms. Besides that closed compounds are a frequent phenomenon in German, this is another reason why we investigated automatic term extraction for closed compounds more intensively in the successive studies.

Concerning the question when to decide for a term or for a non-term, the lay people study showed that this binary distinction is insufficient. The study revealed that lay people perceive gradual differences between terms, which is why we tested scalar instead of binary annotations further on. As a another explorative study, we tested for two attributes that both seem to characterize terms intuitively[1]: centrality (topical relevance) and specificity (symptomatically, the level of difficulty for lay people). As a result, annotators agreed to a greater extent on specificity than on centrality. Annotators struggled with pinning down centrality, and we also could not resolve the critical points in discussions afterwards. In the end, we feel that the problem here is a clash of different intuitions about what constitutes centrality; for example, a broad usage in the domain or the lack of a term's relevance in another domain. In contrast to that, for specificity, annotators can intuitively rely on how difficult it is to understand a term candidate.

Altogether, our studies showed that a more fine-grained term annotation represents intuitive perceptions of terms, and that especially specificity is a concept that is intuitive for annotators. Experiments which break down such a fine-grained perception of terms into ordinal scales for gold standard evaluation gave us satisfying inter-annotator agreement for gold standard creation, and also gave us the possibility to computationally explore these extended term definitions. We conclude that ordinal scales should be preferred for term annotation, since they better represent the multi-faceted character of terms, and terms are characterized in a way that they can be re-used more easily in follow-up tasks because subsets can be selected.

As a result from the previous findings, the following computational modeling experiments were conducted for a) conventional binary automatic term extraction and b) term extraction with underlying extended term frameworks, where the latter mainly focused on specificity. We addressed term complexity and term ambiguity for these two tasks since they are two major challenges. We used hand-crafted term features and vector space models as the basis for automatic term extraction models. By addressing term complexity and term ambiguity for these two tasks, findings could be analyzed for similar tendencies.

For term complexity, the primary focus of the experiments was to investigate the influence of constituents of complex terms. Constituents were investigated for different attributes

[1] as found by other research as well, e.g. Pearson (1998) or recently Rigouts Terryn et al. (2018, 2019)

throughout the studies: their potential to be a term themselves (this was the focus in all studies), but also their position in the complex term and other beneficial attributes to recognize compound terms, like their role in compound formation (represented as features such as productivity) or their semantics in general (represented as word embeddings). We applied two kinds of models, decision tree classifiers using a rich feature set, where we could get insights into the prediction process, and neural networks using word embeddings as input. Throughout the experiments, we found an influence of all these features on the correct prediction of a complex phrase. We achieved notably good results when the term extraction model exploited a constituent's likelihood to be a term; a) either by including it in the model as constituent termhood features or b) by adding auxiliary outputs to a feed-forward neural network for predicting constituent termhood in order to optimize the overall network and thus the prediction of the complex term. Regarding the position of a constituent in a complex term, we observe throughout studies that constituents in earlier positions (i.e., modifying constituents) tend to have a stronger impact on complex term prediction than those in end positions (i.e., heads). In the experiment, where we analyze this effect qualitatively, we find that clearly non-terminological modifiers indicate that the complex terms are non-terminological, as well. This means that even if the head is terminological, the effect of the non-terminological modifier dominantly influences the outcome of the system. Furthermore, even though constituent information is beneficial for improving term extraction systems, all experiments clearly show that constituent features and optimizations for constituents improve complex term prediction, but cannot substitute compound features. In other words, only relying on compound features gives us better results than only relying on constituent features, and using both of them tends to lead to the best results. This is especially interesting with regard to rare complex terms, where there is a lack of information about the compound.

Term ambiguity was the second problem that was addressed both for binary and extended term extraction. Since the approaches throughout this thesis were type-based and not token-based, the problem was not addressed with word sense disambiguation methodologies, but with predicting the strength of a term's meaning variation between general language and domain-specific language. In doing so, contrastive term extraction measures could be corrected, which build upon a comparison of a general-language and a domain-specific text collection. As general-language text collection, we used a huge web-crawled corpus; thus, our definition of 'general language' pragmatically relies on the text collection we had at our disposal. Nevertheless, such a corpus is a good choice since it contains a lot of diverse data. A three-part experiment was conducted to reach the goal of improving term extraction concerning the meaning variations. First, a framework originating from the field of diachronic

meaning change was applied to the task of annotating strengths of the meaning variations on a test set for term extraction. Then a large-scale comparison of state-of-the-art methods for detecting meaning change and meaning variation was conducted. Finally, the best-performing model predictions were incorporated into a contrastive term extraction measure to demonstrate that this corrects the errors for terms with meaning variation. In a qualitative posthoc analysis of predicted meaning variation over a broad set of words, we found that even the best method to predict meaning variation is prone to get biased by repetitions in the data, leading to wrong meaning variation predictions. However, the preciseness of the method for predicting the meaning variation is crucial here, since otherwise errors are introduced while erasing other errors. Thus, in future, duplicate removal should be performed beforehand.

In a second step, we further conducted an experiment addressing an extended term definition, i.e., term specificity. We add a module to a state-of-the-art automatic term extraction system where meaning variation is computed dynamically, which improved extraction results.

All in all, experiments showed that injecting information about meaning variation into models for term extraction with binary and extended term definition improves the extraction results since the error that occurs due to considering wrong senses of a word is corrected.

7.2. Conclusion

The present thesis dealt with domain-specific terminology and automatic term extraction. The conducted experiments treated both theoretical aspects of terminology, as well as computational modeling of automatic term extraction systems. Contributions of the thesis covered new insights into term definition, as well as insights on challenges and advantageous attributes of term complexity and term ambiguity for automatic term extraction. We developed datasets for conventional and extended term definitions, and computational modeling solutions tackling term complexity and ambiguity.

Regarding term definition, we designed human annotation studies to investigate intuitive as well as directed (i.e., by providing term annotation guidelines) understanding of what constitutes a term. We further investigated two term characteristics, centrality and specificity, which resulted in new insights about what are the issues with term annotation. We found that perceptions of terms are granular instead of binary, and that centrality is the term attribute that is harder to agree on than specificity. We conclude that terminology should be annotated in a granular way in order to represent a term's multi-faceted nature.

For term complexity, we evaluated features that model the relationship between complex terms and constituents, and that comprise diverse termhood, compound formation and seman-

tic word embedding features. Throughout the studies, we found that constituent information is beneficial for a term extraction system, especially termhood features of constituents improve the term extraction systems. We also found a dominance of modifier features compared to head features.

Furthermore, inspired by the good performance of termhood features for binary automatic term extraction, we exploited constituent information for automatic term extraction with an extended, fine-grained term definition. We designed the first term extraction system that uses auxiliary output layers to predict the term likelihood of a constituent. The predictions for the constituents are disregarded later on, but the training process gets optimized, and this led to better results for the complex terms predictions. We thus implicitly validated the previous finding that a constituent's likelihood to be a term, in particular, influences the complex term's likelihood to be a term.

For term ambiguity, we contributed a new representation of ambiguity as meaning variation between general and domain-specific language. We implemented a new annotation approach, originating from the field of diachronic semantic change, to collect human annotations to create a term variation gold standard. We performed an exhaustive evaluation of computational models to predict meaning variation. By comparing to a study with the same experimental design for diachronic semantic change and realizing a dual evaluation approach, we additionally proved the results to be stable for two semantic shift (or variation) tasks. Finally, we demonstrated that correcting term extraction measures for meaning variation improved automatic term extraction.

Additionally, we included meaning variation prediction into automatic term extraction models for extended term definition. We improved a state-of-the-art term extraction model by adding a model component that dynamically aligned word embeddings from general-language texts and domain-specific texts. In doing so, meaning variation between general and domain-specific language was represented and learned by the model.

In sum, the contributions of this thesis comprised theoretical considerations about term definitions, human annotation studies and new computational models. We also contributed empirical and qualitative findings for term definition and automatic term extraction challenges. Finally, there are the domain-specific corpora we crawled, cleaned and post-processed, the annotation guidelines that were designed, and the gold standards that were created, which are a basis for future work in this field.

7.3. Future Work

Centrality. We see the biggest need for further research in the area of defining terminology, so that it can be reliably annotated and is characterized granularly enough for being used in various follow-up tasks. As already mentioned, centrality, i.e. the topical association of a term candidate to its respective domain, is the crucial term characteristic that people disagree on. We believe that in addition to interviewing annotators directly for their annotation decisions and trying to refine annotation guidelines, the notion of what constitutes a domain should be reconsidered. A domain can consist of several sub-domains and sub-topics. On closer inspection, it is hard to decide which topics still belong to a domain. Most probably, this is the crux of the matter when it comes to a term's topical domain association. More empirical human annotation studies should address this mutual problem of a term's topical association to a domain, and a domain's composition and association to a term.

