DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Computational Science and Engineering

Improving German Legal Information Retrieval with Contextual Word Embeddings for Word Sense Disambiguation

Andreas Pad Probst

DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Computational Science and Engineering

Improving German Legal Information Retrieval with Contextual Word Embeddings for Word Sense Disambiguation

Verbesserung der juristischen Informationsbeschaffung mit Hilfe von kontextuellen Word Embeddings für Word Sense Disambiguation

Author: Andreas Pad Probst
Supervisor: Prof. Dr. Florian Matthes
Advisor: Ingo Glaser, M. Sc.
Date: July 15, 2021

Ich versichere, dass ich diese Masterarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 14. Juli 2021 Andreas Pad Probst

Acknowledgments

I would like to express my gratitude to my supervisor, Professor Florian Matthes, for giving me the opportunity to write this thesis. I want to thank my advisor Ingo Glaser for his guidance and valuable advice. I would also like to thank my colleagues Jörg Landthaler and David Pfahler for their encouragement and continuous support throughout my thesis journey. I thank the TUM Entrepreneurial Masterclass for providing me with the opportunity to combine my thesis with my entrepreneurial ambitions. My deepest gratitude goes to my parents, who made it possible for me to pursue my goals up to this point. Lastly, I want to express my thanks to Prof. Alexander Mehler and Daniel Baumartz for taking the time to help me reach my research goals.

Abstract

Legal research takes up a large proportion of lawyers' working hours. A lot of time is lost because search engines of German legal publishers retrieve too many irrelevant results, making it more cumbersome for lawyers to find relevant legal information. A solution to this problem is implementing a word sense filter that captures the intention behind the keywords given to the search engine. However, manual sense annotations are time-consuming and supervised German word sense classifiers suffer from data sparsity.

In this thesis, we present a method to automatically generate a vast amount of sense annotated data using the disambiguation pages of the German Wikipedia. We implement and train two linear neural network-based architectures for sense disambiguation that can be used to automatically annotate German legal data. We introduce the fast and efficient "hashing model" and the "fastBERT" model, a sense classifier based on contextual word embeddings generated with BERT. Both sense classifiers are evaluated quantitatively, and parameter studies are conducted.

Our best performing fine-tuned fastBERT model achieves an accuracy of over 74% when tested on a manually annotated legal corpus. Our findings indicate that one major limiting factor of our model's precision is the subjective perception of word meanings. Results are also qualitatively evaluated with legal experts to assess our sense filter's usefulness and collect suggestions for further improvements. Significant findings from the qualitative evaluations were that lawyers want word senses to be clearly distinguishable and that the granularity of distinguished senses should be adjustable.

Contents

Acknowledgements **vii**

Abstract **ix**

1. Introduction **1**
1.1. Motivation . . . 1
1.2. Problem Statement . . . 3
1.3. Research Approach and Research Questions . . . 4
1.4. Structure of the Thesis . . . 7

2. Fundamentals and Related Work **9**
2.1. Word Embedding Technologies . . . 9
2.1.1. Word2Vec . . . 9
2.1.2. FastText . . . 10
2.1.3. Contextual Word Embeddings . . . 11
2.2. Word Sense Disambiguation . . . 12
2.2.1. Baselines and Evaluation Methods . . . 14
2.2.2. Knowledge-Based WSD Approaches . . . 16
2.2.3. Unsupervised WSD Approaches . . . 17
2.2.4. Supervised WSD Approaches . . . 18
2.2.5. Research on WSD in Information Retrieval . . . 21
2.2.6. WSD Resources for the German Language . . . 22

3. Preparation and Resources **25**
3.1. Expert Interviews . . . 25
3.2. Legal Dataset . . . 27
3.3. Legal Test Set for WSD Classifier . . . 31
3.4. Overview of Available Resources . . . 36

4. Implementation **41**
4.1. Key Idea Behind the Implementation . . . 41
4.2. Sense Classifier Overview . . . 42
4.3. Data Creation Pipeline . . . 44
4.3.1. Extractor . . . 45
4.3.2. Exporter . . . 48
4.4. Neural Model Architectures . . . 50

5. Quantitative Evaluation **55**
5.1. Hashing Model Evaluation . . . 55

5.2. FastBERT Evaluation 59

6. Qualitative Evaluation **67**
6.1. General Search Behaviour 67
6.2. Sense Filter Evaluation 68

7. Conclusion and Outlook **73**
7.1. Summary 73
7.2. Critical Reflection and Limitations 75
7.3. Possible Improvements and Future Work Outlook 76

Appendix **79**

A. Technical Documentation **79**
A.1. Data Creation 79
A.2. Model Training 81

B. Expert Interview Guideline **85**

C. Expert Interview Results **89**

D. Qualitative Evaluation Guideline **103**

E. Qualitative Evaluation Results **107**

Bibliography **135**

1. Introduction

1.1. Motivation

The digital transformation is reaching all industry sectors. A publication by [Geissbauer et al., 2017] shows that in 2017, more than 91% of industrial companies are in some form involved in digital initiatives in Europe. 90% of the surveyed companies believe that digitalization will provide them with more opportunities rather than risks. One of the industry sectors that profit from digitalization is the legal sector. In 2020, the worldwide legal service market size was worth $713.6 billion and will reach $908.7 billion by 2025 at an annual growth rate of 5%, according to The Business Research Company[1].

Even though the legal industry is adopting digital technologies slower than other major industries, digital legal technologies are becoming more popular. [Caserta and Madsen, 2019] even go so far as to predict that big data and artificial intelligence will fundamentally change the way law is practiced in the near future. For example, by integrating new digital technologies, law firms can reach more people or develop new niche areas of practice, such as eLawyering and online dispute resolution. One of the best examples of what can be achieved with artificial intelligence was ROSS Intelligence[2]. Based on IBM's cognitive computer Watson[3], they have created an artificially intelligent attorney, albeit with limited functionality.

Nowadays, law firms already deploy technologies to assist with contract management, due diligence, and legal research. Out of those, improvements in digital legal research are one the most significant ones for lawyers, according to [Opijnen and Santos, 2017]. This is hardly surprising. As information keeps steadily increasing, finding relevant information becomes more challenging than finding the needle in the haystack. The relevance of legal information retrieval is further underpinned by the article from [Bench-Capon et al., 2012]. They describe the history of AI and law with the help of 50 influential papers from the international conference on AI and Law (ICAIL). Seven of the papers are related to legal information retrieval, which is a strong indicator of its importance.

Emerging cross-country collaborations further increase the need for fast and accurate information retrieval. One of the best examples of this is the creation and expansion of the European Union. [Lyytikäinen et al., 2000] describe the challenges for European legal information retrieval and how internationalization has lead to a greater need to know more about legal norms and regulations of other countries. People and organizations became

[1] https://www.thebusinessresearchcompany.com/report/legal-services-global-market-report. Visited on Apr. 30, 2021.
[2] https://blog.rossintelligence.com/. Visited on Apr. 30, 2021.
[3] https://www.ibm.com/watson. Visited on Apr. 30, 2021.

regulated by the European Union law in addition to their national one.

An insight paper from [Lastres, 2012] states that law firm associates spend a third of their working hours conducting legal research. Given the relevance for fast and accurate access to legal information, it is hardly surprising that many companies see a lucrative business opportunity in providing smart solutions for digital legal research. One good example is the U.S. company Casetext[4]. They speed up legal research using advanced artificial intelligence algorithms. They are the fastest-growing source of online legal information in the U.S., and over 4,000 small firms adapted Casetext within the last year.

One possible way to improve legal information retrieval is to make machines intelligent enough to understand context. In his 1950 paper, [Turing, 1950] describes the idea that if a machine can hold a conversation in a way that is indistinguishable from a human, it is artificially intelligent. To achieve this, the machine needs to be able to both understand and generate natural language, and an essential part of understanding language is the task of word sense disambiguation (WSD). WSD is the ability to associate ambiguous words with the correct meaning based on the context they appear in. For example, the word "bank" can refer to a financial institution, a reservoir of some kind (e.g., a seed bank), or the land alongside a body of water. Disambiguating word senses is one of the more challenging tasks in natural language processing. As [Ranjan Pal and Saha, 2015] state,

> "A normal human being has an inborn capability to differentiate the multiple senses of an ambiguous word in a particular context, but the machines run only according to the instructions. So, different rules are fed to the system to execute a particular task."

This statement suggests that in order to have machines do well in sense disambiguation tasks, they would need to have vast in-depth extra-linguistic knowledge. In 1960, [Bar-Hillel, 1960] stated that the only reason humans can distinguish word senses is because they are making intelligent use of their general world knowledge. However, publications such as [Papandrea et al., 2017] and [Kumar et al., 2019] show that WSD systems are able to achieve high accuracy on standardized WSD benchmarks consistently nowadays. One of the most recent WSD models is described by [Scarlini et al., 2020] and can consistently score around 80% accuracy on five different English WSD test sets. These promising results make it feasible to use WSD to improve existing technologies. [Agirre and Edmonds, 2007] list the most important application areas, which are machine translation, information extraction, and information retrieval (IR).

The viability of using WSD to improve the performance of IR systems has first been analysed in 1973 by [Weiss, 1973]. In his paper, he uses a machine learning model for the sense disambiguation of five ambiguous hand-picked words from the ADI document collection[5]. He reports an improvement of 1% on the fraction of the relevant documents that are retrieved. Since then, more research on the usage of WSD in IR has been done. A brief overview of early research in this field is given by [Sanderson, 1996]. Later, in 2012, it was shown by [Zhong and Ng, 2012] that when using a sense tagged corpus, their model per-

[4] https://casetext.com/. Visited on Apr. 30, 2021.

[5] "The ADI collection is a set of short papers on automation and scientific communication published by the American Documentation Institute, 1963". [Weiss, 1973]

formance surpassed state-of-the-art IR systems at the time. With all this in mind, it is clear that there much potential in applying WSD in IR. This thesis focuses on the application of WSD in legal IR for the German language.

1.2. Problem Statement

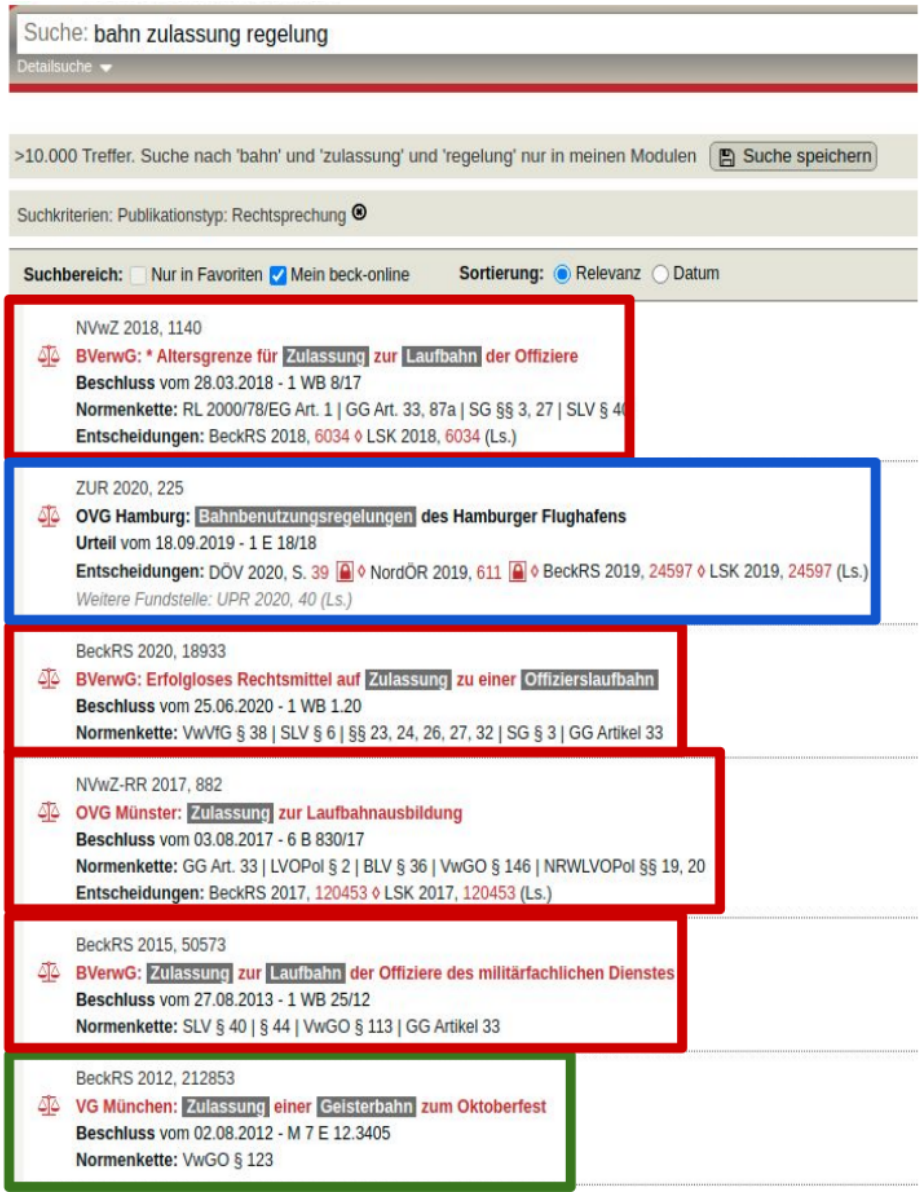


Figure 1.1.: Motivational use case. The search results relate to differing senses of the word "Bahn". The results relate to "Bahn" in the sense of career (red), runway (blue) and rail vehicle (green).

Search engines of German legal databases[6] cannot distinguish between different word senses. Figure 1.1 depicts a concrete motivational use case. Assuming that a legal expert is looking on beck-online[7] for information regarding the commissioning regulation of rail vehicles, one possible search query he could enter into the search bar would be "bahn zulassung regelung" (en. "train admission regulation"). The colors in the figure indicate to which word sense the German word "Bahn" in the search results relates to.

One might argue that legal experts could be more specific in their search intention and use domain-specific keywords. However, according to [Geist, 2016], a user study done by C. H. Beck[8] found that legal experts usually use natural language terms rather than domain specific legal terms for their search. Furthermore, they mention a study done by RIDA[9] which showed that 90% of search queries only contain a single word. This makes automatic sense disambiguation based on word co-occurrences in the search query impossible for most search queries. A solution to this problem is the manual filtering for word senses. We believe that by specifying the intended senses of words in the search query, the user will retrieve documents that are more relevant for him. This thesis aims to investigate the feasibility of utilizing WSD in legal information retrieval (LIR) to improve precision and recall.

1.3. Research Approach and Research Questions

The research approach in this thesis follows the design science research (DSR) framework for information systems, which is described in [Hevner et al., 2004]. DSR is highly application-oriented. The starting point is an underlying problem, based on which a solution, the so-called artifact, is created. In the context of information systems, the artifact is usually a software application. Subsequently, the artifact's performance is evaluated. The framework is depicted in Figure 1.2. It consists of three closely related iterative activity cycles: the relevance cycle, the rigor cycle and the central design cycle. The relevance cycle defines in which context (environment) the problem occurs and sets the application requirements and the acceptance criteria for the artifact. It also clarifies the involved people, organizations, and technology. The rigor cycle ensures that the existing knowledge base (methodologies, theories, and related artifacts) is considered when developing the artifact. Lastly, the design cycle acts as the interface between the other two cycles and is needed to create and evaluate the artifact. The generated knowledge is then subsequently fed back into the knowledge base.

It should be mentioned that the framework just prescribes the overall methodology. The individual methods to achieve the overall goal can be chosen freely. The following methods are chosen based on findings from the respective iteration cycle.

[6] A list of the most widely used German legal databases is given at https://www.legal-tech.de/juristische-datenbanken/. Visited on May 01, 2021.
[7] https://beck-online.beck.de/Home. Visited on Apr. 19, 2021.
[8] https://www.chbeck.de/. Visited on Apr. 19, 2021.
[9] https://www.ridaonline.at/RidaOnline/index1.a5w. Visited on Apr. 19, 2021.

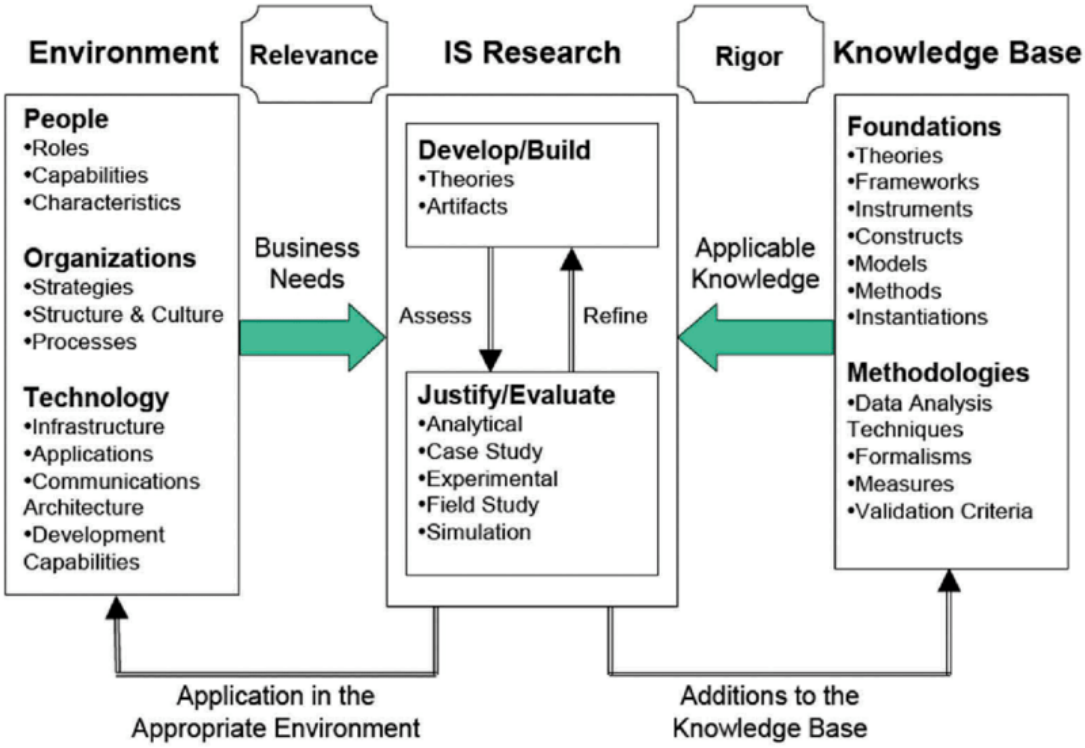


Figure 1.2.: Design science research framework for information systems. Depicted are the three iterative cycles and their relationship to one another. Image taken from [Hevner et al., 2004].