Annotator expertise level. Staying with the problem of annotating terminology more reliably, a second aspect would be to examine more deeply how the expertise level of the annotators influences centrality and specificity annotations. For example, one could imagine that lay annotators can better distinguish lower levels of specificity, which experts consider as non-specialized altogether. On the other hand, lay annotators might have difficulties with distinguishing higher levels of specificity, because they do not know the meanings of the words. It might thus be reasonable to balance annotators with different background knowledge.

Term ranking. For the annotation approaches for fine-grained term definitions, annotators were presented with an ordinal scale for issuing a rating. However, the more fine-grained the scale, the more difficult it usually is to distinguish between ratings. In addition, it is challenging to remain consistent throughout a long annotation study. We feel that conducting ranking instead of rating studies could solve the problem. So instead of asking "Rate specificity of expression X on a scale from 1-5.", it might be better to ask "Is expression X more specific than expression Y?". It is more difficult to set up this kind of annotation task. It might even require a dynamic annotation approach to reduce the complexity of the task. However, the result might improve a gold standard's quality.

Artificial datasets. Our research showed that constituent information is not enough, but that complex term information is more useful for automatic term extraction. It might be an exciting area of research to investigate further that issue, both on an annotation and extraction

level. What conditions have single constituents to fulfill to occur in a compound? Which combination of constituents is accepted? We propose to create an artificial dataset of closed compounds, where constituents are combined to build new phrases. For an extraction approach, we could then evaluate how well a system performs when only confronted with unseen terms.

Text corpora. In this thesis, we worked with a pragmatic approach for defining the notions of 'general language' and 'domain'. A domain was characterized by texts we get from thematically focused crawling, while general language was characterized by huge, multi-purpose corpora that are randomly crawled on the web. However, a closer look should be taken on what constitutes optimal general language and domain corpora. This consideration leads to further ideas to address automatic term extraction. Instead of taking reference corpora and improving the term extraction measures, the problem could be reversed, leaving the term extraction measure fixed and changing up the textual resources instead. This might give valuable insights into how such corpora need to be composed, and on a deeper level, it might also give answers to the question of what constitutes a domain or general language.

Combining term complexity and term ambiguity. So far, when it comes to automatic term extraction, the two phenomena of term complexity and ambiguity have mostly been analyzed in isolation of one another, with underlying term datasets often custom-tailored to the problem; we could not provide large annotation studies, and thus we needed to make sure that the phenomena under consideration definitely occur often enough in the datasets. The same applies to the computational models; they are usually custom-tailored to address either the problem of multi-words or ambiguity. As future work, datasets should be unified and enlarged, and model solutions for the individual phenomena should be integrated into one complete model. In this context, it would also be interesting if different model solutions add to each other in combination, or if they are partially redundant.

Token-level term extraction. Last but not least, the computational experiments in this thesis were all type-based and not token-based. That means, even if terms were extracted from context, we required them to occur more than once and only considered connected phrases (for the user studies, we did not enforce these requirements, and the linking of unconnected phrases was allowed). However, since interrupted phrases can occur and very flexible word combinations can be built for some domains (i.e. for cooking, many recipe ideas are new and thus require newly formed terms), leading to a high amount of hapax legomena, token-level term extraction should be addressed as well. While the theoretical considerations and

computational models from this thesis will still be applicable, other strategies will be needed on top of that; for example, dependency parses to identify disconnected parts of multi-words that belong together.

A. Supplementary Material

A.1. Annotation guidelines for section 4.2 (English translation)

A.1.1. Study: Annotation of domain-specific expressions. Instruction.

1) **Information about annotator:**

- age:
- sex:
- education/profession:

On a scale from 1 (low) to 10 (high), how well do you know the following domains?

DIY:
1 2 3 4 5 6 7 8 9 10
Cooking:
1 2 3 4 5 6 7 8 9 10
Hunting:
1 2 3 4 5 6 7 8 9 10
Chess:
1 2 3 4 5 6 7 8 9 10

2) **Domain-specific expressions:**

Domain-specific expressions are expressions that are special to a certain domain (i.e. DIY, hunting, ...) and thus characterize the domain.

3) **Task:**

For each domain (“Do-it-yourself”/DIY, cooking, hunting, chess), read the three text passages once in advance. Then go through the text again and mark what you would consider to be a domain-specific term. Tip:

Imagine you would have to remove all expressions which indicate to which domain the text can be associated. What would you remove?

4) **Instructions for annotation:**

- There is no restriction what can account for a domain-specific expression - neither for length nor for the part of speech (e.g. noun, verb, adjective,...)
- You have the following options for annotation:
  - With the label **TermBase** you can mark an arbitrary number of tokens, for example:
    - a word
    - several consecutive words
  - With the label **TermRel** you can make a connection between two character strings. To do that, click on the word and drag the mouse pointer to another character string (the direction does not matter). By doing that, the TermRel relation is created, which marks items that belong together.
- Not all words need to be linked. For a coordination, all parts are linked with TermRel.
- A link cannot be made across sentence boundaries.

Here is an example, how such an annotation could look like:

If you have questions, please write to [...]

A.1.2. [Alternative] Study: Annotation of domain-specific expressions for an index. Instruction.

[...]

2) **Index:**

An index is a list at the end of a book that lists the relevant expressions that occur in the book. In the given case the books is about one of our domains. The index is a collection of information ensuring that a reader can get a fast overview about the contents of the book.

[...]

Let us assume you would have created an index. Imagine you need to mark all expressions in text which should be linked to its entry in the index. Which expressions would you mark?

A.1.3. [Alternative] Study: Annotation of domain-specific expressions for specialized translation. Instruction.

[...]

2) **Specialized translation:**

Specialized translation describes the translation of a thematically limited text using scientific or technical language. In order not to confuse the reader of this potentially complicated texts, it is especially important to translate consistently on the one hand, but also to distinguish expressions depending on their contexts on the other hand.

[...]

Imagine you were a translator for Japanese and up to now you mostly translated novels. Now you get the task to translate texts originating from the domains above. For that reason, you engage an expert that knows the domains in both languages. Mark all expressions in text that you would ask the domain expert to translate beforehand.

A.1.4. [Alternative] Study: Annotation of domain-specific expressions for a glossary for specialized text. Instruction.

[...]

2) **Glossary for specialized text:**

A glossary for specialized or technical text lists the terminology of a specialized language or a technical specialized domain along with conceptually objective definitions, that make sure that a specialized expression is used and understood correctly. A glossary is a short variant of an encyclopedia.

(original source: https://de.wikipedia.org/wiki/Glossar)

That means: In a glossary all expressions are listed that have a specialized meaning in a domain, and which need to be distinguished from other expressions by giving a definition for that reason.

A.2. Annotation guidelines for section 4.3

A.2.1. Term Candidate Annotation – Guidelines for SWT

(1) Basic decision:

the noun (or noun group) or verb denotes a concept belonging to the domain of DIY activities? The answer to this question is often not yes/no, but more gradual, with a distinction between more central and more peripheral items. We are relatively

generous in accepting all central and many peripheral items. In case of doubt, annotate "unsure". In case that the phrase seems to be newly coined, annotate "ad-hoc".

(2) Domain-central ontological "classes" of terms (examples):

- physical objects:
  Brett, Holzleim, Schleifpapier, Bohrmaschine
- spare parts of tools:
  Sägeblatt, Bohrkrone, Fräsständer,...
- components of objects:
  Seite (des Bretts), Korpus (des Schranks),...
- materials:
  Leim, Wasser, Holz, Farbe,...
- attributes (property nouns):
  Durchmesser, Umriß, Holzmaserung, Stärke (des Schlauchs)
- named entities:
  IXO, Bosch,...
- "small objects" / consumables in a DIY process:
  Dübel, Nagel, Klebeband,...
- forms:
  Rechteck, Loch, Nut

(3) Typical nominals not belonging to the domain (examples):

- abstract nouns which do not denote properties of domain objects:
  Möglichkeit (but *Wasserundurchlässigkeit* would be a term)
- very general nouns: *Dinge, Sache, Objekt*
- nouns that are not only relevant for DIY, but (almost) everywhere:
  Platz, Preis, Kosten, Baum, Welt, Pflanze
- nouns denoting texts and reports, are only considered as terms, if they denote "DIY recipes"
  Testbericht, Video, Anleitung, Bauplan, Link

(4) Typical verbal items of terminological relevance (examples):

- actions carried out in DIY work:
  sägen, bohren, messen, schneiden, streichen
- general action verbs which are superordinates of typical DIY verbs:
  befestigen, entfernen, fuktionieren
- verbs denoting states of objects involved in DIY work:
  durchschimmern, verschmelzen, verharzen, einziehen (Lasur...)

(5) Verbs without terminological relevance (examples):

- mental activity verbs:
  überlegen, entscheiden, tüfteln...
- speech act verbs:
  sagen, empfehlen, befragen, plädieren
- general verbs of use/need:
  verwenden, brauchen, benutzen,...

(6) Context-dependence:

a) Most items are polysemous. We annotate tokens in context, not types. Thus, in

- *Stoß auf Stoß folgen die nächsten Tapetenbahnen*:*Stoß* is terminologically relevant.
- *Das Material federt Stöße kaum ab*, *Stoß* is NOT terminological.

b) inappropriate application: If some non-tool is misused as a tool, it is still out-of-domain

- "*mit einem Staßenbesen kann die Oberfläche aufgeraut werden*". *Staßenbesen* is out-of-domain.

(7) Borrowing from domains & neighboring domains:
In some cases it can be hard to decide if a term candidate is in-domain, or if it is associated to a related domain or is borrowed from another domain. Here some guiding principles:

- processes or chemical reactions of domain objects or material are not domain-relevant:
  - *Das Holz fault.*

- *Chemie- und Chlorkeulen zerstören recht schnell die Kunststoffoberflächen.*

- Candidates from domains that are necessarily needed in DIY activities are in-domain, neighboring domains are out-of-domain
  - in-domain: e.g. construction techniques and its basic elements (*Winkel, abmessen,* ...)
  - out-of-domain: e.g. sanitary and heating engineering, garden activities, ...

(8) Special items:

- Abbreviations: to be handled like their corresponding full forms;
- Unit names: dimension units are terms (mm, kg,...) but not "%" and other very general ones.
- Numbers: if they are part of the description of the object: *240-er Schleifpapier* (term); and otherwise (*600 ml Wasser, 6cm Länge*).
- Named entities: are terms if they denote objects/institutions of the domain:
  - Yes: BOSCH, IKEA, IXO, OBI, MDF
  - No: CDU, URL, EUR, USA, PDF
- Truncated coordinated compounds:
  to be marked in full, if second element is a term:
  - Yes: *Nut- und Federverbindung*, *Quell- und Schwindungsverhalten*,...

(9) Demarcations:

a) Demarcations have to be global to some extent, they are non-terminological if they only describe some entity in context or distinguish it within the local context
   - non-terminological: *vorderer Teil des Hobelkastens, Elektrowerkzeuge mit abweichenden Einsatzwerkzeugen*, ...)
   - terminological: *Bohrer mit Schlagwerk, akkubetriebenes Elektrowerkzeug*, ...

b) relative specifications: If too vague, ignore them
   - *Ich habe zwei kleine Laubholz-Stämme benutzt*. Only annotate *Laubholz-Stämme*

A.2.2. Practice of multiword annotation

(1) adjacent multiwords:

- annotated in one stretch:
  [*gehärteter Stahl* $_{domain}$]

(2) MWT interupted by irrelevant material:
[*diamantbesetzter* $_{domain}$], aber trotzdem preisgünstiger [*Bohrer* $_{domain}$]

The elements are related by means of an "mwt-glue link". Note that links are directed and go from modifiers (e.g. adjectives) to heads (e.g. nouns). This is also true of Adverb+Verb-combinations: *[Optimal] auf das zu bearbeitende Material [einstellen]* Thus link go:

- from adjectives to nouns
- from adverbs to adjectives
- from adverbs to verbs

We do not annotate verb+object relations.

(3) Idioms separated by non-idiomatic material:

- Standard case:
  wird...auf Gehrung gesägt:
  wird...[auf [Gehrung $_{dom}$]$_{Zusatz}$] [gesägt $_{dom}$]
- Separated case:
  auf Gehrung wird...gesägt
  [auf [Gehrung $_{dom}$] $_{Zusatz}$].....[gesägt $_{dom}$]
  With a link from the *Zusatz* to [gesägt $_{dom}$].
- Analogously: *auf Stoß....verlegt*

(4) Embedded terms

- Example 1: *freihand gebrochene gerade Kante*
  - [freihand gebrochene $_{dom}$] [gerade [Kante $_{dom}$] $_{dom}$]
    With a link from [freihand gebrochene $_{dom}$] to *Kante*;
    We intend to get from this the following term candidates:
    - Kante

        - gerade Kante
        - freihand gebrochene Kante
        - gebrochene Kante (?)
        - freihand gebrochene gerade Kante
    - Example 2: *der selbstnivellierende Kreuzlinienlaser Quigo*
        - We intend to get from this:
            - Quigo
            - Kreuzlinienlaser
            - selbstnivellierender Kreuzlinienlaser
        - If there is an additional adjective, as in *der kompakte, selbstnivellierende K.Q.* we annotate *kompakte* as a "*Zusatz*" with a link to *Kreuzlinienlaser*. So we get
            - kompakter Kreuzlinienlaser

(5) Insertions

- Relative clauses are not annotated
- Information in parentheses or in - ... - can be ignored:
    - e.g. *Akkubohrschrauber (geeignet für Holz), Dispersionsfarbe (dispers = fein verteilt), ...*
- Exception: another valid term can be constructed with the information in parentheses
    - *schwebender (schwibbender) Bogen* → terms (if applicable, created with links): *schwebender Bogen, schwibbender Bogen*
    - *akkubetriebene Elektrowerkzeuge (ohne Netzkabel)* → terms: *akkubetriebene Elektrowerkzeuge, Elektrowerkzeuge ohne Netzkabel)*

(6) Adjective Noun - MWTs

- adjectives to annotate in adj-noun pairs:
    - underspecified dimension adjectives (e.g. *präziser Schnitt, tiefer Schnitt, lange Kante*)
- do NOT annotate:

- evaluative adjectives (*handliche Fräse*)
- uninformative adjectives (*gängiger Handhobel*)
- adjectives related to the given situation (*vordere Schleifplatte*)

(6) Noun Preposition Noun - MWTs

- annotate complete phrase:
  - nominalized verb-object pairs (*Schleifen von Kanten*)
  - purpose constructions (*Sägeblätter für Porenbeton*)
- do NOT annotate as complete phrase (but maybe the nouns separately):
  - positional descriptions (*Querlöchern in Holzwerkstoffen*)

A.3. Annotation guidelines for section 4.4 (English translation)

A.3.1. Study about Ambiguity, Specificity and Centrality

Description of the problem and the task:

Term = expression, that characterizes a specific domain. E.g.: domain: *football/soccer*, term: *striker*

Attached you will find two lists with term candidates and context sentences for two different domains (**"DO-IT-YOURSELF"/DIY** and **COOKING & BAKING**). You are asked to work on the three tasks **after each other**. The tasks are:

1) **Specificity**: How important is expertise knowledge in the domain to understand the expression?

2) **Centrality**: How **prototypical/thematically central** is an expression for a given domain?

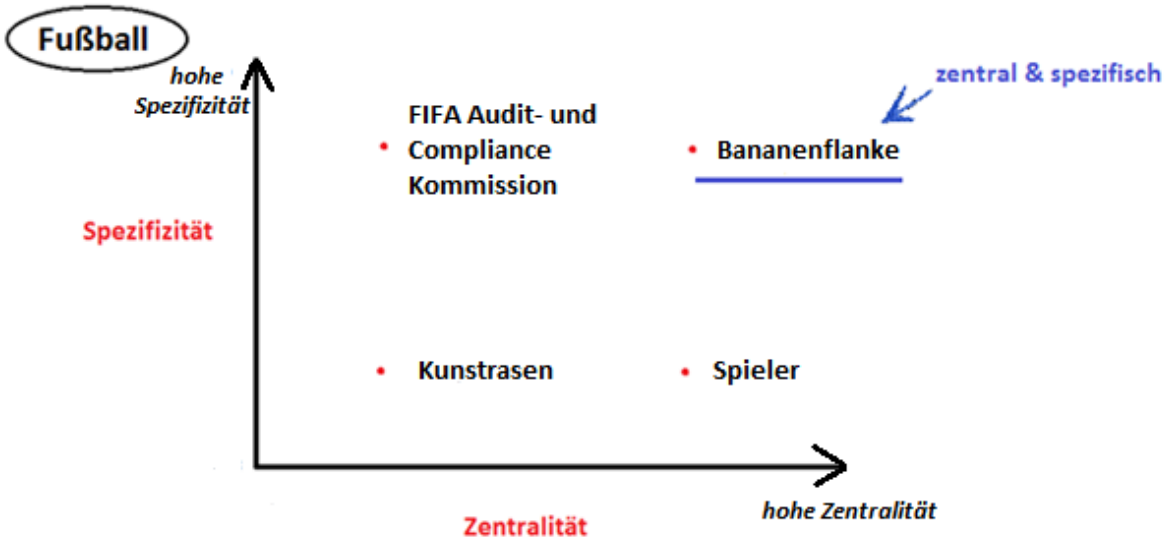
   ⇒ *Comment*: Only the strength of association to the domain is important, even if there are other domains to which the expression is associated (e.g. *"player"* is very central to FOOTBALL, but as well for CHESS, TENNIS, ...)