Relevance Cycle

As described in the previous section, the underlying problem this thesis tackles is that legal search engines do not provide filtering options based on word senses. The idea is to give users of legal databases the option to filter out results based on word meaning. The significant advantage of this is that users can remove irrelevant results by using an additional sense filter instead of adapting and changing their search behaviour. To better understand the demands and search behaviour of legal practitioners and understand the working process of lawyers, interviews are conducted.

Rigor Cycle

As a next step, a WSD model of suitable accuracy needs to be trained. Literature reviews on WSD as a stand-alone application and the application of WSD in IR are conducted. Based on first rigor cycle iterations, it became clear that literature on German WSD is sparse. One of the most comprehensive works on WSD research for the German language is provided by [Henrich, 2015]. However, there is only little training data available for the German language, making the training of supervised WSD models difficult. One possibility would be

to use GermaNet[10], a sizeable lexical-semantic net that was first presented by [Hamp and Feldweg, 1997]. For example, one could take the WordNet[11] based algorithm presented in [Ben Aouicha et al., 2018] and adapt it to GermaNet, while at the same time trying to reduce German-language specific errors that might occur in the process, such as the ones described by [Saito et al., 2002]. A comparative evaluation of GermaNet based WSD algorithms is presented by [Henrich and Hinrichs, 2012].

However, in this thesis, an algorithm that has first been proposed by [Uslu et al., 2018] and that was later adapted by [Schulz, 2021] will be used. Training data for the model is generated using the disambiguation pages of the German Wikipedia. The sense classification algorithm itself consists of a single-layered neural network that can handle big data. Advantages over GermaNet are a more extensive sense inventory, as well as more training data. Being independent of a language-specific lexical-semantic network that requires many resources to create, an additional advantage is that the model is more easily adapted for other languages. All it needs are the freely available Wikipedia dumps[12] in the corresponding language. Another reason why this algorithm was chosen for this thesis is because it is supervised. Supervised algorithms for sense disambiguation generally perform better than unsupervised or knowledge-based algorithms. The algorithm uses an artificial neural network as a classification layer, which is also seen as an advantage over many other supervised methods because neural networks can generalize well and are more stable. The fastSense algorithm comprises two parts, an algorithm for generating the training data from the Wikipedia dumps and an algorithm for the classification task. The two parts can be decoupled, which lets us evaluate the performance on WSD tasks with different classifiers.

Design Cycle

The adapted WSD model produced in the rigor cycle and the application requirements from the relevance cycle are combined into a suitable application. As currently no sense annotated legal corpus exists for the German language, a small legal corpus will be manually tagged with word senses. This allows us to evaluate the trained model on legal data. The model can subsequently be used to tag a legal corpus with additional word senses information. A qualitative evaluation with multiple legal experts will be conducted to assess the quality and relevance of the word senses that the model can distinguish

Research Questions

Based on the research approach described in this section, the following research questions are formulated:

RQ1 What algorithms already exist to automatically classify word senses (knowledge-based, unsupervised, and supervised)?

[10]https://uni-tuebingen.de/en/142806. Visited on Apr. 19, 2021.
[11]https://wordnet.princeton.edu/. Visited on Apr. 19, 2021.
[12]https://dumps.wikimedia.org/backup-index.html. Visited on Apr. 19, 2021.

WSD algorithms are classified as being either supervised, unsupervised or knowledge-based. In order to develop a new algorithm for WSD tasks on legal text, it is essential to have an overview and understanding of the underlying working mechanisms of existing WSD classifiers. We will research how word senses can be modelled mathematically.

RQ2 What possibilities are there to compensate for the lack of German sense annotated (legal) data?

We will investigate what data exist to date to train German WSD classifiers. Possible ways to compensate for the lack of sense annotated data for the German language will be researched.

RQ3 How do WSD algorithms perform on German (legal) text?

Based on the results of the previous two research questions, we investigate what kind of algorithms are best suited for WSD tasks on German texts and German legal corpora. Suitable baselines for benchmarking purposes will be researched.

RQ4 How do legal experts judge the usefulness of our word sense filter?

As our aim is to improve legal information retrieval, we will conduct interviews with legal experts. Two interview rounds are conducted, one in the beginning of the thesis and one at the end. The purpose of the first interview round is to assess the usefulness and requirements of a sense-based filter for legal experts. In the second interview rounds, the experts will judge the quality of the distinguishable word senses and give suggestions for further improvements.

1.4. Structure of the Thesis

The remaining part of this thesis is organized as follows.

Chapter 2: Fundamentals and Related Work

This chapter explains the word embedding technologies used for this thesis in detail. The task of WSD is introduced, along with standard baselines and evaluation measures. Examples of knowledge-based, unsupervised and supervised WSD algorithms and their advantages and disadvantages are discussed. The chapter concludes with an overview of resources for German WSD.

Chapter 3: Preparation and Resources

More concrete use cases are derived from interview results with German legal practitioners. The legal corpus that is given to us by the publisher Dr. Otto Schmidt is analysed. The

process of creating a manual sense tagged legal data set is explained. The chapter finishes with an overview of the used resources for this thesis.

Chapter 4: Implementation

The key ideas behind the algorithms used in this thesis are explained. A method to automatically generate a sense annotated corpus from Wikipedia is described. Two model architectures for sense classification are introduced.

Chapter 5: Quantitative Evaluation

In this chapter, the models are quantitatively evaluated, and parameter studies are conducted. The results of the evaluation are discussed.

Chapter 6: Qualitative Evaluation

Legal practitioners were interviewed to evaluate the quality of the distinguishable senses by our sense classifier. The results of the interview are analysed and discussed.

Chapter 7: Conclusion and Outlook

A summary of the work in this thesis is given in this chapter. The research questions from Section 1.3 are answered. The results of this thesis and limitations are critically reflected, and a future work outlook is given.

2. Fundamentals and Related Work

2.1. Word Embedding Technologies

Neural networks require numeric input. To process and analyse text input, each word in a given vocabulary needs to be mapped to a unique numerical representation. One possibility to achieve this is with word embedding algorithms. Word embeddings are learned representations for textual data where each word is represented by a real-valued dense vector of predefined length. There exist two types of embedding models, count-based models and context-predicting models. In this thesis, we only consider context-predicting models when we speak of word embeddings. It was shown by [Baroni et al., 2014] that they outperform their count-based counterparts. For the sake of completeness, it should be mentioned that some research papers trace the performance differences back to specific system design choices rather than the embedding methods themselves. One example is [Levy et al., 2015].

Embedding algorithms operate on the assumption of the **distributional hypothesis**. The idea goes back to [Harris, 1954] and [Firth, 1957], who stated that linguistic terms with a similar distribution tend have similar meanings. This intuition was further confirmed by [Miller and Charles, 1991]. They found a strong correlation between the meaning of words and the context in which the words appear. This chapter provides an overview of embedding technologies relevant for this thesis.

2.1.1. Word2Vec

The most straightforward approach to map words to numeric values is to use one-hot encoded vector representations. Each component of the vector represents a word from the underlying vocabulary and is either zero or one. This approach leads to sparse vectors, which each have the length of the vocabulary size. However, there is no natural way to compare vectors for their semantic similarity. In their paper, [Mikolov et al., 2013] introduce two model architectures, the Continuous-Bag-of-Words model (CBOW) and Skip-gram model, to create dense word vectors that can capture both syntactic and semantic regularities. The models are trained with unlabelled raw text corpora and only contain a single linear hidden layer. Despite the models' simplicity, they produce high-quality word vectors at a low computational cost.

Furthermore, [Mikolov et al., 2013] train both the CBOW model and the Skip-gram model on a prediction task. While the CBOW model predicts a target word based on the neighbouring words within a specific word window, the Skip-gram model predicts words within a word window of the input word. Figure 2.1 depicts the working mechanisms of the mod-

els. At the input layer of both models, each word is one-hot encoded and projected onto a lower-dimensional vector by a dense projection matrix, whose weights get adjusted during the training process. This projection matrix is later used as a lookup table that maps each word to its unique dense vector representation.

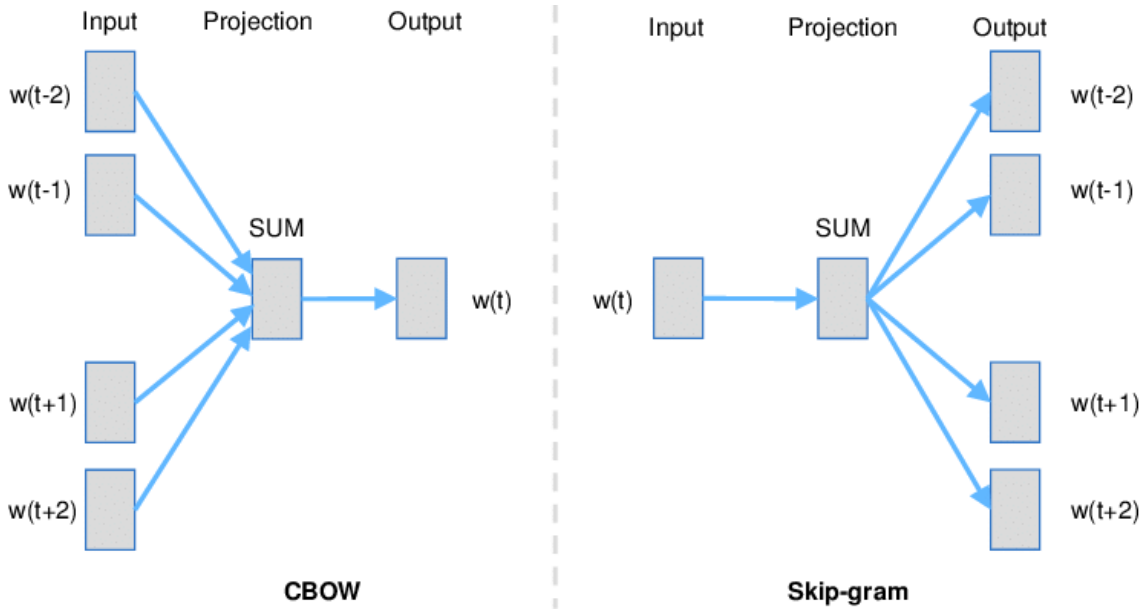


Figure 2.1.: Illustrations of the CBOW and Skip-gram model. Image taken from [Landthaler et al., 2017].

.

2.1.2. FastText

FastText[1] is an extension to the Word2Vec model. It is an algorithm based on the publications of [Joulin et al., 2016] and [Bojanowski et al., 2017]. FastText was designed to be a simple linear classifier. Figure 2.2 depicts the architecture of the fastText model. The main difference to the Word2Vec model is that each word is represented by the average of the vector representations of all possible character n-grams of the token. Composing words out of smaller subwords allows the model to take morphological aspects of words into account. For example, the representations of the words "walk", "walking", and "walked" share some of the trained parameters of the model as they all consist of the subword "walk". By combining the respective n-grams, fastText can find representations for out-of-vocabulary words that did not appear during the training process. The fastText model usually outperforms models that do not take subword information into account, especially in morphologically rich languages[2].

Internally fastText adds the angle brackets "<" and ">" to each word to mark its beginning and its end. The word is then split into character n-grams of length n. For example, when

[1]https://research.fb.com/downloads/fasttext/. Visited on Apr. 19, 2021.
[2]https://fasttext.cc/blog/2016/08/18/blog-post.html. Visited on May 02, 2021.

working with 3-grams, the word "eating" is first transformed into "<eating>" and then split into the following 3-grams: "<ea", "eat", "ati", "tin", "ing", "ng>"[3]. The vector representation for "eating" is calculated by taking the average of the vector representations of each token, as well as the vector representation of the word itself. FastText can also compute the vector representation based on a range of character n-grams with different n to account for subwords of different lengths. For example, setting the input parameters[4] $minn = 3$ and $maxn = 5$ would compute a vector representation based on the average of the representations of all the possible 3-grams, 4-grams, 5-grams and the word itself. [Bojanowski et al., 2017] state that to reduce the memory needed to store representations of all the individual n-grams, they map each n-gram to an integer and hash the character sequence using the FNV-1a variation of the Fowler-Noll-Vo[5] hashing function.

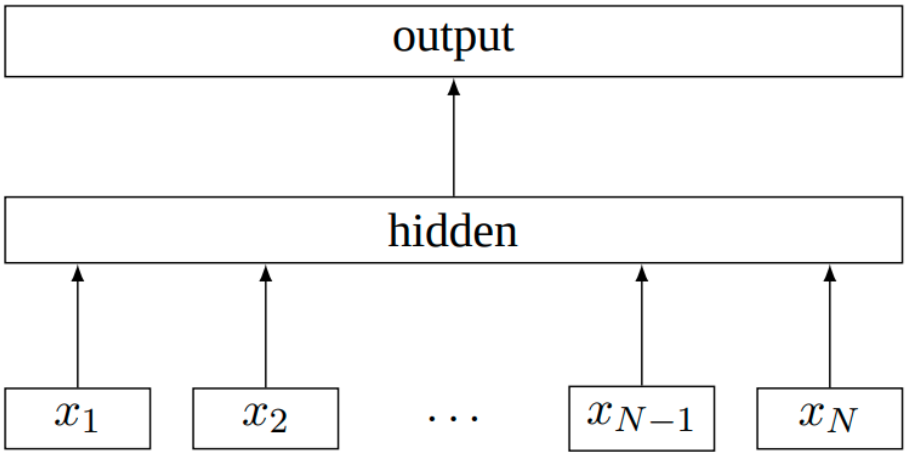


Figure 2.2.: Architecture of the linear fastText Classifier. $x_1, ..., x_N$ denote the n-gram features that get averaged and fed into the linear hidden layer. Image taken from [Joulin et al., 2016].

2.1.3. Contextual Word Embeddings

Like traditional word embedding techniques, such as Word2Vec and fastText, algorithms for contextual word embeddings learn a continuous vector representation for each word in the underlying vocabulary. The difference is that traditional word embedding algorithms learn one global representation for each word. As the name suggests, contextual embedding models can compute vector representations based on each token's context. The most famous models based on contextualized embeddings are the series of Generative Pre-trained Transformers (GPT-3, which was first presented by [Brown et al., 2020], and its

[3]Example taken from a blog post at https://amitness.com/2020/06/fasttext-embeddings/. Visited on May 02, 2021.
[4]Input parameters for the fastText model are listed at https://fasttext.cc/docs/en/options.html. Visited on May 02, 2021.
[5]http://www.isthe.com/chongo/tech/comp/fnv/. Visited on May 02, 2021.

precursors GPT-2 and GPT), as well as BERT and its many derivatives. BERT, which was published by [Devlin et al., 2019], and the deep generative models are based on the transformer architecture presented by [Vaswani et al., 2017]. Models using this architecture can move beyond word-level semantics and create language models that encode additional information in their vector representations based on the linguistic context in which words appear in. Figure 2.3 visualizes an example for contextual word embeddings. In the example, the word *mouse* gets assigned a context-dependent vector representation. Like the fastText model discussed in the previous subsection, transformer-based models can also compute representations of out-of-vocabulary words by decomposing the words into subwords or characters for which they can generate an embedding. A recent in-depth review on different contextual embedding algorithms was published by [Liu et al., 2020].

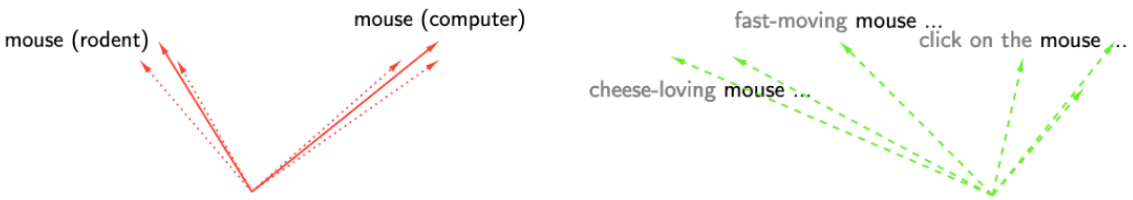


Figure 2.3.: Example of a contextual embedding representation. Depending on the context, the embedding vector (right) of the word *mouse* aligns more with the embedding vector (red) for *mouse* in the sense of either rodent or computer. Image taken from The Stanford AI Lab Blog[6].

Models like BERT and GPT were created with the idea in mind that language processing tasks are a two-step process, namely an unsupervised language modelling task, followed by a supervised fine-tuning task. The idea is that one can train large-scale language models on data-rich tasks so that the models develop an intrinsic general-purpose knowledge that can subsequently be transferred to downstream tasks, as explained by [Raffel et al., 2020]. For example, by putting an additional linear classification layer on top of a pre-trained language model, one could achieve state-of-the-art results on a wide range of tasks, like named-entity recognition or natural language inference. An in-depth study on the linguistic knowledge transferability of contextual word vectors was done by [Liu et al., 2019].