3) **Ambiguity**: How strongly is the given meaning of the expression (given in the context sentences) diverging from the meaning you know from general language?

⇒ *Comment*: Do only evaluate if an expression's meaning diverges from its general language meaning, but not from meanings in other domains. (in doubt, general language takes precedence of other domains - *solution* is known from general language, although it could also be associated to th domain of MATHS.

The **general language** represents the contrast to a **domain language** here. You will know general language expressions from daily life, even if they are no used frequently there.

Centrality and **Specificity** can be regarded as two dimensions which describe all terms of a domain. In the following, we give an example for the domain "FOOTBALL":



TODO:

After fulfilling the task, you will have six lists which are filled in. For each term candidate in the lists you are asked to rate on a scale of 1 (weak) and 6 (very strong). This means: weak to strong specific, weak to strong central, weak to strong ambiguous. If rating is not possible, rate 0. For each term candidate, you will find three context sentences per file.

[...]

A.4. Annotation guidelines for section 5.3 (English translation)

A.4.1. Annotation Guidelines for Usage Relatedness

Introduction. Your task is to rate the semantic relatedness between two uses of a word. For instance, presented with a sentence pair as in (A.1), you are asked to rate the semantic relatedness between the two uses of *passieren* in (A.1a) and (A.1b).

(A.1) a. Das Sieb auf einen zweiten Topf legen und mit der Schöpfkelle die Flüssigkeit durch das Sieb **passieren**.

b. Die Autobahn ist in diesem Bereich sechsspurig ausgebaut, was es Wildschweinen kaum möglich macht diese Barriere zu **passieren**.

Task Structure. You are provided with an ODS table document as shown in Table A.1. One row in the table corresponds to one target sentence pair. For each such row, the columns provide a 'target sentence 1' and a 'target sentence 2', illustrating the two uses of the same word and their contexts. The target word is marked in bold font in both contexts. For these pairs of target sentences, your task is to rate how related in meaning the two uses of the target word in the two target sentences are. Since language is often ambiguous, please read each sentence separately first, and decide upon the most plausible meaning of the target word in each sentence BEFORE comparing the two uses. In some cases, the target sentences provide sufficient information to understand the meanings of the target word; for more unclear cases, additional context is provided in gray.

	A	B	C	D
1	**Satz 1**	Bewertung	Kommentar	**Satz 2**
2	Aus dem **Ofen** nehmen und sofort mit Kristallzucker bestreuen oder die noch heißen Shortbreads in Kristallzucker wenden.			Hält man seine Hand in den **Ofen** fühlt sich die Luft im Backrohr warm an.
3	Er **rieb** seine schmerzende Wange.			Den Käse mit einer Reibe fein **reiben** und beiseite stellen.
4	Nun soviel von dem Gemisch aus Mayonnaise und Clotted Cream über die Eier geben, dass diese vollständig bedeckt (oder **"maskiert")** sind.			Viele Künstler **maskieren** sich nicht nur zu Fasching, sondern das ganze Jahr über.
5	Wenn Rex dann nicht **parierte**, schlug man ihn .			Die Leber waschen, **parieren,** auf gewünschte Größe portionieren, zwei Stunden in Milch einlegen, damit die Leber ausbluten kann und gewisse Bitterstoffe entzogen werden.

Table A.1.: Annotation table.

The judgment scale. The scale that you will be using for your judgments ranges from 1 (the two uses of the word have completely unrelated meanings) to 4 (the two uses of the word have identical meanings). This four-point scale is shown in detail in Table A.2.

4: Identical
3: Closely Related
2: Distantly Related
1: Unrelated

0: Cannot decide

Table A.2.: Four-point Scale of Relatedness.

Please try to ignore differences between the uses that do not impact their meaning. For example *isst* and *aß* can express the same meaning, even though one is in present tense, and the other is in past tense. Also, distinctions between singular and plural (as in *Karotte* vs. *Karotten*) are typically irrelevant for the meanings. Note that there are no right or wrong answers in this task, so please provide your subjective opinion. However, please try to be consistent in your judgments.

Annotation examples. We now zoom into the individual rows from Table A.1 and provide rating examples, in order to illustrate the different degrees of relatedness that you may find in the judgment task. Note again that these are just examples and you should always provide your own subjective opinion.

The two instances of *Ofen* in Example A are judged identical in meaning (rating: 4), because both uses refer to an apparatus for heating and burning.

	A	B	C	D
1	**Satz 1**	Bewertung	Kommentar	**Satz 2**
2	Aus dem **Ofen** nehmen und sofort mit Kristallzucker bestreuen oder die noch heißen Shortbreads in Kristallzucker wenden.	4		Hält man seine Hand in den **Ofen** fühlt sich die Luft im Backrohr warm an.

Beispiel A.: rating 4 (Identical).

In contrast, the two uses of *reiben* in Example B are judged closely related but not identical (rating: 3), because the meaning of *reiben* in target sentence 1 is 'to comminute' while the meaning in target sentence 2 is 'to rub', thus it is a more gentle kind of pressing and moving.

In Example C, the two uses of the word *maskieren* are related, but more distantly (rating: 2): Both meanings are related by a figurative similarity, in the sense that they both denote a kind of covering or veiling. However, unlike *reiben* above, the two uses of *maskieren* in this

	A	B	C	D
1	**Satz 1**	Bewertung	Kommentar	**Satz 2**
3	Er **rieb** seine schmerzende Wange.	3		Den Käse mit einer Reibe fein **reiben** und beiseite stellen.

Beispiel B.: rating 3 (Closely Related).

example have different meanings. Target sentence 1 is about glazing a dish with a sauce, while target sentence 2 is about people masking themselves for carnival.

1	**Satz 1**	Bewertung	Kommentar	**Satz 2**
4	Nun soviel von dem Gemisch aus Mayonnaise und Clotted Cream über die Eier geben, dass diese vollständig bedeckt (oder **"maskiert")** sind.	2		Viele Künstler **maskieren** sich nicht nur zu Fasching, sondern das ganze Jahr über.

Beispiel C.: rating 2 (Distantly Related).

A rating of 1 is assigned for two uses of a word that are completely unrelated in their meanings, as it is the case for *parieren* in Example D. Note that this pair of uses is semantically more distant than the two uses of *maskieren* above. The meaning in target sentence 1 is 'to obey', while the meaning in target sentence 2 denotes removing connective tissue from meat.

	A	B	C	D
1	**Satz 1**	Bewertung	Kommentar	**Satz 2**
5	Wenn Rex dann nicht **parierte**, schlug man ihn .	1		Die Leber waschen, **parieren,** auf gewünschte Größe portionieren, zwei Stunden in Milch einlegen, damit die Leber ausbluten kann und gewisse Bitterstoffe entzogen werden.

Beispiel D.: rating 1 (Unrelated).

Finally, there is also the option for you to provide the judgment 'Cannot decide' (rating:0). Please use this rating only if absolutely necessary, when you are unable to make a decision as to the degree of relatedness in meaning between the two bold words. Please provide a comment for why you cannot decide about this pair of uses.

General and domain-specific language. The sentences provided for the annotation task were gathered both from an automatically crawled general-language German web corpus and from an automatically crawled text collection for a specific domain. Due to the fact that some content is user-generated and that the content was automatically crawled, some sentences might be less grammatical than extracts from standard text (e.g. books). Do not be led astray by that.

Also, please note that terms might be used differently as you might think. Specific domains, as for example cooking, DIY or hunting, have special terminology which includes everyday

terms with different meanings. Concentrate only on the target word in its given context, and try to understand its meaning. If you find that a sentence is too flawed to understand it, or the use of the target word is ambiguous, or the two instances of the target word do not match (i.e., they do not have the same lemma), please provide a comment to this effect.

Progressing through the task. While annotating the sentence pairs, you can always go back to previous judgments and change them, if you change your mind when new material is coming up. Also, you do not have to annotate the whole file in one session. If you wish to leave a comment at any point during the task, please type it into the comment field.

You may also want to turn off the spell checker to not be disturbed by additional highlighting.

Finishing the task. Please make sure that you do not change anything in the file apart from column width, font size, your judgments and comments. Return the annotated document to E-MAIL. If you have any further questions on the task, do not hesitate to ask.

A.5. Annotation guidelines for section 6.2 (English translation)

Domain: Cooking & Baking

→ A terminology in the domain of cooking comprises all expressions that are needed to prepare food (e.g. for cooking, baking, preparing drinks, ...)

Rating	Meaning	Examples
1	not a term	*Deutschland* "Germany", *Neuzeit* "modern age", *Thailand* "Thailand"
2	semantically related to the domain	*Vitaminbedarf* "requirement of vitamins", *Esstisch* "dining table", *Plastikdose* "tupperware"
3	understandable term	*Schweinebraten* "roast pork", *Kräuterbutter* "herb butter", *Schokoladenplätzchen* "chocolate cookie"
4	non-understandable term	*Blausud [blue boiling]* "special kind of boiling fish by adding acid", *Darioleform* "dariole mould", *Myrrhenkerbel* "cicely"

Important: When a term is ambiguous, only take the domain-specific meaning into consideration (e.g. consider *Strudel* as food, not in the meaning of "whirlpool")

Tip: non-understandable term: For which expression would you need an explanation or picture to understand the term.

A.6. Annotation guidelines for section 6.3 (English translation of guidelines by Julia Bettinger)

Dear participant,

this study investigates the degree of difficulty of terms (expressions of a domain) in relation to their understandability.

Every term will be presented to you in **bold** in context of three sentences, which you are asked to read before you fill in your rating. After that, please rate the term for its difficulty, or in other words respectively the question to what extent specialized knowledge is needed to understand the term's meaning.

Your rating can comprise the ratings 1 to 4, which are defined as follows:

1: The term does not require any specialized knowledge in order to be understood.

2: The term requires little specialized knowledge in order to be understood.

3: The term requires specialized knowledge.
Parts of its meaning can be inferred from its context.

4: The term requires specialized knowledge.
Its meaning cannot be inferred from its context.

You will receive three XLSX table documents, as illustrated in figure A.1.
The terms you are asked to rate are written in bold within their context sentences. Please fill in your rating in the table cell highlighted in yellow in column 2, according to the criteria given above. Different terms are separated by a blank line.

	A	B
1	*Satz*	*Bewertung*
2	Zunächst Wasser in einem **Kochtopf** zum Kochen bringen.	
3	Den **Kochtopf** von der Herdplatte nehmen.	
4	Für die Menge an Nudeln empfiehlt sich ein großer **Kochtopf.**	
5		
6	Teig erneut durchkneten und in den gewässerten und eingefetteten **Römertopf** geben.	
7	Im geschlossenen **Römertopf** im nicht vorgeheizten Backofen bei 180 Grad ca. 3 1/2 Stunden garen.	
8	Den **Römertopf** wässern.	
9		
10	Im Idealfall sind **Darioleformen** aus Edelstahl, da sich die Speisen daraus leichter lösen und stürzen lassen.	
11	Eine **Darioleform** eignet sich beispielweise zur Herstellung von Soufflé.	
12	Als Ersatz für eine **Darioleform** kann auch eine Kaffeetasse verwendet werden.	
13		
14	Nach Belieben kann man den **Blausud** abseihen oder das Gemüse herausnehmen.	
15	Gelatine nach Vorschrift einweichen, ausdrücken und in dem heißen **Blausud** einschließlich auflösen.	
16	Als Einlage für einen **Blausud** eignen sich u.a. Röstzwiebeln, Wacholderbeeren, Nelken und Lorbeerblätter.	

Figure A.1.: annotation table.

To clarify the task, we provide you with four examples from the domain of cooking in the following. Please note that these are only examples which do not necessarily have to match with the rating you would assign.

(A)

	A	B
1	*Satz*	*Bewertung*
2	Zunächst Wasser in einem **Kochtopf** zum Kochen bringen.	1
3	Den **Kochtopf** von der Herdplatte nehmen.	
4	Für die Menge an Nudeln empfiehlt sich ein großer **Kochtopf.**	

Rating = 1

(A) shows an example for an expression that is used in daily life and for which no specialized knowledge is needed.

(B)

	A	B
1	*Satz*	*Bewertung*
2	Teig erneut durchkneten und in den gewässerten und eingefetteten **Römertopf** geben.	2
3	Im geschlossenen **Römertopf** im nicht vorgeheizten Backofen bei 180 Grad ca. 3 1/2 Stunden garen.	
4	Den **Römertopf** wässern.	

Rating = 2

For the expression in (B) one can understand that it denotes some kind of pot (*Topf* means pot). However, not every lay person would understand that *Römertopf* denotes a special clay pot.

(C)

	A	B
1	*Satz*	*Bewertung*
2	Im Idealfall sind **Darioleformen** aus Edelstahl, da sich die Speisen daraus leichter lösen und stürzen lassen.	3
3	Eine **Darioleform** eignet sich beispielweise zur Herstellung von Soufflé.	
4	Als Ersatz für eine **Darioleform** kann auch eine Kaffeetasse verwendet werden.	

Rating = 3

The expression in (C) is not understandable for lay people. If the term *Darioleform* is unknown, its meaning can be understood from the context sentences in which it is used. However, the exact meaning remains unclear.

(D)

	A	B
1	*Satz*	*Bewertung*
2	Nach Belieben kann man den **Blausud** abseihen oder das Gemüse herausnehmen.	4
3	Gelatine nach Vorschrift einweichen, ausdrücken und in dem heißen **Blausud** einschließlich auflösen.	
4	Als Einlage für einen **Blausud** eignen sich u.a. Röstzwiebeln, Wacholderbeeren, Nelken und Lorbeerblätter.	

Rating = 4

The expression in (D) is not understandable for lay people. If the term *Blausud* is not known, the context does not help to understand that it denotes a stock with acid feeding (which is frequently used to prepare fish).

Implementation

Answer intuitively and try to rate to what extent specialized knowledge is needed for **general understanding**. Thus, do not only consider your own knowledge. That means that even if you know a term, this might be due to the fact that you have a certain amount of specialized knowledge.

Please note that there are no correct or false answers in this task. Please rate intuitively.

While annotating, you can go back to the previous rating at ay time and you can change it, in case you changed your opinion. You do not have to annotate the wohle file in one session.

Comment on the data

Due to the user-created contents and the way how we automatically extracted the texts sentences might be grammatically incorrect or duplicates might occur. Do not get distracted by that.

Submission

Please make sure that you do not change the files except for column width, character size and your ratings. Send the documents to [...]

If you have questions about the task do not hesitate to ask them.

Bibliography

Ahmad, K., Davies, A., Fulford, H., and Rogers, M. (1994). What is a term? The semi-automatic extraction of terms from text. *Translation Studies: An Interdiscipline. Selected papers from the Translation Studies Congress, Vienna, 1992*, 2:267—278.

Amjadian, E., Inkpen, D., Paribakht, T. S., and Faez, F. (2016). Local-global vectors to improve unigram terminology extraction. In *Proceedings of the 5th International Workshop on Computational Terminology*, pages 2–11, Osaka, Japan.

Amjadian, E., Inkpen, D., Paribakht, T. S., and Faez, F. (2018). Distributed specificity for automatic terminology extraction. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 24(1):23–40.

Arcan, M. (2017). *Machine Translation of Domain-Specific Expressions within Ontologies and Documents*. PhD thesis, Insight Centre for Data Analytics, National University of Ireland, Galway.

Arcan, M., Turchi, M., Tonelli, S., and Buitelaar, P. (2014). Enhancing statistical machine translation with bilingual terminology in a cat environment. In *Proceedings of the 11th Biennial Conference of the Association for Machine Translation in the Americas*, pages 54–64, Vancouver, BC, Canada.

Artetxe, M., Labaka, G., and Agirre, E. (2016). Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 451–462, Vancouver, Canada.

Artetxe, M., Labaka, G., and Agirre, E. (2018a). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5012–5019, New Orleans, Louisiana, USA.

Artetxe, M., Labaka, G., and Agirre, E. (2018b). A robust self-learning method for fully

unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 789–798, Melbourne, Australia.

Astrakhantsev, N. A., Fedorenko, D. G., and Turdakov, D. Y. (2015). Methods for automatic term recognition in domain-specific text collections: A survey. *Programming and Computer Software*, 41(6):336–349.

Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Baumgartner Jr., W. A., Bretonnel Cohen, K., Verspoor, K., Blake, J. A., and Hunter, L. E. (2012). Concept annotation in the CRAFT corpus. *BMC Bioinformatics*, 13(161):1–20.

Baker, M. (1988). Sub-technical vocabulary and the ESP teacher: An analysis of some rhetorical items in medical journal articles. *Reading in a Foreign Language*, 4(2):91–105.

Baldwin, T., Li, Y., Alexe, B., and Stanoi, I. R. (2013). Automatic term ambiguity detection. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 804–809, Sofia, Bulgaria.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The WaCky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! A Systematic Comparison of Context-counting and Context-predicting Semantic Vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 238–247, Baltimore, MD, USA.

Baroni, M., Matiasek, J., and Trost, H. (2002). Predicting the components of German nominal compounds. In *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 470–474, Lyon, France.

Barrón-Cedeno, A., Sierra, G., Drouin, P., and Ananiadou, S. (2009). An improved automatic term recognition method for spanish. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 125–136. Springer, Berlin Heidelberg.

Basile, P., Caputo, A., and Semeraro, G. (2015). Temporal random indexing: A system for analysing word meaning over time. *Italian Journal of Computational Linguistics*, 1:55–68.