2.2. Word Sense Disambiguation

Word Sense Disambiguation (WSD) is a classification task where the correct meaning ("sense") of an ambiguous word is selected from a list of possible senses (the sense inventory) for that word. As [Lesk, 1986] puts it,

> "The meaning of an English word can vary widely depending on which sense is intended. Does a fireman feed fires or put them out? It depends on whether or not he is on a steam locomotive."

In order to find the correct sense, information from surrounding words, which make up

the context of the target word to be disambiguated, are taken into account. Linguistic clues from these words can be extracted (part-of-speech (POS), morphological features, NER, ...) and enriched with additional information from knowledge sources. The clues are subsequently used to predict the correct sense of a word. Contextual models like *SensEmBERT*, which was proposed by [Scarlini et al., 2020], are already able to achieve over 80% accuracy on English WSD tasks consistently. An upper bound on WSD performance was estimated by [Gale et al., 1992] to be 96.8%. They determined the upper bound by estimating the ability of human judges to agree with each other on word senses, or in other words, the ability to obtain reliable judgements from humans when assigning the correct meaning to a word.

Figure 2.4 depicts a WSD process workflow. WSD methods can be categorized as being either knowledge-based, supervised or unsupervised. This section provides an overview of the different WSD methods and standard baselines and evaluation metrics.

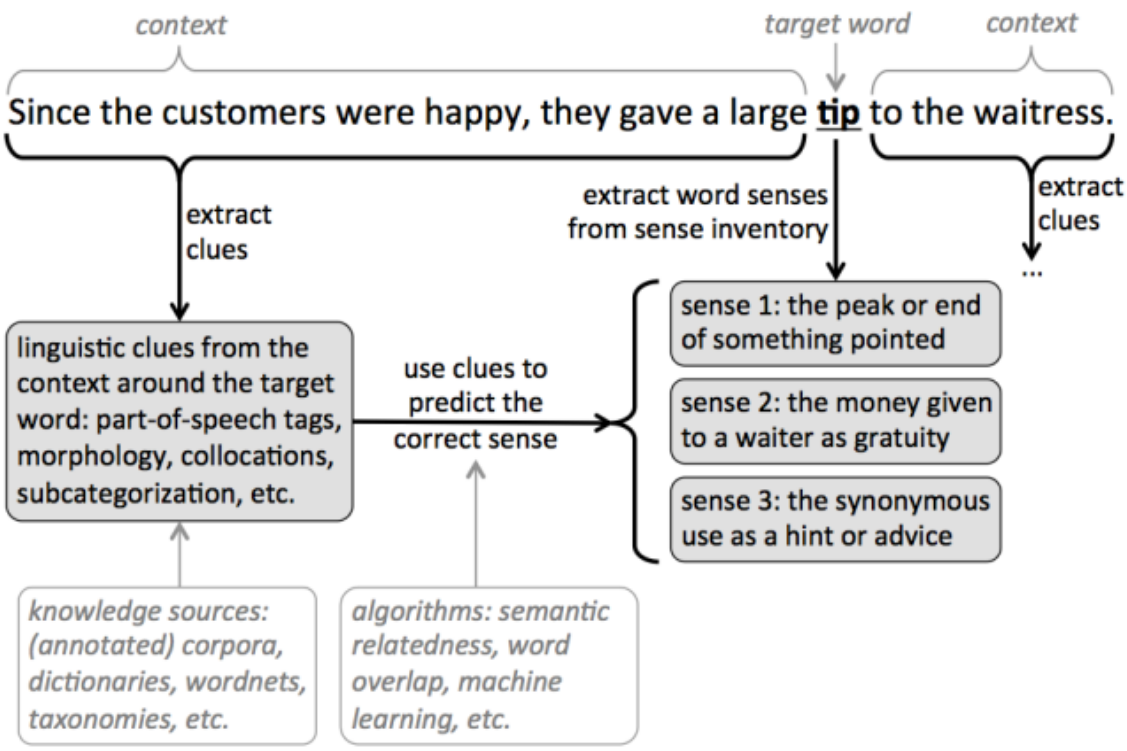


Figure 2.4.: Visualization of a WSD process workflow. The surrounding words of the target word to be disambiguated make up the context of the target word. Linguistic clues from those words are used to predict the correct sense of the target word. The list of possible senses for that word is taken from a sense inventory. In this specific example, the context indicates that *"sense 2"* is the correct sense for the target word *tip*. Image taken from [Henrich, 2015].

2.2.1. Baselines and Evaluation Methods

For comparison purposes, simple heuristics are used as a baseline to define a lower performance bound for algorithms. One possibility is the "Random Sense" baseline mentioned by [Navigli, 2009]. [Ranjan Pal and Saha, 2015] list three additional baselines, namely the "Most Frequent Sense", "One Sense per Discourse" and "One Sense per Collocation" baselines.

- **Most Frequent Sense**
  This is a strong baseline and often hard to beat by WSD systems. It was introduced by [Gale et al., 1992]. For each ambiguous word, it looks up the sense that appears most often in the corpus for that particular word. It then assigns that sense to each instance of the word. [Preiss et al., 2009] states that the reason for the high accuracy of the most frequent sense baseline is the skewed Zipfian distribution of word senses. Because a sense's frequency is approximately inversely proportional to its rank in the sense frequency table of the corpus, the top-ranked sense will appear much more likely than any other sense.
- **One Sense per Discourse**
  This baseline is based on the work of [Yarowsky, 1993]. It follows the heuristic that if the same word appears multiple times in the same text, it will preserve its sense.
- **One Sense per Collocation**
  This baseline is similar to the "One Sense per Discourse" baseline, except that it assumes that the nearer two polysemous words appear, the higher the probability that their senses are the same.
- **Random Sense**
  This baseline chooses a random sense from the sense inventory each time an ambiguous word appears. If word senses would be uniformly distributed, the success rate of a correct guess is inversely proportional to the number of senses for a particular word. However, due to the skewed word sense distribution, this baseline does not perform well usually.

To evaluate individual stand-alone WSD applications, sense annotated corpora are needed. The senses are taken from a predefined sense inventory. One of the most widely used sense inventories for the English language is Princeton University's WordNet[7], an extensive lexical database first presented by [Miller, 1995]. One example of such a sense annotated corpus is the semantic concordance (SemCor[8]) presented by [Miller et al., 1993]. A semantic concordance is a corpus where each substantive word in the text is linked to its appropriate sense according to a given lexicon. In the case of SemCor, the Brown Corpus[9] is used in combination with WordNet as a lexicon. Another popular dataset used for evaluation purposes is the Semeval[10] (formerly Senseval) dataset. The WSD tasks in Semeval

[7]https://wordnet.princeton.edu/. Visited on May 07, 2021.
[8]https://www.sketchengine.eu/semcor-annotated-corpus/. Visited on May 08, 2021.
[9]https://www.sketchengine.eu/brown-corpus/. Visited on May 08, 2021.
[10]https://semeval.github.io/. Visited on May 08, 2021.

are very diverse and comprise, amongst others, lexical sampling and machine translation tasks. There are many other sense annotated datasets available. A list of resources is given by [Navigli, 2009]. The creation of manually sense-tagged corpora is very resource expensive, and a disadvantage is that the annotated data is only helpful in combination with one particular sense inventory. Also, different WSD tasks require different types of sense annotated corpora. [Navigli, 2009] distinguishes two variants of generic WSD tasks that can be used for evaluation purposes:

- Lexical sampling (or targeted WSD), where the WSD system disambiguates a selected sample of target words. Each instance of the target word in the corpus must be labelled with the correct sense from a given sense inventory.
- All-words WSD, where the application classifies all ambiguous words in the corpus. Systems that perform this task require vast amounts of training data. The model has to be trained with samples where each sense of each ambiguous word appears, which causes supervised systems to suffer from data sparseness.

[Navigli, 2009] describes the following four evaluation measures that are suitable when assessing stand-alone WSD systems.

1. **Coverage**
   The coverage C is the percentage of ambiguous tokens in the test set for which the WSD system assigned a sense. It measures how likely it is that the system classifies an instance where classification is due.

$$C = \frac{\textit{\# provided answers}}{\textit{\# total answers to provide}}. \tag{2.1}$$

2. **Precision**
   The precision P is the proportion of correctly predicted word senses. The precision indicates how good the returned answers are.

$$P = \frac{\textit{\# correctly provided answers}}{\textit{\# total answers provided}}. \tag{2.2}$$

3. **Recall**
   The recall R is the fraction of correctly predicted senses over the total number of answers that can be given. It measures how well the system can detect and correctly classify all instances in the dataset that need to be classified. From the definitions, it becomes clear that $R \leq P$. $R = P$ is only possible for 100% coverage.

$$R = \frac{\textit{\# correctly provided answers}}{\textit{\# total answers to provide}}. \tag{2.3}$$

4. **F_1-Score**
   The F_1 score is the weighted average of precision and recall. As it takes both preci-

sion and recall into account, it is a useful measure when comparing systems that do not have 100% coverage.

$$F_1 = \frac{2PR}{P+R}. \tag{2.4}$$

2.2.2. Knowledge-Based WSD Approaches

The idea behind knowledge-based approaches is to avoid using large amounts of training material and instead use lexical knowledge bases. The algorithms can then use a rule-based system to classify word senses. [Escudero, 2006] lists typical resources used for such algorithms. These are machine-readable dictionaries, thesauri, computational lexicons, and lexical knowledge bases. A popular and cheap solution that uses machine-readable dictionaries (MRDs) is the Lesk algorithm introduced by [Lesk, 1986]. The underlying assumption is that words within a certain word window share common topics. The correct word sense of a target word can then be computed by counting the overlaps between dictionary definitions of the different possible senses of the words contained in the target word's neighbourhood.

For example, let us consider the words "pine cone" together. If we assume that according to a given dictionary, the word *pine* has the following two definitions:

1. kinds of evergreen tree with needle-shaped leaves
2. waste away through sorrow or illness

And if we assume that the word *cone* has the definitions

1. solid body which narrows to a point
2. something of this shape whether solid or hollow
3. fruit of evergreen tree

Then the correct sense for our target word *pine* is the first one, as the first sense definition of *pine* and the third sense definition of *cone* have the most word overlap (two words are overlapping, as stop words are not counted). As a direct consequence, the correct sense for the word *cone* would be the third one.

The performance of the Lesk algorithm strongly depends on the wording and the richness of the glosses in the used dictionary. To circumvent this, adaptations of the Lesk algorithms have been implemented, such as those presented by [Banerjee and Pedersen, 2002]. Rather than using glosses from traditional dictionaries, they use the semantic relations provided by WordNet. Their algorithm compares glosses of words related to the target word (e.g. synonyms) to the glosses of the other words in the text segment. This way, they can improve the overall WSD accuracy. A more recent variation of the Lesk algorithm that was able to achieve state-of-the-art results at that time was presented by [Basile et al., 2014].

Instead of using WordNet, they employ BabelNet[11] and use word similarity algorithms that measure context overlap instead of just counting the individual word overlap.

2.2.3. Unsupervised WSD Approaches

The most significant advantage of unsupervised methods is that they are not affected by the data bottleneck of sense annotated data since they work with unannotated corpora. They are also independent of predefined sense inventories. Such algorithms are also often called "fully unsupervised", as they do not use any other kind of knowledge besides unlabelled corpora. As [Navigli, 2009] states, unsupervised algorithms are based on the idea that the same sense of a word has similar neighbouring words. They induce word senses by clustering word occurrences that appear in similar contexts. This is why they are more suited to identify sense clusters instead of classifying word senses of individual words. Unsupervised WSD algorithms can be classified as clustering algorithms or graph-based algorithms.

Most unsupervised methods described in the literature are not fully unsupervised, as they use external knowledge sources like dictionaries. For example, [Martín and Berlanga-Llavori, 2012] introduces a clustering-based approach together with feature vectors built from WordNet, and [Mihalcea, 2005] uses a graph-based approach that uses WordNet as a sense inventory. It should also be mentioned here that some papers describe their algorithms as full unsupervised, like the method proposed by [Chen et al., 2009]. However, their algorithm requires a dictionary and thus is not fully unsupervised in the context of this thesis.

The idea behind clustering-based algorithms and graph-based algorithms shall be briefly explained in the following.

1. **Clustering-Based Algorithm**
   An example of clustering-based algorithms is the creation of context clusters. By assigning each word a vector representation one can compute the similarity between two words (e.g. with the cosine similarity). The similarity measures whether two words belong to the same context cluster or not. Each context cluster can be assigned a context vector by computing the average of the word vectors in the cluster. Subsequently, similarities between a new word and the context clusters can be computed. The new word is subsequently assigned to its appropriate context cluster, and the context vector is updated. Each context vector represents a distinct sense.

   This method is sometimes also referred to as context discrimination. As stated by [Schütze, 1998], while normal WSD performs sense labelling, context clustering divides occurrences of a word into classes by determining if two occurrences belong to the same sense class or not. This makes quantitative performance evaluation more difficult, as clusters might not relate to the traditional senses that are found in a dictionary.

[11] https://babelnet.org/. Visited on May 09, 2021.

2. **Graph-Based Algorithm**
   An example of graph-based approaches are algorithms that use co-occurrence graphs. The idea is to create graphs $G = (V, E)$, where each vertex V represents a word, and each edge E links the word to other words that appear in the same context. Here, context refers to a paragraph or more comprehensive text segment. An example of such an algorithm was presented by [Dorow and Widdows, 2003]. After creating a graph around a target word, graph clustering algorithms are used to partition the nodes into categories. Figure 2.5 illustrates an example of such a partitioned graph. In the example, clusters around the word *mouse* are formed.

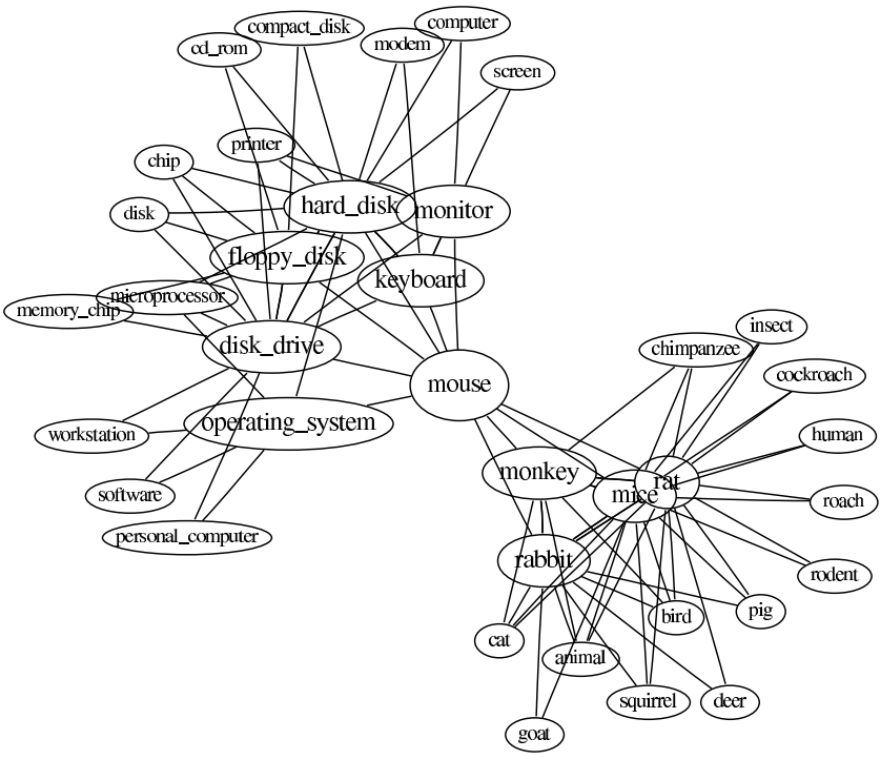


Figure 2.5.: Partitioned graph of the word *mouse*. One can see that the left cluster contains electric devices, while the right cluster comprises different animals. These two otherwise unrelated areas are linked together through the word *mouse*. *Mouse* can be mentioned either in the sense of an electronic device or a rodent. Image taken from [Dorow and Widdows, 2003].

2.2.4. Supervised WSD Approaches

Out of the three different approaches, supervised WSD methods typically achieve the best results. They need manually sense-tagged datasets for training. Their main limitation is the availability of large training corpora, as their creation costs a lot of time and resources. Another disadvantage compared to knowledge-based and unsupervised methods is that

they are limited to the words that appear in the training set. [Aliwy and Taher, 2019] and [Borah et al., 2014] provide a list of supervised methods suited for WSD. They describe supervised methods as a two-step process. First, vector representations of ambiguous word instances in the text are computed. Then supervised learning algorithms are applied to the embedded words. Additional information like POS or NER can also be taken into account during classification.

An interesting discovery was made by [Yarowsky and Florian, 2002], who concluded that there are no "one-size-fits-all" solutions. Even weakly performing algorithms can outperform top performers on the disambiguation of specific keywords. Furthermore, they observed that the quality of the used features has a much more significant impact on WSD performance than the choice of the classification algorithm itself. [Lee and Ng, 2002] found that the impact that different features and knowledge sources have on WSD performance depends on the used algorithm. For example, when taking POS information of neighbouring words into account, naïve Bayes classifiers got the highest performance increase, In contrast, Support Vector Machines benefited the most from taking word collocations (words that tend to appear together, e.g. *United States*) into account. In general, using a combination of different features resulted in better model performance.

In the following, some common types of supervised methods employed for WSD are listed.

1. **Decision List**
   Decision lists are based on predefined rules (*if-else* conditions) and are amongst the simplest algorithms. They were introduced by [Rivest, 1987]. They represent an ordered list of boolean functions that learn from examples. The algorithm learns by reassigning the order in which the boolean functions process the input. Instances of ambiguous words are checked against each boolean function in a specific order and are classified based on the first boolean function where all the conditions are fulfilled. According to [Aliwy and Taher, 2019], decision lists are prone to overfitting.

2. **Decision Tree**
   Decision trees are similar to decision lists, with the difference that decision trees constitute of an unordered list of more basic rules inside a tree structure. Each leaf of the decision tree corresponds to a single rule. A set of rules is formed by all the rules connected along the path of edges from the root to a particular leaf. The rules accumulate and make up more complex rules the further one traverses down the tree. However, decision trees are highly unstable as small changes in the input data can significantly change the output. An example of applying them for WSD is given by [Pedersen, 2001]. The nodes in his implementation check if certain bigrams appear within a window of approximately 100 words around the target word that should be disambiguated.

3. **naïve Bayes**
   Naïve Bayes is a probabilistic classifier based on Bayes' theorem. The main advantage of naïve Bayes is that the algorithm is straightforward to implement. It assumes that all features are conditionally independent. For each feature of an instance, it computes the probability of that feature belonging to a particular class. The indi-

vidual probabilities are then multiplied (because independence is assumed), which results in the (naïve) probability of that instance belonging to a specific class (e.g. a particular word sense).

4. **Exemplar/Memory Based Learning Methods**
   Exemplar based learning is based on learning from examples by progressively extending the model. According to [Chatterjee, 2012] the most common method of this type is the *k-Nearest-Neighbour* (kNN) algorithm. The kNN algorithm assumes that similar examples are closely together within an abstract feature space and according to a metric of choice. A new instance is assigned to the class to which most of the first k-nearest examples belong. The most significant disadvantage is that kNN based methods require a large amount of memory, as every training instance needs to be stored to compare them to the new input.

5. **Support Vector Machines**
   Support Vector Machines (SVMs) are linear classifiers. They are based on the idea of finding the best hyperplane that separates data points into classes. The points closest to the dividing hyperplane are referred to as the support vectors. The distance between the support vectors and the hyperplane is called the margin. The goal of the SVM classifier is to choose the hyperplane with the biggest possible margin. A disadvantage of SVMs is that they are less effective when classes of datasets overlap with each other in the feature space (e.g. due to noise) and cannot be separated cleanly.

6. **AdaBoost**
   AdaBoost ("Adaptive Boosting") is an ensemble learning method. It was first proposed by [Freund and Schapire, 1997]. It combines weak classifiers (classifiers who perform poorly at prediction tasks, like decision trees or logistic regression) into a single robust classifier with good performance. AdaBoost uses an iterative approach to learn from the mistakes of the weak classifiers. Misclassified instances by one classifier are corrected by other classifiers that are added progressively during each training iteration. This iterative process can be continued until the model does not improve any further.

7. **Neural Networks**
   Neural networks consist of layers of neurons that are connected to each neuron of adjacent layers. Each neuron processes its input based on weight variables adjusted during the training process and an activation function. The input instance passes from the input layer through the hidden layers until it reaches the output layer, and a class is assigned to the instance. Figure 2.6 visualizes a neural network. This is the type of supervised method that we use for this thesis. One example of a neural network-based classifier is given by [Uslu et al., 2018]. There, vector representations of sentences or paragraphs are fed into a single layer neural network to predict the most likely sense of a chosen target word.

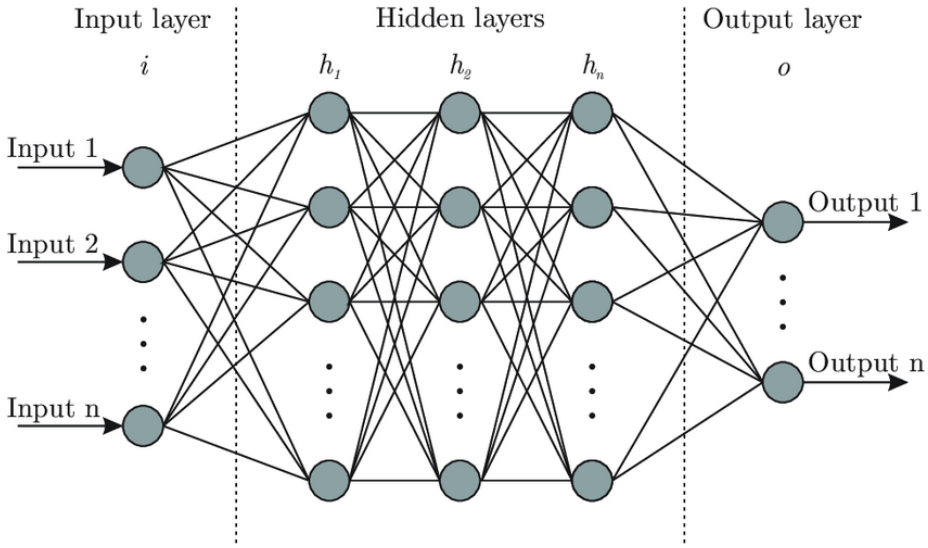


Figure 2.6.: Neural network with three hidden layers. The input passes from the input layer i through the hidden layers h_1, h_2, h_3 until it reaches the output layer o. Image taken from [Bre et al., 2017].

2.2.5. Research on WSD in Information Retrieval

Information retrieval (IR) systems assist in searching document collections and finding relevant documents that satisfy the user's requirements. The requirements are given to the system by the user in the form of a search query that contains natural language. The effects of polysemy on IR performance have been extensively studied. According to [Stokoe et al., 2003], word sense ambiguity has a detrimental effect on the precision of IR systems. [Uzuner et al., 1999] states that low disambiguation power is most likely the main cause for poor IR performance, as missing or incorrect disambiguations hinder techniques employed in IR systems, such as query expansion.

Some scientific literature states that WSD is only beneficial to IR if it achieves a high enough precision, as wrongly disambiguated words have a profound negative effect on IR performance. [Sanderson, 1996] concluded that WSD systems are somewhat resilient to ambiguity in search terms, but very sensitive to incorrectly disambiguated terms. He claims that WSD systems of at least 90% accuracy are needed before they positively affect IR systems. Explicit WSD with less than 90% accuracy yields worse results than if no WSD is employed at all. Even classifiers that had an accuracy of 75% were reported to harm IR performance. It would require a significant amount of effort to manually disambiguate occurrences of ambiguous words in the underlying test corpus for his evaluation. That was the reason why [Sanderson, 1996] used a technique that injects automatically generated ambiguous pseudo-words into the underlying corpus that can then be disambiguated for training and evaluation purposes. It is not clear how meaningful the results of this method are when compared to actual IR tasks. [Gonzalo et al., 1998] tested the impact that WSD has on IR with documents that contained manually disambiguated words and found that WSD accuracy over 70% still improved performance. [Stokoe et al., 2003] reported that an

accuracy of only 62.1% on WSD tasks showed an improvement in IR performance compared to systems that do not use WSD. A more recent study by [Zhong and Ng, 2012] discovered that when incorporating word senses into language modelling approaches used in IR, they outperformed state-of-the-art baseline IR systems. With all this in mind, it is reasonable to believe that it is possible to utilize WSD for our use case, namely to improve legal IR.

2.2.6. WSD Resources for the German Language

Most of the research done on WSD focuses on the English language. Research on German WSD is made difficult by the lack of appropriate resources. Sense inventories and high-quality sense-annotated corpora are needed for training and evaluation purposes, but their construction requires considerable time and manual work. This has lead to research on the automatic creation of large-scale lexical-semantic resources for the German language. One of them is the dissertation by [Matuschek, 2015], in which algorithmic approaches are presented to identify equivalent senses of the same word among different resources. The algorithms can be used to create lexical-semantic resources for training WSD models.

Regarding WSD itself, the dissertation of [Henrich, 2015] is probably the most comprehensive work on German WSD to date. She provides an overview of sense-annotated data and existing gold standard corpora for the German language and related literature in her work. In the following, a list of resources for German WSD that have been considered for this thesis is given.

GermaNet

GermaNet is a lexical-semantic network that is maintained by the University of Tübingen. GermaNet was first presented by [Hamp and Feldweg, 1997]. It was designed to be compatible with Princeton University's WordNet (but with some design differences[12]), so that previous research on automatic WSD also works for the German language with GermaNet as sense inventory. GermaNet also contains sense information for each word. Senses are represented as synsets. Synsets are collections of synonyms (in their lemmatized form, as GermaNet only contains lemmatized words), so each word sense is represented by a set of synonyms that describe the same concept. Furthermore, hierarchical hypernymy and hyponymy relations between different synsets exist. GermaNet (Release 16.0[13]) contains more than 195,000 lexical units and over 150,000 synsets. A disadvantage is that not every polyseme has a sense description, as there are only around 29,500 sense descriptions provided in total. Also, since the sense descriptions are taken from the German Wiktionary, a description usually only comprises one sentence, which is too little to train supervised WSD classifiers. Additionally, with GermaNet senses annotated corpora are needed to

[12] More information on the design and structure of GermaNet can be found at https://uni-tuebingen.de/en/faculties/faculty-of-humanities/departments/modern-languages/department-of-linguistics/chairs/general-and-computational-linguistics/ressources/lexica/germanet/description. Visited on May 19, 2021.

[13] Information on the current coverage of GermaNet is found at https://uni-tuebingen.de/en/142806. Visited on May 19, 2021.

train supervised German WSD models with enough training samples for each sense to achieve high accuracy.

Research around GermaNet as sense inventory has been conducted extensively[14]. One example is [Broscheit et al., 2010], who developed an algorithm to obtain predominant sense information with GermaNet as sense inventory. Missing predominant sense information in German was the reason why several unsupervised WSD methods or strong heuristics, like the Lesk algorithm, could not be used.

BabelNet

Babelnet is a multilingual lexical-semantic network. It was presented by [Navigli and Ponzetto, 2010]. BabelNet was automatically constructed by enriching sense inventories like WordNet and GermaNet with text from resources like Wikipedia or Wiktionary[15]. Like in GermaNet, senses in BabelNet are represented by synsets containing synonym that express a certain concept or meaning. To date, the German portion of BabelNet (Release 5.0[16]) contains over 6,700,000 synsets that comprise over 11,750,000 senses. As sense descriptions in BabelNet are enriched with encyclopedic content that covers a great range of topics, BabelNet could have been for this thesis. However, the resources were not available to us.

WebCAGe

WebCAGe is a web-harvested corpus that has been automatically annotated with senses from GermaNet. A first mapping between GermaNet senses and sense descriptions from the German Wiktionary has been implemented by [Henrich et al., 2011]. To ensure a high-quality mapping, manual post-correction steps were performed. The alignment yields an accuracy of 93.8%. Based on this work, [Henrich et al., 2012] created the WebCAGe sense-annotated corpus. Wiktionary sense entries of a given target word often contain descriptive examples that are linked to external web-based resources like Wikipedia articles and the German Gutenberg text archive. Those texts can be harvested and the target words that appear in the texts can be linked to their appropriate GermaNet sense. For this task, the "One Sense per Collocation" heuristic described in Subsection 2.2.1 was assumed. So target words that appear multiple times throughout the same text were assumed to share the same sense. WebCAGe comprises 46,457 words and 70,339 senses. However, only parts of it are available for download and usage[17].

[14] A list of publications around GermaNet is provided at https://uni-tuebingen.de/fakultaeten/philosophische-fakultaet/fachbereiche/neuphilologie/seminar-fuer-sprachwissenschaft/arbeitsbereiche/allg-sprachwissenschaft-computerlinguistik/ressourcen/lexica/germanet-1/publikationen/. Visited on May 19, 2021.

[15] A list of resources used in the construction of BabelNet can be found at https://babelnet.org/about. Visited on May 20, 2021.

[16] Coverage of BabelNet 5.0 can be found at https://babelnet.org/statistics. Visited on May 20, 2021.

[17] Parts of it can be downloaded from https://uni-tuebingen.de/fakultaeten/philosophische-fakultaet/fachbereiche/neuphilologie/seminar-fuer-sprachwissenschaft/arbeitsbereiche/allg-sprachwissenschaft-computerlinguistik/ressourcen/corpora/webcage/. Visited on May 20, 2021.

TüBa-D/Z treebank

The TüBa-D/Z treebank (Release 11.0) is a German newspaper corpus comprising a little under 2 million tokens[18]. Not every token in the treebank has been sense annotated. The structure of the treebank is described by [Telljohann et al., 2004]. The sense annotations were added manually. Senses from GermaNet have been used to annotate a selected set of 109 words, comprising 30 nouns and 79 verbs. In total, there are 17,910 sense annotated tokens in the corpus. As the coverage of sense annotations in the treebank is not that great, the treebank was not suited for our use case.

Sense annotated corpus created by fastSense

Parts of the fastSense algorithm described by [Uslu et al., 2018] can be used to create sense annotated training data from the German Wikipedia. The algorithm uses the disambiguation pages in Wikipedia to create a sense inventory for every ambiguous word. The sense definitions in the disambiguation pages link to the corresponding Wikipedia articles. They use this internal link structure within Wikipedia to create training corpora for each sense of each ambiguous word. To date, there exist 303,643 disambiguation pages in the German Wikipedia[19]. The corpus created this way has the advantage that the sense inventory covers various topics. Because the training data for each sense is harvested from Wikipedia, more training data is available for each sense than it is the case with GermaNet. Because of the reasons mentioned in Section 3.1, this thesis focuses on the word sense disambiguation of natural language terms in legal documents. The dataset generated from the German Wikipedia has the additional advantage that the training corpus contains a vast vocabulary of natural language terms. These are the reasons why the fastSense algorithm is used for this thesis. More information on the fastSense algorithm and its implementation is given in Chapter 4.

[18] Information regarding the treebank is given at https://uni-tuebingen.de/fakultaeten/philosophische-fakultaet/fachbereiche/neuphilologie/seminar-fuer-sprachwissenschaft/arbeitsbereiche/allg-sprachwissenschaft-computerlinguistik/ressourcen/corpora/tueba-dz/. Visited on May 20, 2021.

[19] This information is provided by Wikipedia at https://de.wikipedia.org/wiki/Wikipedia:Begriffsklärung. Visited on May 20, 2021.

3. Preparation and Resources

3.1. Expert Interviews

The underlying base hypothesis of this thesis is that WSD can be applied to German court rulings to improve legal information retrieval. To acquire domain expertise and get more familiar with lawyers' methods and search strategies, semi-structured expert interviews were conducted at the start of the thesis. This was done to assess if WSD can aid legal practitioners in legal information retrieval tasks, and if so, which applications are most attractive for them. Six lawyers from different age groups (between **26** and **55**) and different legal areas were interviewed. The interviews consisted of two parts. The first part covered legal research in general and the second part went more in-depth on use cases for WSD in law. The interview guideline can be found in Appendix B.

The most relevant key findings were:

- On average, the interviewed people spend 5.25 hours/week on legal research. However, this fluctuates strongly throughout the year. Some of the interviewees mentioned that there are weeks where they do nothing but legal research, but there are also weeks where they do not conduct any legal research at all.
- When using German online legal databases, too many irrelevant search results are retrieved, even when working with the provided filters.
- It is challenging to find relevant results when the exact legal keywords are not known. Spelling and wording have too much influence on the retrieved results.
- There exist ambiguous legal terms. However, they exist only across different legal domains. Within a particular law domain, each legal term and legal keyword has a unique meaning.
- A word sense filter for legal terms would be similar to already existing filters that can filter results for different legal areas. For example, the German word "Berufungsgrund" can either be meant as reason for an appeal within the context of a civil process, or be meant in the sense of why someone is appointed to inherit something from a deceased person. However, the senses appear in two distinct legal areas, either within civil law or within inheritance law. Lawyers already filter results based on legal areas, which gives them more accurate results than a sense filter would for this use case.
- A sense filter for natural language terms would be helpful.
- The majority of the interviewees preferred that the possible word senses of words in their search query are already displayed during the query input into the search bar.

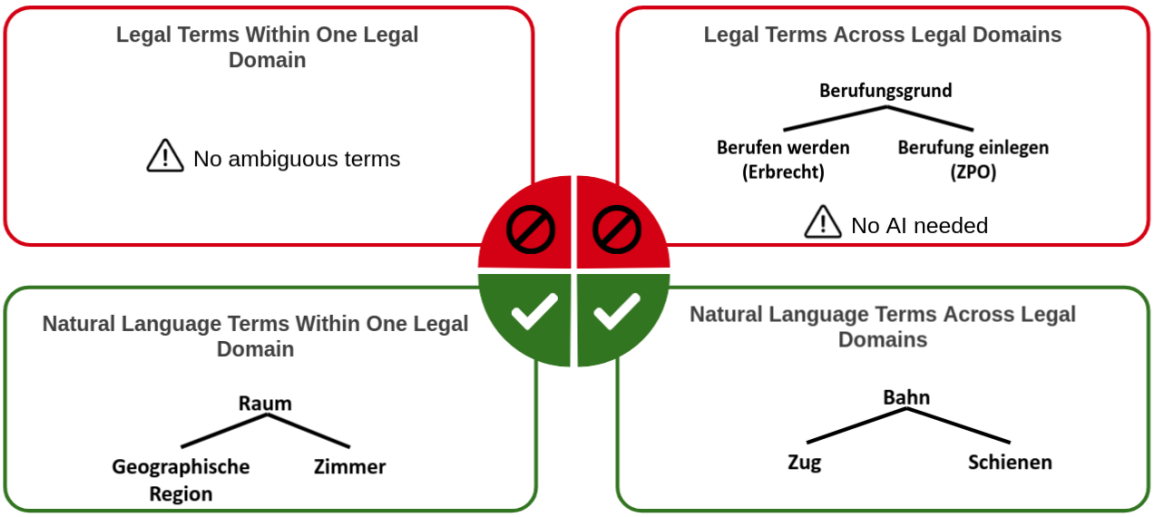


Figure 3.1.: Our four possible use cases for the German language where WSD can be applied to. The disambiguation of legal terms within one legal domain has no use case, as no ambiguous legal terms exist within a single law area. The disambiguation of legal terms across legal domains was judged by the interviewed legal experts as not beneficial. Such a filter would have the same effect as using a filter for the legal area. The provided example makes this clear: the German word "Berufungsgrund" has different meanings across different law domains, such as in inheritance law ("Erbrecht") or in the code of civil procedure ("Zivilprozessordnung"). The disambiguation of natural language terms was judged by legal experts as being helpful, which is why this thesis will focus on the disambiguation of natural language terms within one legal domain and the disambiguation of natural language terms across legal domains.