Basili, R., Pazienza, M. T., Moschitti, A., and Zanzotto, F. M. (2001). A contrastive approach to term extraction. In *4th Terminology and Artificial Intelligence Conference*, Nancy, France.

Bauer, L. (2003). *Introducing linguistic morphology*. Edinburgh University Press.

Beck, I. L., McKeown, M. G., and Kucan, L. (2002). *Bringing words to life*. New York, NY: The Guilford Press.

Bernier-Colborne, G. (2012). Defining a gold standard for the evaluation of term extractors. In *In Proceedings of the Terminology and Knowledge Representation Workshop, LREC 2012*, page 15.

Bernier-Colborne, G. and Drouin, P. (2014). Creating a test corpus for term extractors through term annotation. *Terminology*, 20(1):50–73.

Bernth, A., McCord, M., and Warburton, K. (2003). Terminology extraction for global content management. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 9(1):51–69.

Bettinger, J. (2019). Predicting term difficulty of closed German noun compounds. Master's thesis, Institute for Natural Language Processing, University of Stuttgart, Stuttgart.

Bettinger, J., Hätty, A., Dorna, M., and Schulte im Walde, S. (2020). A Domain-Specific Dataset of Difficulty Ratings for German Noun Compounds in the Domains DIY, Cooking and Automotive. In *Proceedings of the 12th International Conference on Language Resources and Evaluation*, Marseille, France.

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, Beijing, China. Association of Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bonin, F., Dell'Orletta, F., Venturi, G., and Montemagni, S. (2010a). A contrastive approach to multi-word term extraction from domain corpora. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 19—21, Malta.

Bonin, F., Dell'Orletta, F., Venturi, G., and Montemagni, S. (2010b). Contrastive filtering of domain-specific multi-word terms from different types of corpora. In *Proceedings of the 2010 workshop on multiword expressions: From theory to applications*, pages 77–80.

Bouamor, D., Llanos, L. C., Ligozat, A.-L., Rosset, S., and Zweigenbaum, P. (2016). Transfer-based learning-to-rank assessment of medical term technicality. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, Portorož, Slovenia. European Language Resources Association.

Bourigault, D. (1992). Surface grammatical analysis for the extraction of terminological noun phrases. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 977–981, Nantes, France.

Breiman, L. (1996). Technical note: Some properties of splitting criteria. *Machine Learning*, 24(1):41–47.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth Publishing Company, Belmont, CA.

Bullinaria, J. and Levy, J. (2012). Extracting semantic representations from word co-occurrence statistics: Stop-lists, stemming, and svd. *Behavior research methods*, 44:890–907.

Cabré, M. T. (1996). Terminology today. *Terminology, LSP and Translation: studies in language engineering in honour of Juan C. Sager*, 18.

Cabré, M. T. (1999). *Terminology: Theory, methods and applications*, volume 1. John Benjamins Publishing, Amsterdam.

Cabré Castellví, M. T. (2003). Theories of terminology: Their description, prescription and explanation. *Terminology*, 9(2):163–199.

Cap, F. (2014). *Morphological processing of compounds for statistical machine translation*. dissertation, Institute for Natural Language Processing (IMS), University of Stuttgart.

Chawla, N., Bowyer, K., O. Hall, L., and Philip Kegelmeyer, W. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Chen, P. and Al-Mubaid, H. (2006). Context-based term disambiguation in biomedical literature. In *Proceedings of the International Conference of the Florida Artificial Intelligence Research Society*, pages 62–67.

Chopra, S., Hadsell, R., and Lecun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of Computer Vision and Pattern Recognition Conference*, pages 539–546. IEEE Press.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Clouet, E. L. and Daille, B. (2014). Splitting of compound terms in non-prototypical compounding languages. In *Proceedings of the First Workshop on Computational Approaches to Compound Analysis*, pages 11–19, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Cohen, J. D. (1995). Highlights: Language-and domain-independent automatic indexing terms for abstracting. *Journal of the American society for information science*, 46(3):162–174.

Cole, W. D. (1987). Terminology: Principles and methods. *Computers and Translations*, 2(2):77–85.

Cowan, J. R. (1974). Lexical and syntactic research for the design of EFL reading materials. *Tesol Quarterly*, pages 389–399.

Cram, D. and Daille, B. (2016). Terminology extraction with term variant detection. In

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. System Demonstrations, pages 13–18, Berlin, Germany.

da Silva Conrado, M., Salgueiro Pardo, T. A., and Oliveira Rezende, S. (2013). A machine learning approach to automatic term extraction using a rich feature set. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Student Research Workshop*, pages 16–23.

Daiber, J., Quiroz, L., Wechsler, R., and Frank, S. (2015). Splitting compounds by semantic analogy. In *Proceedings of the 1st Deep Machine Translation Workshop*, pages 20–28, Praha, Czechia. ÚFAL MFF UK.

Daille, B., Jacquey, E., Lejeune, G., Melo, L., and Toussaint, Y. (2016). Ambiguity diagnosis for terms in digital humanities. In *Language Resources and Evaluation Conference*.

de Castilho, E., R., Mújdricza-Maydt, ., Yimam, S., Hartmann, S., Gurevych, I., Frank, A., and Biemann, C. (2016). A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the workshop on Language Technology for Digital Humanities at COLING 2016*, Osaka, Japan. Association for Computational Linguistics.

De Jong, N. H., Schreuder, R., and Baayen, H. R. (2000). The morphological family size effect and morphology. *Language and cognitive processes*, 15(4-5):329–365.

Del Tredici, M. and Fernández, R. (2017). Semantic variation in online communities of practice. In *Proceedings of the 12th International Conference on Computational Semantics*, Montpellier, France.

Deutsches Textarchiv (2017). Grundlage für ein Referenzkorpus der neuhochdeutschen Sprache. Herausgegeben von der Berlin-Brandenburgischen Akademie der Wissenschaften.

Dobrov, B. and Loukachevitch, N. (2011). Multiple evidence for term extraction in broad domains. In *Proceedings of the international conference recent advances in natural language processing 2011*, pages 710–715.

Donoso, G. and Sanchez, D. (2017). Dialectometric analysis of language variation in twitter. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 16–25, Valencia, Spain.

Drouin, P. (2003). Term extraction using non-technical corpora as a point of leverage. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 9(1):99–115.

Drouin, P. (2004). Detection of domain specific terminology using corpora comparison. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 79–82, Lisbon, Portugal.

Dubossarsky, H., Hengchen, S., Tahmasebi, N., and Schlechtweg, D. (2019). Time-Out: Tem-

poral Referencing for Robust Modeling of Lexical Semantic Change. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy.

Dubossarsky, H., Weinshall, D., and Grossman, E. (2017). Outta control: Laws of semantic change and inherent biases in word representation models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1147–1156, Copenhagen, Denmark.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218.

Erbs, N., Santos, P. B., Zesch, T., and Gurevych, I. (2015). Counting what counts: Decompounding for keyphrase extraction. In *Proceedings of the ACL 2015 Workshop on Novel Computational Approaches to Keyphrase Extraction*, pages 10–17.

Escartín, C. P. (2014). Chasing the perfect splitter: A comparison of different compound splitting tools. In *In Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 3340–3347.

Estopà, R. (2001). Les unités de signification spécialisées: élargissant l'objet du travail en terminologie. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 7(2):217—237.

Evert, S. (2005). *The statistics of word cooccurrences: word pairs and collocations*. PhD thesis, Institute for Natural Language Processing, University of Stuttgart.

Faaß, G. and Eckart, K. (2013). SdeWaC – A corpus of parsable sentences from the web. In Gurevych, I., Biemann, C., and Zesch, T., editors, *Language Processing and Knowledge in the Web*, volume 8105 of *Lecture Notes in Computer Science*, pages 61–68. Springer, Berlin Heidelberg.

Faber Benítez, P. (2009). The cognitive shift in terminology and specialized translation. *MonTi: Monografías de Traducción e Interpretación*, 1(1):107–134.

Faber Benítez, P., Márquez Linares, C., and Vega Expósito, M. (2005). Framing terminology: A process-oriented approach. *Meta: Journal des traducteurs/Meta: Translators' Journal*, 50(4).

Fedorenko, D., Astrakhantsev, N., and Turdakov, D. (2014). Automatic recognition of domain-specific terms: an experimental evaluation. *Proceedings of the Institute for System Programming*, 26(4):55–72.

Ferrari, A., Donati, B., and Gnesi, S. (2017). Detecting domain-specific ambiguities: An NLP approach based on wikipedia crawling and word embeddings. In *Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference Workshops*, pages 393–399.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological*

Bulletin, 76(5):378–382.

Foo, J. and Merkel, M. (2010). Using machine learning to perform automatic term recognition. In *LREC 2010 Workshop on Methods for automatic acquisition of Language Resources and their evaluation methods*, pages 49–54, Valletta, Malta. European Language Resources Association.

Frantzi, K., Ananiadou, S., and Mima, H. (2000). Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, 3(2):115–130.