Based on the interview results, this thesis will focus on the sense disambiguation of natural language terms that appear in German court rulings. In contrast to the initial belief, the sense disambiguation of legal terms with the help of artificial intelligence provides no additional benefit over the already existing "legal-area-filters". Because the legal area information is contained in the metadata of each legal document, a "legal-area-filter" can resolve ambiguous legal terms with 100% accuracy. A machine learning based sense filter would perform worse for this particular use case. However, the disambiguation of natural language terms was judged as beneficial by the experts. We distinguish four possible use cases for our word sense filter. They are depicted in Figure 3.1. A concrete use case is depicted in Figure 3.2. This is a genuine example of a search query that one of the interviewees has recently used.

Figure 3.3 shows a use case diagram that depicts the possible interaction with the legal information retrieval system that uses a word sense filter. Based on the feedback from the interviewees we know that the majority of legal practitioners would prefer to select the sense information of each word via a drop down menu. They want to get the option to filter results based on sense information for each ambiguous word in their search query. Based on the interviews for the qualitative evaluation in Chapter 6, we also know that they

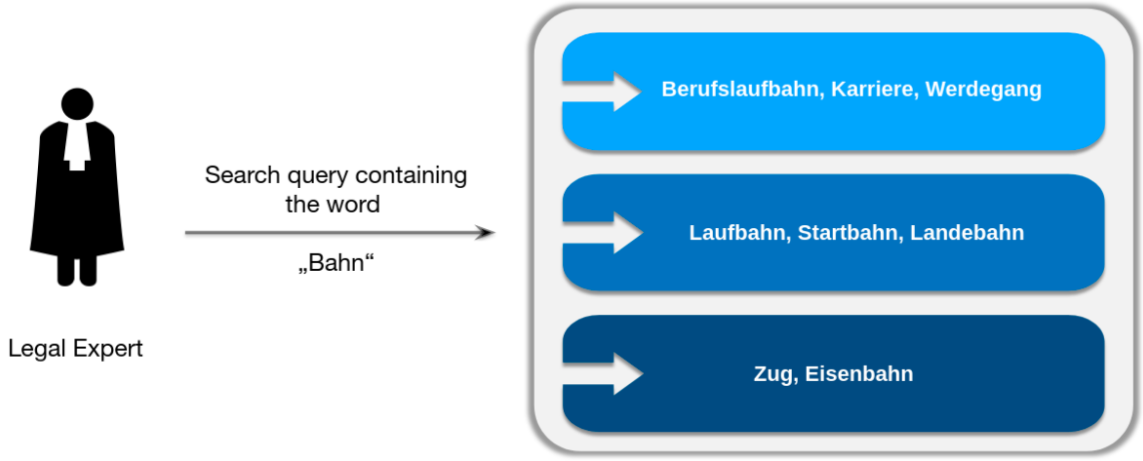


Figure 3.2.: Example of a German legal information retrieval task. A sense filter for natural language terms, such as the German word "Bahn", was judged as beneficial by the interviewed legal experts. Depicted on the right are some of the possible senses for the word "Bahn". Legal experts prefer to be informed about possible senses of ambiguous words already during the search query input into the search bar of the legal search engine.

prefer being able to select multiple senses at once. Because we discovered that only the disambiguation of natural language terms that appear in legal texts is useful for lawyers, the sense disambiguator will be trained to distinguish senses of natural language terms. For this purpose, we take the sense information from the disambiguation pages[1] of the German Wikipedia. The sense information will be used to mark ambiguous words in legal texts. This process is described in detail in Section 3.3.

3.2. Legal Dataset

A legal corpus is provided by the legal publisher Dr. Otto Schmidt KG.[2] The dataset contains 56,606 court rulings from 1980 different courts in Germany. Each court ruling is given in XML file format. Statistical parameters regarding the number of words in the text segments in the court rulings are given in Table 3.1.

Out of the 56,606 files, 448 files share their name with another file in the dataset. Out of those, 434 share their filename with one other file, 14 share their name with two other files. Files with the same filename are related to the same case. There are cases where two files that share their name are duplicates, but there are also cases where they have different content. As the duplicates make up less than 0.08% of the whole dataset, it is assumed that their impact on further analyses can be neglected, and they are left in the dataset as they are.

[1]More information about Wikipedia's guidelines for resolving ambiguous words can be found at https://en.wikipedia.org/wiki/Wikipedia:Disambiguation. Visited on July 09, 2021.
[2]https://www.otto-schmidt.de/. Visited on Apr. 16, 2021.

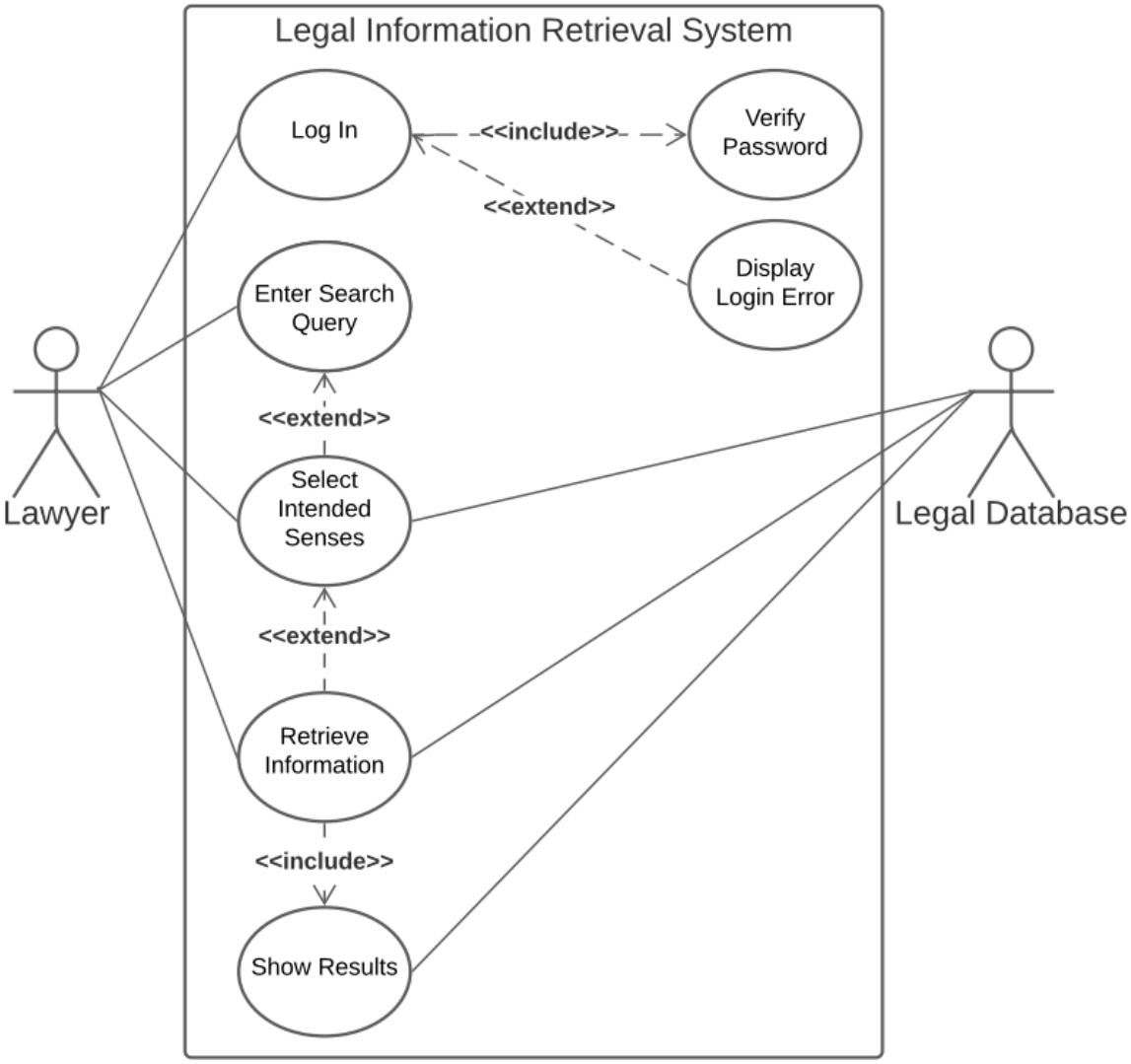


Figure 3.3.: Use case diagram for the interaction with a minimalistic legal database that uses word sense filters. Based on the feedback from the interviewees we know that the majority of legal practitioners would prefer to select the sense information of each word via a drop down menu.

The histogram in Figure 3.4 depicts the distribution of the different court types that appear in the dataset. In total, there are 26 different court types, with court rulings from the OLG ("Oberlandesgericht") making up more than a third of the documents in the dataset. Three of the court types only appear one time each in the whole dataset.[3] The histogram in Figure 3.5 illustrates the distribution of the publication date of the documents in the dataset. The dataset comprises documents from 1955 to 2017. However, there exist years for which there are no documents in the legal corpus.

Figure 3.6 depicts the distribution of legal decision types in the dataset. In German law, there exist three types of legal decisions: judgements (dt. Urteil), court orders (dt. Beschluss) and injunctions (dt. Verfügung).[4]. In this thesis, all legal decision types are mapped into one of those three. For example, the type "Gerichtsbescheid" counts as judgement because it has the same effect as a judgement under German law[5].

	Max	**Min**	**Mean**	**St. Dev.**	**25th Pctl.**	**Median**	**75th Pctl.**
Tokens	50,626	3	1,989	2,011	764	1,459	2,574

Table 3.1.: Token statistics of the text segments in the legal dataset.

Figure 3.7 shows the distribution of the ambiguous word tokens within the legal corpus. For this, each lemmatized word token in the legal corpus is matched against a list of ambiguous words. The database containing ambiguous words is created by the algorithm described in Chapter 4 of the thesis. Ambiguous multi-word tokens in the database are almost always proper names, so only unigrams are considered for the statistic. Tokens that are stopwords[6], or that are less than three characters long are also not considered. Taking this into account, there are in total 23,920 unique ambiguous word tokens in our legal corpus. More than 50 % of the tokens appear less than 10 times each throughout the whole dataset. Figure 3.8 shows that the total number of times a token appears in the legal corpus correlates strongly with the number of different documents in which these tokens appear. However, the German word "Kläger" is an outlier. As the sense data used in this thesis is generated by extracting sense information from the Wikipedia disambiguation pages, the word "Kläger" is only classified as ambiguous because of individuals who share the same last name "Kläger".[7] A possibility to solve this issue would be to filter out all senses from the database that only refer to proper names. However, this lies outside the scope of this thesis.

[3]Court types that appear in the dataset (in descending frequency order) are: 'OLG', 'FG', 'LAG', 'BFH', 'BGH', 'BVerwG', 'BAG', 'KG', 'BVerfG', 'LG', 'ArbG', 'AG', 'BSG', 'EuGHMR', 'VG', 'AGH', 'OVG', 'VGH', 'EuG', 'SG', 'LSG', 'VerfGH', 'EuGH', 'Kantonsgericht', 'GmS-OGB', 'OGH'. Explanations regarding the abbreviations can be found at https://beck-online.beck.de/Dokument?vpath=bibdata%2Fkomm%2Fbeckok_str_6%2Fcont%2FBeckOK.AbkVerzSozR.C2.htm. Visited on Apr. 16, 2021.

[4]https://www.gesetze-im-internet.de/zpo/__160.html. Visited on Apr. 16, 2021.

[5]https://www.buzer.de/s1.htm?g=FGO&a=90a. Visited on Apr. 16, 2021.

[6]Stopwords are taken from https://github.com/stopwords-iso/stopwords-de/blob/master/stopwords-de.txt. Visited on Apr. 16, 2021.

[7]https://de.wikipedia.org/wiki/Kläger_(Begriffsklärung). Visited on Apr. 16, 2021.

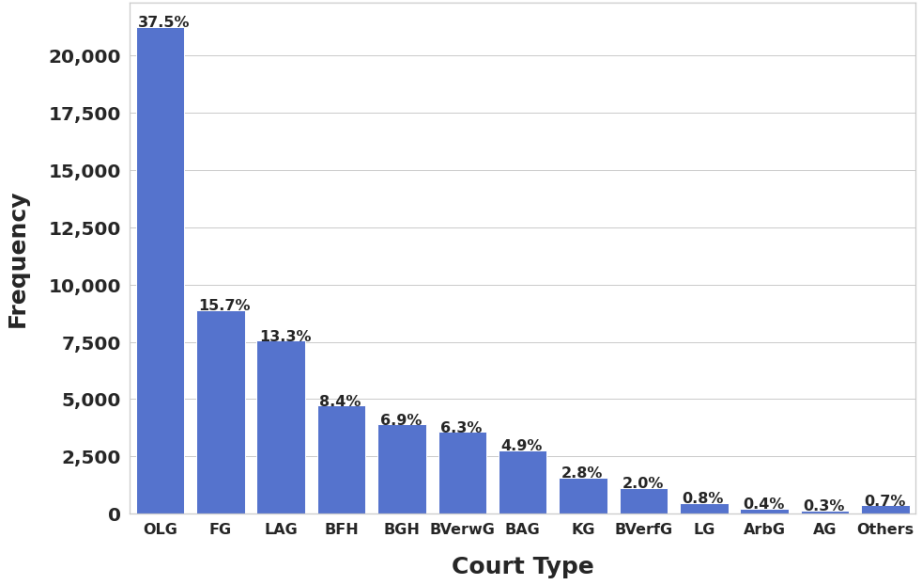


Figure 3.4.: Distribution of the different court types from the dataset. There are 26 different court types in total. The label "others" comprises the remaining 14 court types.

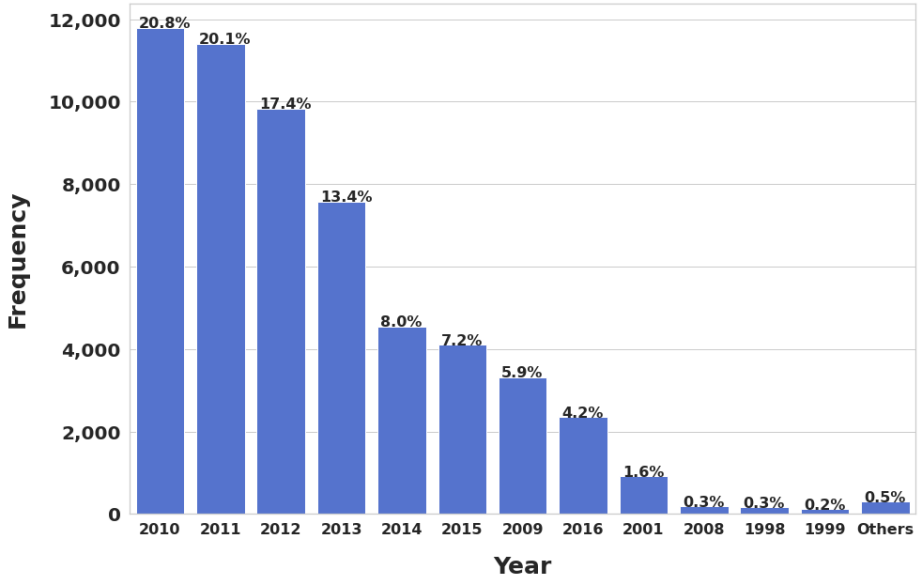


Figure 3.5.: Distribution of the years in which the court rulings were published. The years range from 1955 to 2017, however not every year is present as there are some years from which there are no documents in our legal corpus.

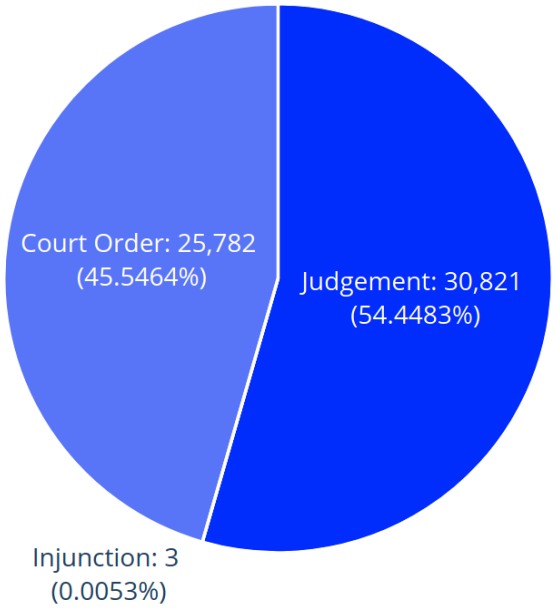


Figure 3.6.: Distribution of legal decision types. In German law, there exist three types of legal decisions, namely judgements, court orders and injunctions. Other decision types that appeared in the dataset were mapped to the legal decision type that best matched them.