Frantzi, K. T., Ananiadou, S., and Tsujii, J. (1998). The c-value/nc-value method of automatic recognition for multi-word terms. In *International Conference on Theory and practice of Digital Libraries*, pages 585–604. Springer, Berlin Heidelberg.

Frermann, L. and Lapata, M. (2016). A Bayesian model of diachronic meaning change. *Transactions of the Association for Computational Linguistics*, 4:31–45.

Fuertes-Olivera, P. A. and Tarp, S. (2014). A critical view of terminography. In *Theory and practice of specialised online dictionaries: Lexicography versus terminography*, volume 146, chapter 7, pages 104–128. Walter de Gruyter GmbH & Co KG.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Goldberg, Y. (2017). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309.

Goldberg, Y. and Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Grabar, N., Hamon, T., and Amiot, D. (2014). Automatic diagnosis of understanding of medical words. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 11–20, Gothenburg, Sweden. Association for Computational Linguistics.

Gulordava, K. and Baroni, M. (2011). A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 67–71, Edinburgh, Scotland. Association of Computational Linguistics.

Ha, A. Y. H. and Hyland, K. (2017). What is technicality? A technicality analysis model for eap vocabulary. *Journal of English for Academic Purposes*, 28:35–49.

Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2016a). Cultural shift or linguistic drift? Comparing two computational measures of semantic change. In *Proceedings of the 2016*

Conference on Empirical Methods in Natural Language Processing, pages 2116–2121, Austin, Texas.

Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2016b). Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany.

Hätty, A., Dorna, M., and Schulte im Walde, S. (2017a). Evaluating the reliability and interaction of recursively used feature classes for terminology extraction. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 113–121, Valencia, Spain.

Hätty, A., Heid, U., Moskvina, A., Bettinger, J., Dorna, M., and Schulte im Walde, S. (2019a). Akkubohrhammer vs. akkubohrhammer: Experiments towards the evaluation of compound splitting tools for general language and specific domains. In *Proceedings of the 15th Conference on Natural Language Processing*, pages 59–67, Erlangen, Germany.

Hätty, A., Schlechtweg, D., and Schulte im Walde, S. (2019b). SURel: A gold standard for incorporating meaning shifts into term extraction. In *Proceedings of the 8th Joint Conference on Lexical and Computational Semantics*, Minneapolis, MN, USA.

Hätty, A., Schlechtweg, D., and Schulte im Walde, S. (2020). Predicting Degrees of Technicality in Automatic Terminology Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Seattle, WA, USA.

Hätty, A. and Schulte im Walde, S. (2018a). Fine-grained termhood prediction for German compound terms using neural networks. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions*, pages 62–73, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Hätty, A. and Schulte im Walde, S. (2018b). A laypeople study on terminology identification across domains and task definitions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 321–326, New Orleans, Louisiana.

Hätty, A., Tannert, S., and Heid, U. (2017b). Creating a gold standard corpus for terminological annotation from online forum data. In *Proceedings of the IWCS Workshop on Language, Ontology, Terminology and Knowledge Structures*, Montpellier, France.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

Hazem, A. and Morin, E. (2017). Bilingual word embeddings for bilingual terminology extraction from specialized comparable corpora. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 685–

693. Asian Federation of Natural Language Processing.

Hirschman, L. and Sager, N. (1982). Automatic information formatting of a medical sublanguage. *Sublanguage: studies of language in restricted semantic domains*, pages 27–80.

Hoffmann, L. (1985). *Kommunikationsmittel Fachsprache: eine Einführung*. Gunter Narr Verlag Tübingen.

Honnibal, M. and Johnson, M. (2015). An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

Hovy, D. and Purschke, C. (2018). Capturing regional variation with distributed place representations and geographic retrofitting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4383–4394, Brussels, Belgium.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456, Lille, France.

ISO 1087-1 (2000). *Terminology work – Vocabulary – Part 1: Theory and application*. International Organization for Standardization, Geneva, Switzerland.

Johnson, W. B. and Lindenstrauss, J. (1984). Extensions to Lipshitz mapping into Hilbert space. *Contemporary mathematics*, 26.

Justeson, J. S. and Katz, S. M. (1995). Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27.

Kageura, K. (2012). *The quantitative analysis of the dynamics and structure of terminologies*. John Benjamins Publishing, Amsterdam.

Kageura, K. and Umino, B. (1996). Methods of automatic term recognition: A review. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 3(2):259–289.

Karan, M., Šnajder, J., and Bašić, B. D. (2012). Evaluation of classification algorithms and features for collocation extraction in Croatian. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, Istanbul, Turkey. European Languages Resources Association.

Kim, J.-D., Ohta, T., Tateisi, Y., and Tsujii, J. (2003). Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Kim, Y., Chiu, Y., Hanaki, K., Hegde, D., and Petrov, S. (2014). Temporal analysis of language through neural language models. In *Proceedings of the ACL Workshop on Language Technologies and Computational Social Science*, pages 61–65.

Kit, C. and Liu, X. (2008). Measuring mono-word termhood by rank difference via corpus comparison. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 14(2):204–229.

Koch, P. and Oesterreicher, W. (1985). Sprache der Nähe – Sprache der Distanz. *Romanistisches Jahrbuch*, 36(85):15–43.

Kochetkova, N. A. (2015). A method for extracting technical terms using the modified weirdness measure. *Automatic Documentation and Mathematical Linguistics*, 49(3):89–95.

Koehn, P. and Knight, K. (2003). Empirical methods for compound splitting. In *10th Conference of the European Chapter of the Association for Computational Linguistics*.

Kucza, M., Niehues, J., Zenkel, T., Waibel, A., and Stüker, S. (2018). Term extraction via neural sequence labeling a comparative evaluation of strategies using recurrent neural networks. In *Interspeech*, pages 2072–2076.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1).

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 2177–2185, Montreal, Canada.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

L'Homme, M.-C. (2004). *La terminologie : principes et techniques*. Les Presses de l'Université de Montréal.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37:145–151.

Lison, P. and Kutuzov, A. (2017). Redefining context windows for word embedding models: An experimental study. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 284–288, Gothenburg, Sweden. Association for Computational Linguistics.

Liu, L., Kang, J., Yu, J., and Wang, Z. (2005). A comparative study on unsupervised feature selection methods for text clustering. In *2005 International Conference on Natural Language Processing and Knowledge Engineering*, pages 597–601. IEEE.

Loginova, E., Gojun, A., Blancafort, H., Guégan, M., Gornostay, T., and Heid, U. (2012). Reference lists for the evaluation of term extraction tools. In *Proceedings of the 10th International Congress on Terminology and Knowledge Engineering*, Madrid, Spain.

Lopes, L., Fernandes, P., and Vieira, R. (2016). Estimating term domain relevance through term frequency, disjoint corpora frequency-tf-dcf. *Knowledge-Based Systems*, 97:237–249.

Lyse, G. I. and Andersen, G. (2012). Collocations and statistical analysis of n-grams. *Exploring Newspaper Language: Using the web to create and investigate a large corpus of modern Norwegian*, 49:79.

Macken, L. and Tezcan, A. (2018). Dutch compound splitting for bilingual terminology extraction. *Multiword Units in Machine Translation and Translation Technology*, 341.

Manning, C. D., Manning, C. D., and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Maynard, D. and Ananiadou, S. (1998). Term sense disambiguation using a domain-specific thesaurus. In *Proceedings of 1st International Conference on Language Resources and Evaluation*, pages 681–687, Granada, Spain.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Menon, S. and Mukundan, J. (2010). Analysing collocational patterns of semi-technical words in science textbooks. *Pertanika Journal of Social Sciences and Humanities*, 18(2):241–258.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mikolov, T., Yih, W., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Moreno-Ortiz, A. and Fernández-Cruz, J. (2015). Identifying polarity in financial texts for sentiment analysis: A corpus-based approach. *Procedia-Social and Behavioral Sciences*, 198:330–338.

Mykowiecka, A., Marciniak, M., and Rychlik, P. (2018). Recognition of irrelevant phrases in automatically extracted lists of domain terms. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 24(1):66–90.

Nakagawa, H. and Mori, T. (2003). Automatic term recognition based on statistics of compound nouns and their components. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 9(2):201–219.

Nokel, M., Bolshakova, E., and Loukachevitch, N. (2012). Combining multiple features for

single-word term extraction. *Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue"*, 1(11):490–501.

Ohta, T., Tateisi, Y., and Kim, J.-D. (2002). The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the second international conference on Human Language Technology Research*, pages 82–86. Morgan Kaufmann Publishers Inc.

Paul, H. (2002). *Deutsches Wörterbuch: Bedeutungsgeschichte und Aufbau unseres Wortschatzes*. Niemeyer, Tübingen, 10. edition.

Pearson, J. (1998). *Terms in context*. Studies in Corpus Linguistics. John Benjamins Publishing, Amsterdam.