3.3. Legal Test Set for WSD Classifier

Manual Sense Tagging

There is currently no sense labelled legal dataset publicly available. To evaluate our trained WSD classifier on a sense tagged legal dataset, a test set comprising **3,826** manually sense annotated words was created. This task was outsourced and the words were manually tagged by a German clickworker. Because ambiguous natural language terms were tagged, it was not necessary that the clickworker had a legal background. The only requirement was that the clickworker possesses a high proficiency in German. The clickworker was provided with a sense database that contains all the ambiguous words our sense classifier can distinguish, as well as the possible senses for each word. The senses are taken from the sense database created by the algorithm described in Chapter 4 of the thesis. The structure of the database is shown in Figure 4.6. In addition, an Excel sheet that contained randomly extracted text segments from the legal corpus described in Section 3.2 was provided. All ambiguous words in the text segments were marked with <*disambig*> tags. This was done automatically by lemmatizing[8] each word token in the text segments and checking the word tokens against ambiguous words in the sense database. The clickworker then marked the correct sense of the word by adding the appropriate sense ID of the word into the opening tag. A number between 1 and 5 was added as additional information into the closing tag to indicate how confident the clickworker is with the decision, with 5 being

[8]The words were lemmatized because all ambiguous words that are stored in the sense database are stored only in their lemmatized form.

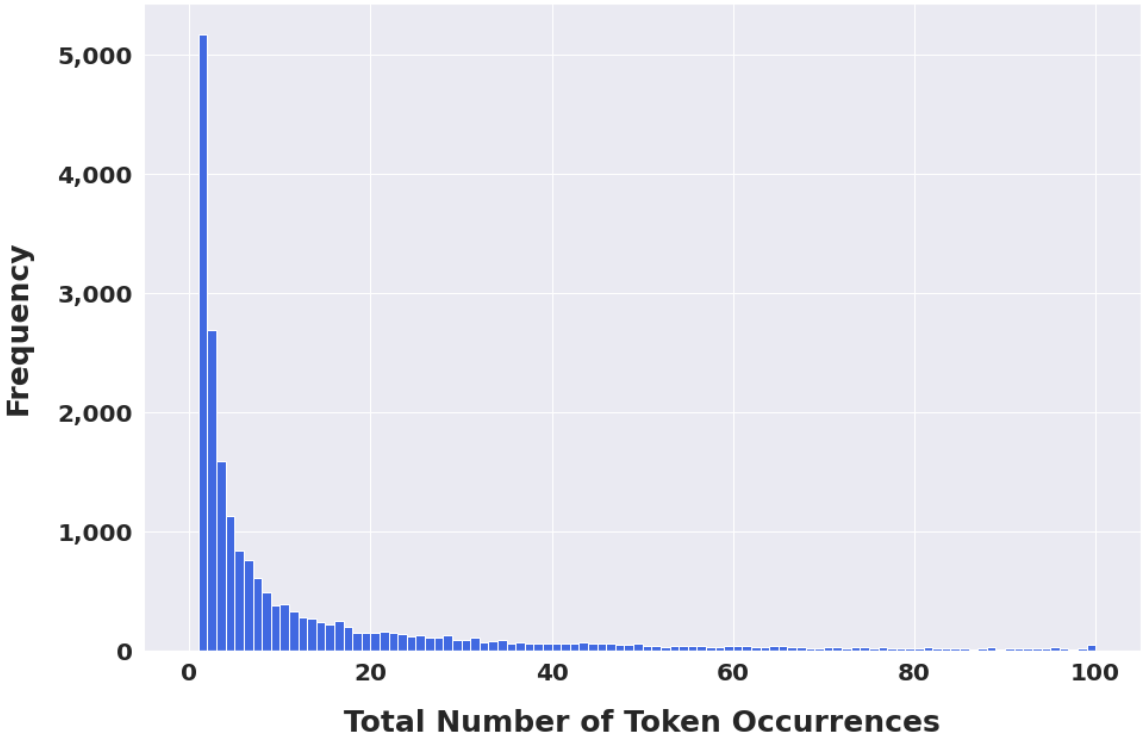


Figure 3.7.: Sense distribution in the legal corpus. The histogram shows how many ambiguous tokens occur how often in total in the legal corpus, e.g. around 5,200 Tokens occur only one time each in the whole corpus. A bin size of 1 is chosen for this histogram. The x-axis is cut at 100 for reasons of clarity. However, some of the ambiguous tokens in the dataset occur more than 100,000 times in total.

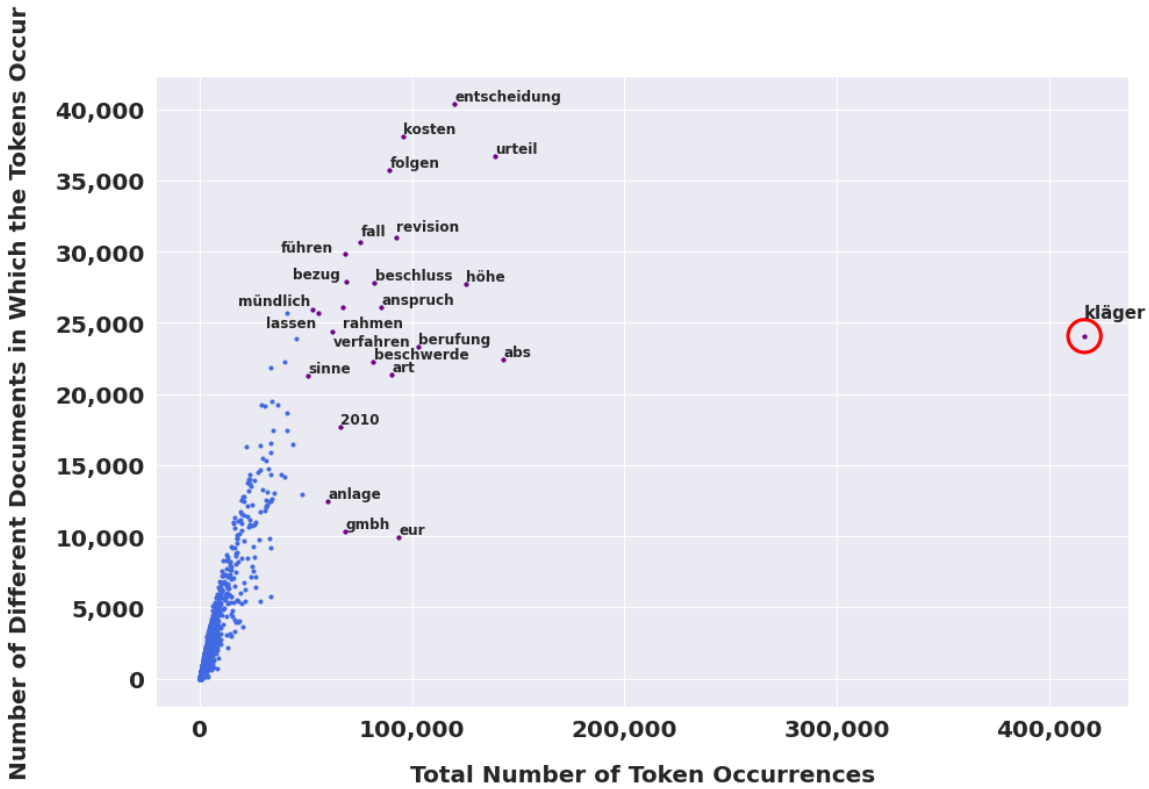


Figure 3.8.: Correlation between total token occurrences and the number of documents in which the tokens occur. Each token is an ambiguous word that has multiple senses according to our sense inventory. The more often a token occurs in the corpus in total, the more documents the token appears in. This shows that there are no tokens with a high total token count that appear in only a small number of documents. The German word "Kläger" is an outlier and is circled in red.

very confident and 1 being very unsure. Below, an example of this process is provided. A sentence from the German legal corpus where all ambiguous words are marked looks like the following.

> Das <disambig> Gesetz </disambig> über das Bundesverfassungsgericht hat zwar in § 64 Organteile ermächtigt, Rechte des <disambig> Organs </disambig> selbst geltend zu machen, ihnen also insofern Aktivlegitimation verliehen.

Subsequently, the sense tagged sentence looks like the following.

> Das <disambig_5458> Gesetz </disambig_5> über das Bundesverfassungsgericht hat zwar in § 64 Organteile ermächtigt, Rechte des <disambig_16332> Organs </disambig_5> selbst geltend zu machen, ihnen also insofern Aktivlegitimation verliehen.

Because of the reason stated in Section 3.2, sense labels for "Kläger" are not contained in the test set. There are multiple ways to sample a test set out of our legal corpus. It can be sampled either by randomly selecting whole documents or by randomly selecting individual text segments from the corpus. Both methods result in a test set with a sense distribution similar to the sense distribution of the whole corpus. The sense distribution of the document sampled test set is depicted in Figure 3.9. Figure 3.10 shows the sense distribution of the paragraph sampled test set.

We decided to create the test set by randomly selecting text segments comprising at least 1,000 characters. This leads to a court type distribution similar to the one of the whole dataset. This also has the advantage that a greater variety of words is contained in the test set. Some words might solely appear in documents of a particular court type (for example, some words related to tax-related matters might appear exclusively in documents from a "Finanzgericht"). The test set's sense and court type distribution are depicted in Figure 3.10 and Figure 3.11, respectively.

The sense labels can be looked up from a database, whose structure is explained in Chapter 4. In addition to the normal sense labels from the sense database, labels for exceptional cases are introduced in the test set. This was needed as there are cases where a sense label cannot be clearly identified. These labels are *"NV"*, *"RW"*, *"EN"*, *"AK"*, and *"?"*. The label *"NV"* is used in case the correct sense is not contained within the database. *"RW"* is an indication of an idiom or figure of speech. *"EN"* marks proper names, and *"AK"* indicates that something is an abbreviation. Lastly, the label *"?"* indicates that the tagger did not understand the meaning of the word or does not know the word. Figure 3.12 shows the proportion of the different label types in the manually labelled test set.

Inter-Rater Reliability

To measure inter-rater reliability of the manually labelled test set, a subset of the test set was taken. An independent second person manually labelled the ambiguous tokens in that subset. This was done to estimate the chance that two independent annotators make the same sense annotation decision for ambiguous tokens. A strong, robust measure for inter-rater reliability is Cohen's kappa. It was introduced in the seminal paper [Cohen,

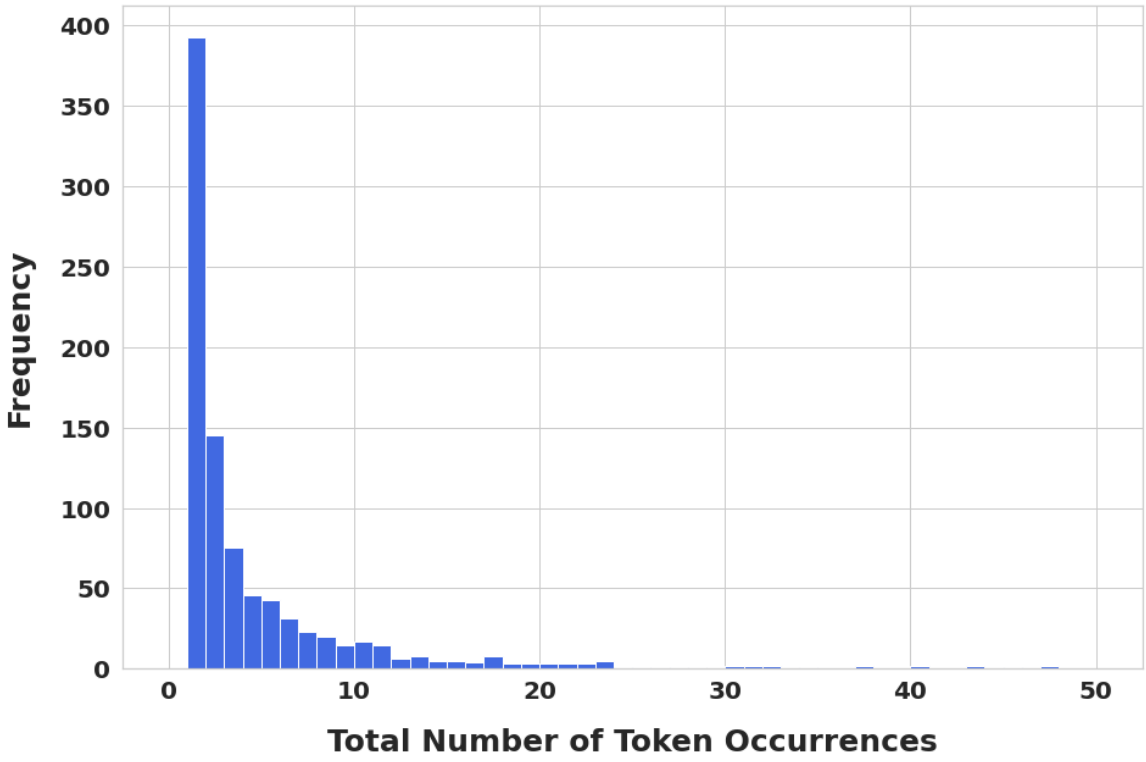


Figure 3.9.: Sense distribution of the document sampled test set. The histogram shows how many ambiguous tokens occur how often in total in the legal corpus, e.g. around 500 Tokens occur only one time each in the test set. A bin size of 1 is chosen for this histogram.

1960]. In contrast to a simple percentage agreement, Cohen's kappa considers random chance agreements of two people annotating a dataset. However, Cohen's kappa does not support multi-label input. We argue that a percentage agreement is a robust measure for inter-rater reliability in our case, as each token can have multiple possible senses, which makes a random chance agreement between two independent annotators unlikely. On average, there are **6.02** possible senses for each ambiguous token. The subset we take contains **375** ambiguous words, which make up **9.80%** of the manually tagged test set. Out of the 375 labels, the annotators agreed in 261 instances and disagreed in 114 instances. This results in inter-rater reliability of **69.6%** for the manually tagged test set.

Words where the annotators disagreed were words like *"Darlehen"* (en. loan), that, according to our sense database, has the senses *"Kredit"*, *"Darlehen_(Deutschland)"*, *"Darlehen_(Österreich)"*, and *"Mutuum"*. For annotators, it would not be precisely clear which label to use. The label *"Kredit"* is a very general term that can describe the word *"Darlehen"*. The sense label *"Darlehen_(Deutschland)"* focuses more on the legal aspects involved when taking out a loan in Germany. There are cases where both labels describe the context of the word, as one is a more general term and the other one is a term that focuses on the legal aspects of the word. So one reason for low inter-rater reliability is that some contexts overlap. It is not always clear which label best describes the meaning of the word in the given context. Another reason for low inter-annotator reliability are errors in the sense database. They can result from internal forwarding within Wikipedia, as the sense database is created from the German Wikipedia. For example, the ambiguous word *"Anlage"* can have, amongst others, the possible sense *"Addendum_(Schriftwesen)"* or *"Addendum"* according to the sense database. The Wikipedia article for *"Addendum_(Schriftwesen)"* redirects to the article for *"Addendum"*, so those two senses describe, in fact, the same meaning of the word *"Anlage"*. However, annotators have to choose between one of the senses. This leads to instances of mismatching annotations.

3.4. Overview of Available Resources

To conclude this chapter, an overview of the resources that are used for this thesis are presented in Table 3.2.

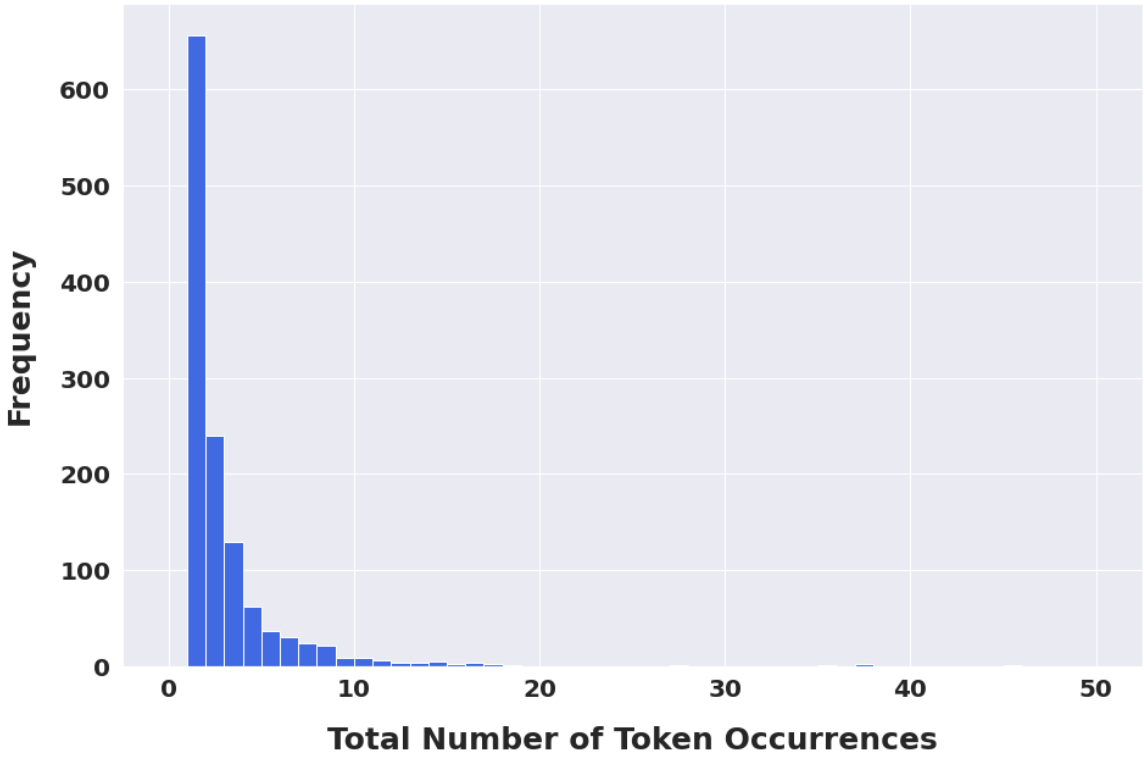


Figure 3.10.: Sense distribution of the test set. The histogram shows how many ambiguous tokens occur how often in total in the legal corpus, e.g. around 500 Tokens occur only one time each in the test set. A bin size of 1 is chosen for this histogram.

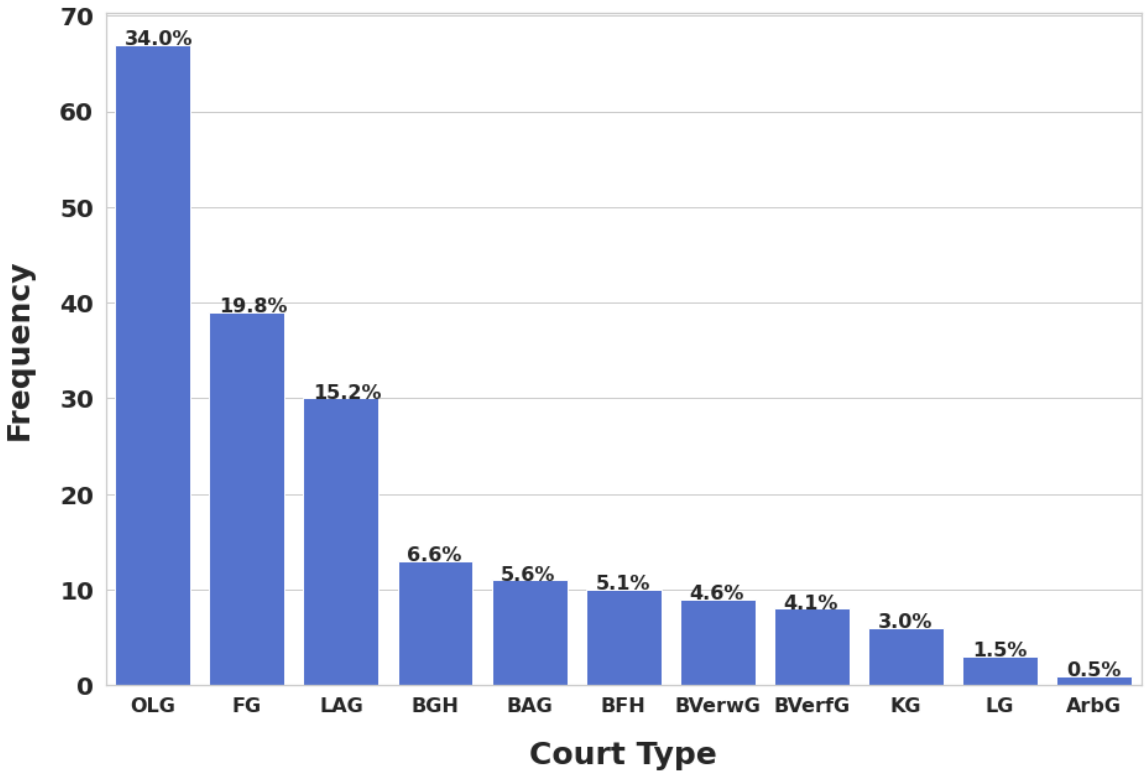


Figure 3.11.: Distribution of the different court types from the test set. There are 26 different court types in total, but not every court type from the whole legal corpus is contained within the test set.

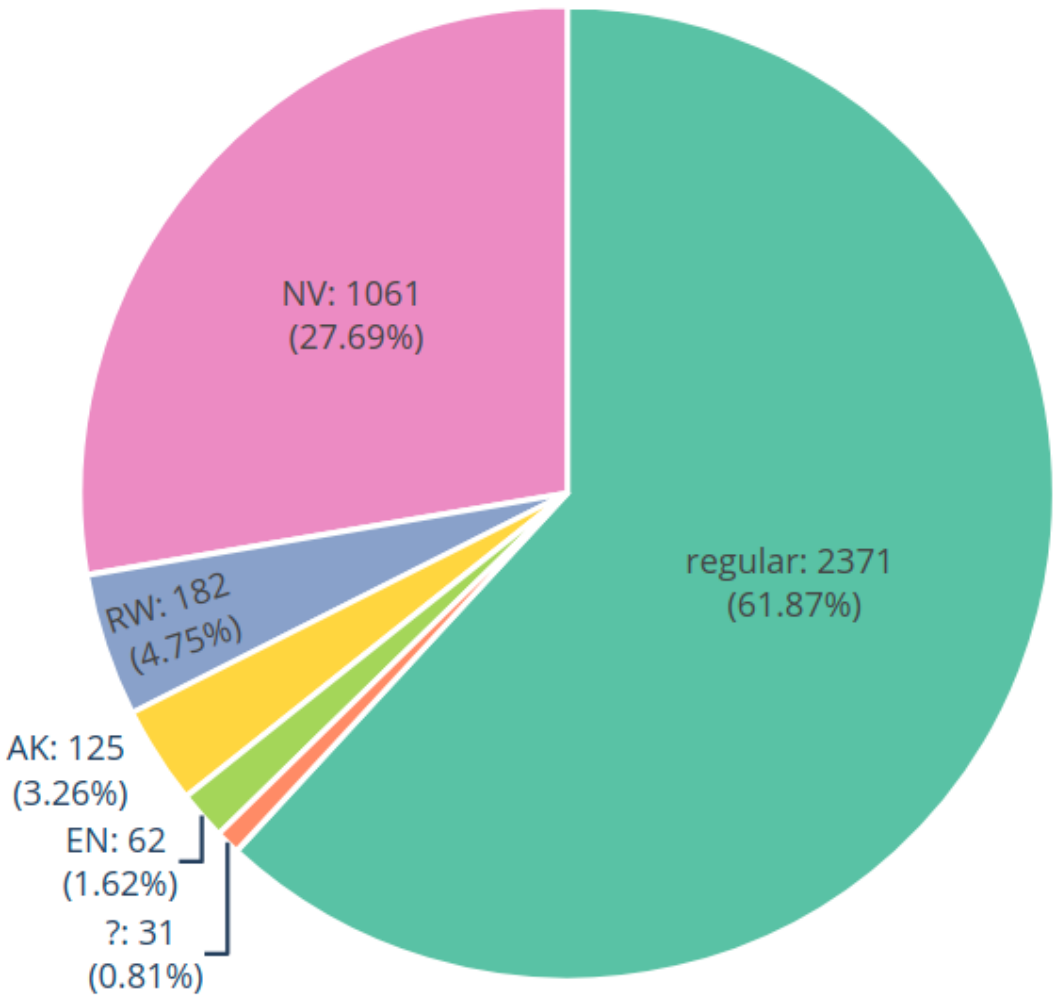


Figure 3.12.: Label distribution of the manually tagged test set. In total, there are 3,826 sense annotated words contained in the test set. In addition to the normal sense labels from the sense database, labels for exceptional cases are introduced in the test set. The label "NV" is used in case the correct sense is not contained within the database. "RW" is an indication of an idiom or figure of speech. "EN" marks proper names, and "AK" indicates that something is an abbreviation. The label "?" indicates that the tagger did not understand the meaning of the word himself. In the pie char, "regular" indicates the proportion of the normal sense labels found in the sense database.

Resource	Short Description
Legal Corpus provided by the legal publisher Dr. Otto Schmidt	A legal corpus in XML format that contains court rulings from different courts in Germany. This corpus is not used for the training of the WSD classifier but to analyze token statistics and extract a test set that is manually labelled with sense data.
Manually Tagged Legal Test Set	A small test set extracted from the legal corpus by Dr. Otto Schmidt. The test set comprises paragraphs that contain 3,826 manually labelled ambiguous words. For the labels, an inter-annotator agreement of 69.6% between two independent annotators was calculated.
Training Set and Sense Database	The training data for the WSD Classifier comprises paragraphs from the German Wikipedia. The sense groups and sense labels are stored in a database. The layout of the database and the structure of the training set are explained in Chapter 4.

Table 3.2.: Overview over the resources that are used for this thesis.

4. Implementation

4.1. Key Idea Behind the Implementation

Following the interview results described in Section 3.1, we focus on creating a sense classifier that can distinguish natural language terms. For this we require a sense inventory and a sense tagged corpus to train a sense classifier on. Different WSD resources for the German language that were considered for this thesis are listed in Subsection 2.2.6, together with their advantages and disadvantages. Because we want to be able to distinguish a great variety of different senses, the most suitable option for our use case is to use the sense annotated corpus that is created by the fastSense algorithm. The underlying idea behind the algorithm is to make use of the different word meanings that are listed on the disambiguation pages of Wikipedia. We use the word meanings to create a sense inventory. To get text examples that contain words with specific senses, we take paragraphs from the corresponding Wikipedia article. If we assume, for example, that every instance of the word "bank" within the Wikipedia article of "bank" as financial institution always has the sense "financial institution", we can create our own sense tagged dataset. With this, we have the necessary building blocks to create a sense database and sense tagged textual training examples to train a sense classifier.

To test the performance of our sense classifier on legal texts, we need a sense tagged legal corpus. However, to the best of our knowledge, there exists no German sense tagged legal corpus that is freely available. The only option is to manually tag a legal corpus with word senses from the sense database that is created with the help of Wikipedia. The manual sense tagging procedure is described in detail in Section 3.3. Moving forward, we can subsequently test the performance of our sense classifier on legal texts. With the trained sense classifier, we can sense tag ambitious words within legal texts with their corresponding sense and automatically create sense-tagged legal corpora.

This should make the intuition behind the implementation of our sense classifier more clear. Figure 4.1 visualizes the above described workflow of generating a sense inventory together with sense annotated text examples for each sense from Wikipedia. Figure 4.2 shows that the created sense inventory can be uses to manually sense annotate the legal corpus that is provided to us. The manually annotated legal corpus can subsequently be used to fine-tune the sense classifier and to measure its performance on legal texts. With the fine-tuned sense classifier, sense annotated legal corpora can be generated from raw legal corpora. This process is visualized in Figure 4.3.

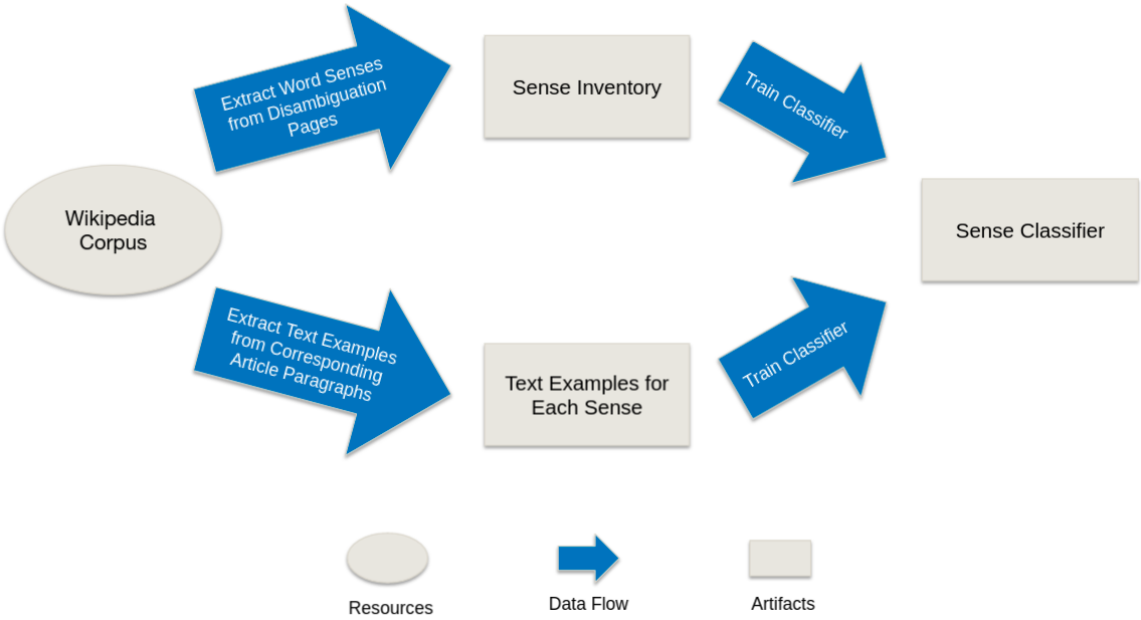


Figure 4.1.: Graphical overview of the training data generation process to train a sense classifier. The disambiguation pages of Wikipedia contain the different word meanings of ambiguous words. Those word meanings are used to create a sense inventory. To get text examples that contain words with a specific sense, paragraphs are taken from the corresponding Wikipedia articles. For this, it is assumed that, for example, every instance of the word "bank" within the Wikipedia article of "bank" as a financial institution always has the sense "financial institution. Under these assumptions, a sense tagged training set can be automatically generated.

4.2. Sense Classifier Overview

The implementation of the sense classifier is based on the fastSense model, which was first presented by [Uslu et al., 2018]. The intention behind their model was to provide a neural network-based word sense classifier that can process large amounts of data in a relatively short time. For their model, they created a sense annotated corpus out of the German Wikipedia. Every ambiguous target word that is listed on a disambiguation page in Wikipedia is linked to its corresponding article. From the article, they took each paragraph that contains the corresponding target word. They annotated the target word in the paragraph accordingly. With this, they could extract over 50 million training samples from the German Wikipedia. The different WSD resources for the German language considered for this thesis are listed in Subsection 2.2.6, together with their advantages and disadvantages.

The classifier itself consists of a single hidden layer used to predict the sense of the chosen target words. In addition, a post-processing step is applied. As the neural network's output contains all possible senses, it is not guaranteed that the top-ranked output sense is

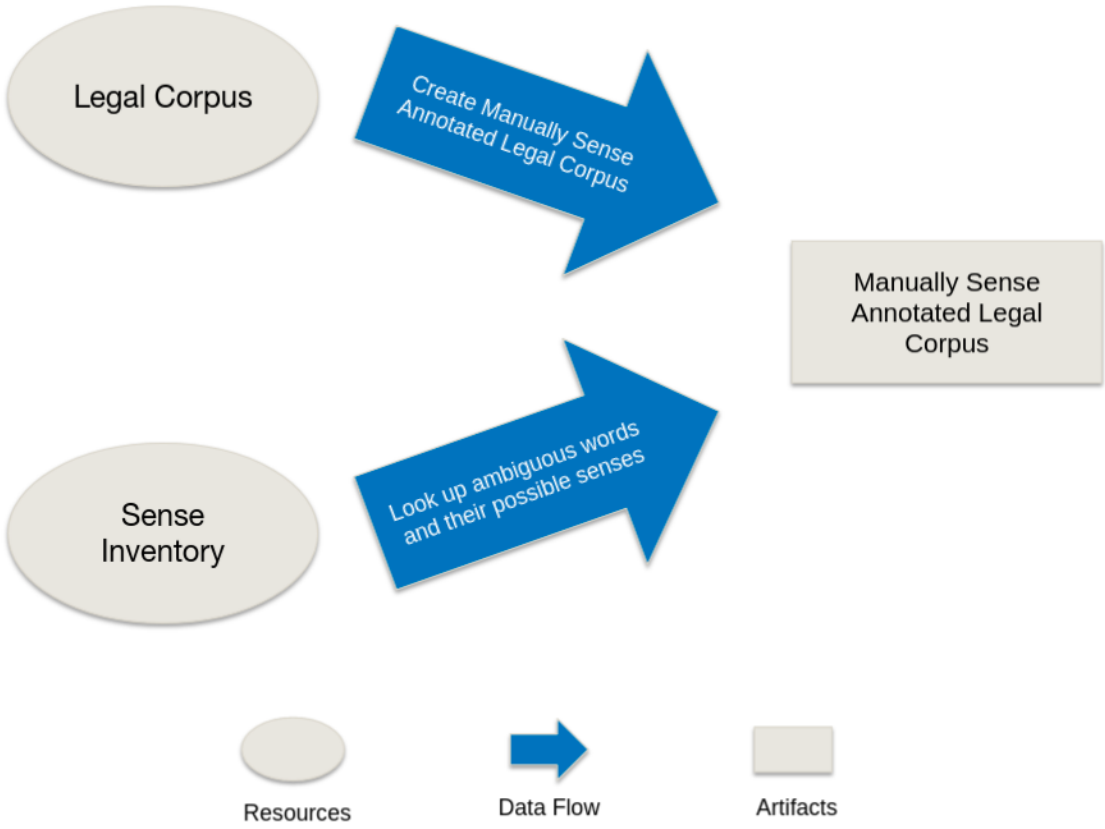


Figure 4.2.: The sense inventory generated from Wikipedia can be used to manually sense annotate a legal corpus.

a valid sense of the target word. So instead, the actual target word sense with the highest probability is taken. For example, as the word *bank* is likely to appear in texts related to financial topics, word senses for words like *finance* or *capital* are also predicted with high probability. Therefore, instead of the highest overall ranking sense, the highest-ranking sense for the particular target word *bank* is chosen. An example is illustrated in Figure 4.4.

[Uslu et al., 2018] state that they used a tool called TextImager for preprocessing, which is described in [Hemati et al., 2016]. However, the fastSense model has since been modified by [Schulz, 2021], who adapted the model for the English language. His implementation is self-contained and does not make use of any third-party tools like TextImager. In addition to the neural sense classifier itself, his implementation is capable of generating and pre-process its own training, validation, and test data. This is the fastText implementation that is used for this thesis. This implementation generates the training data by parsing Wikipedia dumps and subsequently trains the model with the generated data. The implementation was made publicly available[1]. We will adapt the model to the German language. The work in this chapter and the described implementation is heavily based on the

[1]The model can be cloned from https://github.com/texttechnologylab/fastSense. Visited on May 22, 2021.