Pecina, P. and Schlesinger, P. (2006). Combining association measures for collocation extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 651–658, Sydney, Australia. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Pérez, M. J. M. (2016). Measuring the degree of specialisation of sub-technical legal terms through corpus comparison. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 22(1):80–102.

QasemiZadeh, B. and Schumann, A.-K. (2016). The acl rd-tec 2.0: A language resource for evaluating term extraction and entity recognition methods. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 1862–1868.

Ramisch, C., Villavicencio, A., and Boitet, C. (2010). mwetoolkit: a framework for multiword expression identification. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, volume 10, pages 662–669, Valletta, Malta. European Languages Resources Association.

Rayson, P. and Garside, R. (2000). Comparing corpora using frequency profiling. In *Proceedings of the Workshop on Comparing Corpora*, pages 1–6, Hong Kong.

Reddy, S., McCarthy, D., and Manandhar, S. (2011). An empirical study on compositionality in compound nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 210–218.

Riedl, M. and Biemann, C. (2016). Unsupervised compound splitting with distributional semantics rivals supervised methods. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 617–622.

Rigouts Terryn, A., Hoste, V., and Lefever, E. (2018). A gold standard for multilingual automatic term extraction from comparable corpora: Term structure and translation equivalents. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, Miyazaki, Japan.

Rigouts Terryn, A., Hoste, V., and Lefever, E. (2019). In no uncertain terms: a dataset for monolingual and multilingual automatic term extraction from comparable corpora. *Language Resources and Evaluation*, page 1–34.

Roelcke, T. (1999). *Fachsprachen*. Grundlagen der Germanistik. Erich Schmidt Verlag.

Rösiger, I., Bettinger, J., Schäfer, J., Dorna, M., and Heid, U. (2016). Acquisition of semantic relations between terms: How far can we get with standard NLP tools? In *Proceedings of the 5th International Workshop on Computational Terminology*, pages 41–51, Osaka, Japan.

Sager, J. C. (1990). *A Practical Course in Terminology processing*. John Benjamins Publishing, Amsterdam.

Sahlgren, M. (2004). An introduction to random indexing. *Language*, pages 1–9.

Salton, G. and McGill, M. J. (1983). *Introduction to modern information retrieval*. McGraw-Hill, New York.

Santus, E., Lenci, A., Lu, Q., and Schulte im Walde, S. (2014). Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 38–42.

Schäfer, J., Rösiger, I., Heid, U., and Dorna, M. (2015). Evaluating noise reduction strategies for terminology extraction. In *Proceedings of the conference Terminology and Artificial Intelligence*, pages 123–131, Granada, Spain.

Schlechtweg, D., Eckmann, S., Santus, E., Schulte im Walde, S., and Hole, D. (2017). German in flux: Detecting metaphoric change via word entropy. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pages 354–367, Vancouver, Canada.

Schlechtweg, D., Hätty, A., del Tredici, M., and Schulte im Walde, S. (2019). A wind of change: Detecting and evaluating lexical semantic change across times and domains. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 732–746.

Schlechtweg, D. and Schulte im Walde, S. (2018). Distribution-based prediction of the degree of grammaticalization for German prepositions. In Cuskley, C., Flaherty, M., Little, H., McCrohon, L., Ravignani, A., and Verhoef, T., editors, *The Evolution of Language: Proceedings of the 12th International Conference*.

Schlechtweg, D., Schulte im Walde, S., and Eckmann, S. (2018). Diachronic Usage Relatedness (DURel): A framework for the annotation of lexical semantic change. In *Pro-*

ceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 169–174, New Orleans, Louisiana, USA.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Schmid, H. (1995). Improvements in Part-of-Speech Tagging with an Application to German. In *Proceedings of the Special Interest Group on Linguistic data and corpus-based approaches to NLP, ACL 95*, Dublin, Ireland.

Schmid, H., Fitschen, A., and Heid, U. (2004). SMOR: A German computational morphology covering derivation, composition and inflection. In *Proceedings of the fourth international conference on Language Resources and Evaluation*, pages 1–263. Lisbon.

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10.

Schulte im Walde, S., Hätty, A., and Bott, S. (2016). The role of modifier and head properties in predicting the compositionality of English and German noun-noun compounds: A vector-space perspective. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 148–158.

Schulte im Walde, S., Müller, S., and Roller, S. (2013). Exploring vector space models to predict the compositionality of German noun-noun compounds. In *Second Joint Conference on Lexical and Computational Semantics, Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 255–265.

Shannon, C. E. (1948). *A Mathematical Theory of Communication*. CSLI Publications, Stanford, CA.

Siegel, S. and Castellan, N. J. (1988). *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, Boston, MA.

Taghipour, K. and Ng, H. T. (2015). Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 314–323, Denver, Colorado.

Temmerman, R. (2000). *Towards new ways of terminology description: The sociocognitive-approach*, volume 3. John Benjamins Publishing, Amsterdam.

Tilley, J. W. (2008). *A comparison of statistical filtering methods for automatic term extraction for domain analysis*. PhD thesis, Virginia Tech.

Trimble, L. (1985). *English for Science and Technology. A Discourse Approach*. Cambridge

University Press, Cambridge.

Trimble, R. M. T. and Trimble, L. (1978). The development of EFL materials for occupational English: The technical manual. *English for Specific Purposes. Science and Technology*, pages 74–132.

Tuggener, D. (2016). *Incremental coreference resolution for German*. PhD thesis, Universität Zürich.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Tutin, A. (2007). Traitement sémantique par analyse distributionnelle des noms transdisciplinaires des écrits scientifiques. *Actes de Traitement Automatique des Langues Naturelles*, pages 283–292.

Vydiswaran, V.-V., Mei, Q., Hanauer, D. A., and Zheng, K. (2014). Mining consumer health vocabulary from community-generated text. In *AMIA Annual Symposium Proceedings*, volume 2014, pages 1150–1159. American Medical Informatics Association.

Wang, R., Liu, W., and McDonald, C. (2016). Featureless domain-specific term extraction with minimal labelled data. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 103–112.

Wang, Y., Gutiérrez, I. L. M., Winbladh, K., and Fang, H. (2013). Automatic detection of ambiguous terminology for software requirements. In *Proceedings of the International Conference on Application of Natural Language to Information Systems*, pages 25–37.

Weller-Di Marco, M. (2017). Simple compound splitting for German. In *Proceedings of the 13th Workshop on Multiword Expressions*, pages 161–166.

Wermter, J. and Hahn, U. (2006). You can't beat frequency (unless you use linguistic knowledge): a qualitative evaluation of association measures for collocation and term extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 785–792. Association for Computational Linguistics.

Wüster, E. (1974). General terminology theory – fine line between linguistics, logic, ontology, information science and business sciences. *Linguistics*, 119(1):61–106.

Wüster, E. (1979). *Einführung in die allgemeine Terminologielehre und terminologische Lexikographie*. Springer, Berlin Heidelberg.

Yimam, S. M., Gurevych, I., Eckart de Castilho, R., and Biemann, C. (2013). Webanno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6, Sofia, Bulgaria. Association for Computational Linguistics.

Yuan, Y., Gao, J., and Zhang, Y. (2017). Supervised learning for robust term extraction. In *2017 International Conference on Asian Language Processing*, pages 302–305. IEEE.

Zadeh, B. Q. and Handschuh, S. (2014). The ACL RD-TEC: A dataset for benchmarking terminology extraction and classification in computational linguistics. In *Proceedings of the 4th International Workshop on Computational Terminology*, pages 52–63, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Zeng, Q., Kim, E., Crowell, J., and Tse, T. (2005). A text corpora-based estimation of the familiarity of health terminology. *International Symposium on Biological and Medical Data Analysis*, pages 184–192.

Zeng-Treitler, Q., Goryachev, S., Tse, T., Keselman, A., and Boxwala, A. (2008). Estimating consumer familiarity with health terminology: a context-based approach. *Journal of the American Medical Informatics Association*, 15(3):349–356.

Zhang, C., Niu, Z., Jiang, P., and Fu, H. (2012). Domain-specific term extraction from free texts. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 1290–1293. IEEE.

Zhang, Q., Wang, Y., Gong, Y., and Huang, X. (2016a). Keyphrase extraction using deep recurrent neural networks on twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 836–845, Austin, TX, USA.

Zhang, X., Song, Y., and Fang, A. C. (2010). Term recognition using conditional random fields. In *Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering*, pages 1–6.

Zhang, Z., Gao, J., and Ciravegna, F. (2016b). Jate 2.0: Java automatic term extraction with apache solr. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 2262–2269.

Zhang, Z., Iria, J., Brewster, C., and Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms. In *Proceedings of the sixth international conference on Language Resources and Evaluation*.

Ziering, P., Müller, S., and van der Plas, L. (2016). Top a splitter: Using distributional semantics for improving compound splitting. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 50–55.