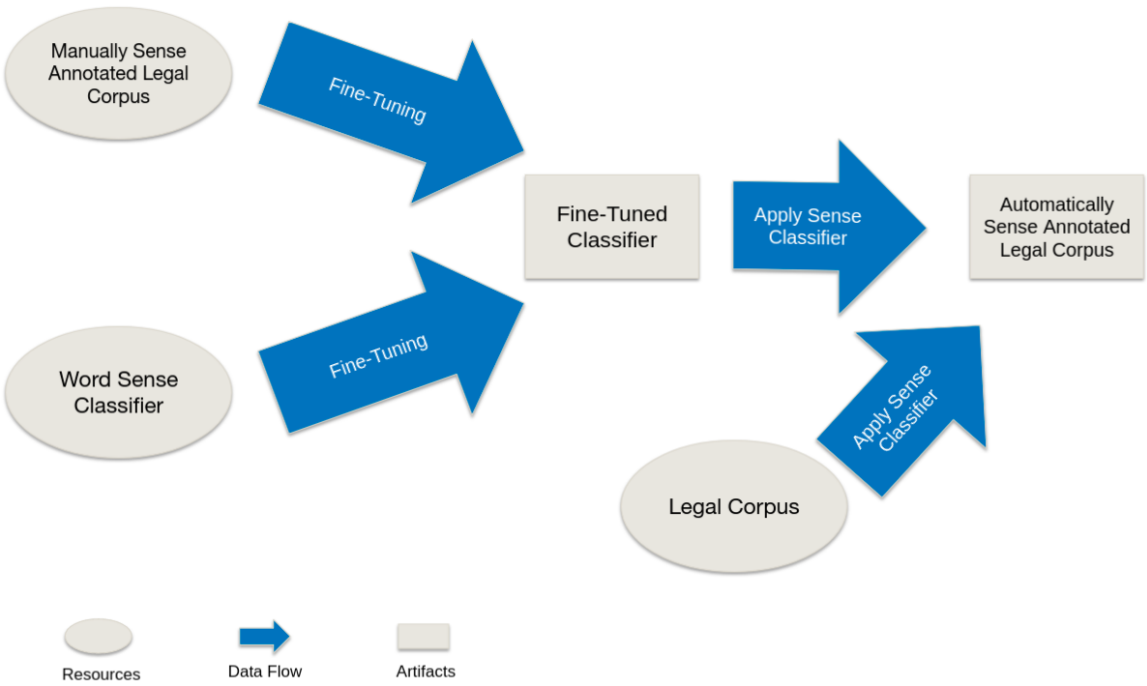


Figure 4.3.: The sense classifier can be fine-tuned with the manually annotated legal corpus. The fine-tuned sense classifier can subsequently be applied to a raw legal corpus to automatically create a sense annotated legal corpus.

implementation and work of [Schulz, 2021].

The long term goal is to provide users with a tool that can filter results based on the actual meaning of words, which they get from the context they appear in. For this purpose, ambiguous words in the underlying corpus from which information is retrieved must be identified. These words subsequently need to be tagged with the proper sense labels. Algorithm 1 describes the procedure to detect and annotate ambiguous words in the text automatically with their correct sense.

Our implementation is split into two parts, one for the data generation and one for the training of the sense classifier. The two parts work independently from each other. This has the advantage that the generated data can be used to train other classifiers, and that the classifier can be trained with data from other sources as long as the data provided to the classifier has the correct format. The algorithms are implemented entirely in Python. The implementation details of the two parts are discussed in this chapter.

4.3. Data Creation Pipeline

The implementation of the data creator is split into two parts. The pipeline consists of an extractor that extracts data from the Wikipedia dumps and an exporter that exports the intermediate output from the extractor into a structure that subsequent classification

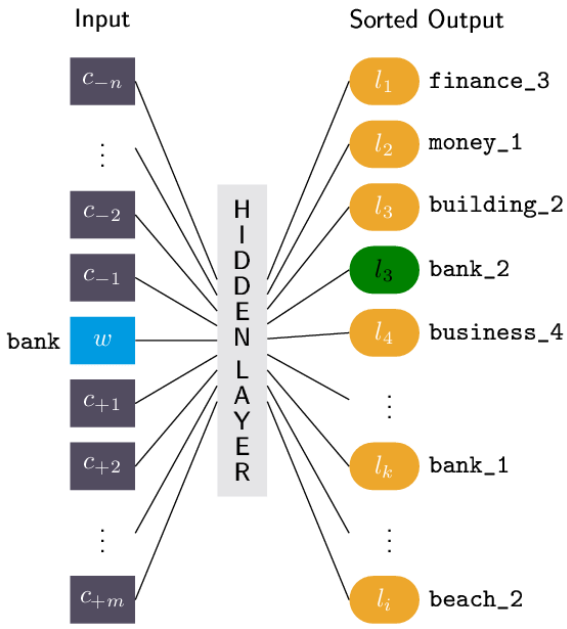


Figure 4.4.: The working mechanism of the fastSense classifier. The output contains all senses for all ambiguous words in the corpus. Since the word *bank* tends to appear in financial related topics, other senses for words that appear in such topics are predicted with a high probability. This is why an additional post-processing step is performed. The highest scoring sense output for the target word *bank* is selected instead of the overall top scoring output, which in this example is a sense for the word *finance*. Image taken from [Uslu et al., 2018].

models can use. The implementation is publicly available[2]. The script is designed to run on 6 threads simultaneously using the Python "multiprocessing" package[3]. The two parts are explained in the following.

4.3.1. Extractor

The extractor takes Wikipedia backup dumps as input[4]. The required dumps are listed in Table 4.1. The title of all pages in the dumps are standardized by the extractor. Upper case characters are converted to lower case, and every article title gets standardized by removing the German words "der", "die", "das", "ein", and "einer" when they appear as prefixes in the article title. Furthermore, parenthesis and their contents are removed if they appear after a whitespace character at the end of the article title, as they either indicate a disambiguation page or further specify the meaning of the title. For example, the titles

[2]The implementation of the data creation algorithm can be found at https://github.com/AndreasPadProbst/fastSense_CreateData. Visited on May 22, 2021.
[3]Documentation for the multiprocessing package is given at https://docs.python.org/3/library/multiprocessing.html. Visited on May 22, 2021.
[4]For this thesis, dumps from the 20.02.2021 were used. The dumps can be downloaded from https://dumps.wikimedia.org/dewiki/. Visited on May 22, 2021.

Algorithm 1 Automatic detection and labelling of ambiguous words (from [Schulz, 2021])

Input: Text that should be sense annotated
Output: Text where each ambiguous word has been sense annotated

```
Split text into paragraphs;
for each paragraph do
    Tokenize paragraph;
    Lemmatize word tokens;          ▷ sense database only contains lemmatized tokens
    Check if tokens are ambiguous;
    if at least 1 token is ambiguous then
        Process tokens;             ▷ e.g. build n-grams to preserve word order
        for each ambiguous token T do
            Retrieve the set S of all possible senses for T;
            Process the output of the sense classifier for S and T;
            Annotate the corresponding word in the paragraph with the correct sense;
        end for
    end if
end for
```

"Bank (Möbel)" and "Bank (Begriffsklärung)" both get standardized to "bank". Regarding the text content of the articles themselves, content like tables, template pages and content between certain Wikicode tags is removed to improve the quality of the text data.

Pages in the category "Begriffsklärung" (disambiguation page) resolve ambiguous page titles by listing the different meanings of the ambiguous terms and linking the terms to their corresponding article page. This internal link structure is used to find training examples for the different senses of each ambiguous word. It should be mentioned that linked articles in the section "Siehe auch" (en. "See also") of disambiguation pages are not added to the sense group as they tend to be unrelated. Senses are collected and grouped together under a collective sense group name. The name of the disambiguation page is taken as the sense group name. So each sense group represents an ambiguous word, and each member of the group represents a specific sense for that ambiguous word. The individual word senses are represented by URL links that link to the corresponding article. For example, within the sense group titled "bank" the senses that link to `https://de.wikipedia.org/wiki/Kreditinstitut`, `https://de.wikipedia.org/wiki/Bank_(Möbel)` or `https://de.wikipedia.org/wiki/Bank_(Meer)` are found.

Furthermore, the disambiguation pages are enriched with additional articles automatically by processing all links within Wikipedia. Redirections[5] are resolved, and it is checked if there are at least five articles with the same standardized article title that all link to the same page. If that is the case and the articles share their standardized title with one of the sense group titles, the link to the article is added as sense to the sense group. One article might be part of multiple sense groups. In order to uniquely specify the sense of a

[5]More on redirections can be found at https://en.wikipedia.org/wiki/Wikipedia:Redirect. Visited on May 24, 2021.

particular word, the sense label (the article link) and the sense group label are needed.

File	Description
dewiki-*-categorylinks.sql.gz	A category is a grouping of related pages in Wikipedia. The *categorylinks*[6] table stores links that indicate if a page belongs to a particular category. For example, by placing a "{{Disambiguation}}" (in German "{{Begriffsklärung}}") flag in an article in the pages-articles.xml dump, the page is added to the category of disambiguation pages[7].
dewiki-*-page.sql.gz	The *page*[8] table contains metadata for each page. Each page is assigned a unique primary key through which information like the page title, and the namespace[9] to which the page belongs to, can be looked up. Namespaces are collections of documents that have a similar purpose. For this thesis, we need documents in the "Category" namespace, as we are only interested in pages from the category "Disambiguation pages".
dewiki-*-pages-articles.xml.bz2	This dump contains the actual text content of Wikipedia in XML format[10]. It contains content from the namespace "Main", which comprises all articles in Wikipedia, and other namespaces relevant to this thesis, like the "Category" namespace.

Table 4.1.: Description of the required Wikipedia dumps.

In order to parse Wikitext[11], a preprocessing step is performed. Tables and style tags are removed, and templates are expanded. The preprocessing output is given to a "WikitextParser" class that uses the Python module *mwparserfromhell*[12] to extract paragraphs and

[6]The structure and content of the categorylinkstable is explained at https://www.mediawiki.org/wiki/Manual:Categorylinks_table. Visited on May 22, 2021.

[7]More information about the category "Disambiguation pages" is given at https://en.wikipedia.org/wiki/Category:Disambiguation_pages. Visited on May 22, 2021.

[8]The structure and content of the page table is explained at https://www.mediawiki.org/wiki/ Manual:Page_table. Visited on May 22, 2021.

[9]More on namespaces can be found at https://www.mediawiki.org/wiki/Manual:Namespace. Visited on May 22, 2021.

[10]The XML schema of the dump can looked up at https://meta.wikimedia.org/wiki/ Data_dumps/Dump_format. Visited on May 23, 2021.

[11]Wikitext is the markup language that is used for the XML dumps. A documentation for Wikitext is given at https://en.wikipedia.org/wiki/Help:Wikitext. Visited on May 23, 2021.

[12]The mwparserfromhell package provides a parser to parse Wikitext. A documentation can be found at

other meta information like the section ID of the paragraph and links contained within the paragraph. The preprocessing is performed in order to minimize errors in subsequent parsing steps with the *mwparserfromhell* parser. A UML class diagram of the preprocessor and parser class is depicted in Figure 4.5.

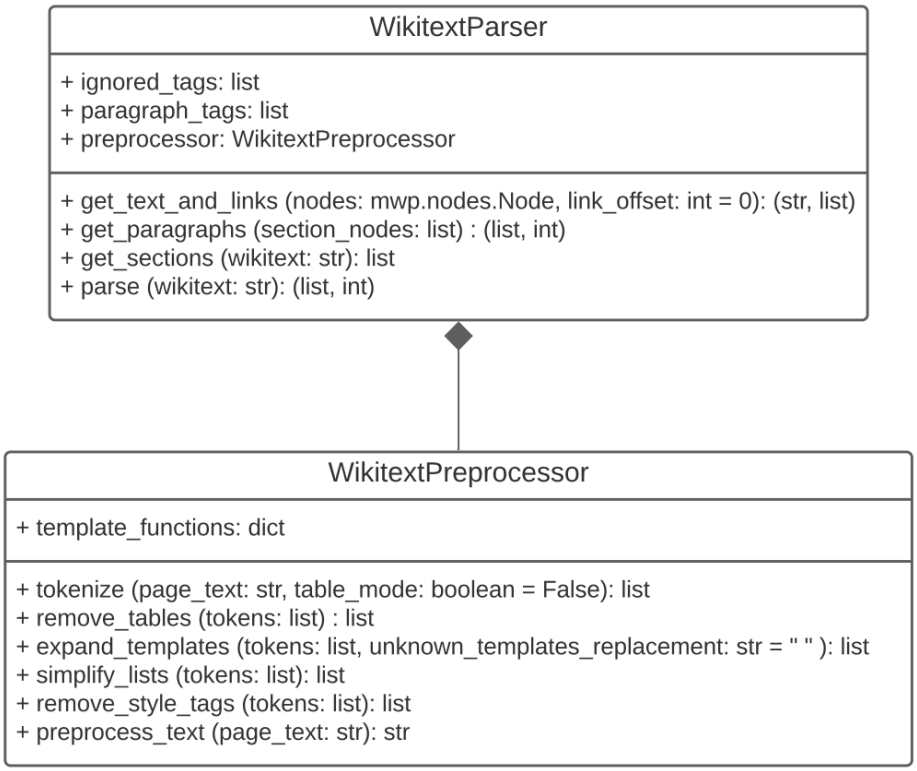


Figure 4.5.: UML class diagram of the preprocessor and parser. Preprocessing steps like removing tables and style tags are performed first in order to minimize errors in subsequent parsing steps with the *mwparserfromhell* parser. The parser class works with the nodes package from the mwparserfromhell Python library.[13] All class attributes and members are public.

4.3.2. Exporter

The exporter processes the output files from the extractor and brings the training data into a more suitable format for training. It generates a database[14] containing information about the different sense groups and the senses contained in each sense group. Figure 4.6 shows the entity-relationship diagram of the sense database. It should be mentioned that this is the database structure as it is returned from the exporter, and the database schema might not be intuitive at first. Documentation of the database schema is provided by [Schulz, 2021].

https://mwparserfromhell.readthedocs.io/en/latest/. Visited on May 23, 2021.

[14]The database for this thesis was generated with dumps downloaded on Feb 21, 2021. It can be downloaded from https://github.com/AndreasPadProbst/sense_db. Visited on May 23, 2021.

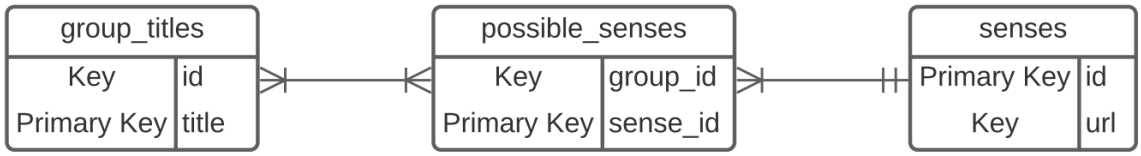


Figure 4.6.: The entity-relationship diagram of the database containing sense groups and senses. The lemmas of ambiguous words and their corresponding sense group ID are contained in the "group_titles" table. The "possible_senses" table comprises the senses that are contained inside a specific sense group. The "senses" table contains a mapping between each sense and the URL of the Wikipedia article the senses refers to. It should be mentioned that the same article can be part of different sense groups. However, for each sense group, it will get assigned a separate ID in the "senses" table so that senses of the same word that appear in different sense groups can be distinguished.

Besides generating a database that contains the sense groups and their respective senses, the exporter produces textual training data that the sense classifier can be trained with. The training data are stored as TensorFlow Records[15] (TFRs) that contain the sense ID of ambiguous words, the sense IDs of possible senses for those ambiguous words, and a tokenized text segment in which the ambiguous word appears. TFRs have the advantage that sequences of data are stored in a memory-efficient way. The format is optimized for using it together with the TensorFlow Python library. The structure and data types of the features in the TensorFlow Records, as well as a short description, are given in Table 4.2. A process flowchart of the complete data creation process is depicted in Figure 4.7.

The datasets are split into a training set for parameter tuning, a validation set for hyperparameter tuning and a test set. In total, the dataset comprises 64,274,208 sense-tagged examples. The examples are split in a 70/15/15 fashion, so the training set comprises 70% of the generated data. The validation set comprises 15% of the data. The test set also comprises 15% of the generated data. The train-val-test-split is stratified so that the examples in the datasets have the same sense distribution. Table 4.3 lists the proportions of the train, validation, and test set.

TFR Feature Name	TFR Data Type	Description
tokens	tf.train.BytesList	This feature contains the UTF-8 serialized paragraph or sentence that contains the ambiguous target word.
possible_senses	tf.train.Int64List	A list of all the possible senses for the target word.
sense	tf.train.Int64List	The correct sense of the target word.

Table 4.2.: Structure of the TFRs that are produced by the exporter.

[15]https://www.tensorflow.org/tutorials/load_data/tfrecord. Visited on July 04, 2021.

Dataset	Proportion
Train Set	70%
Validation Set	15%
Test Set	15%

Table 4.3.: Train/validation/test split ratio of the generated data for the hashing model depicted in Figure 4.8. The train set is used to update the neuron weights in the model. The validation set is used to fine-tune the hyperparameters of the model (for example, the learning rate of the gradient descent algorithm). The test set is used to measure the accuracy of the model on a dataset that it has not seen before. The whole dataset comprises 64,274,208 examples and covers 342,861 different senses. The data split is stratified, so each dataset has a similar sense distribution.

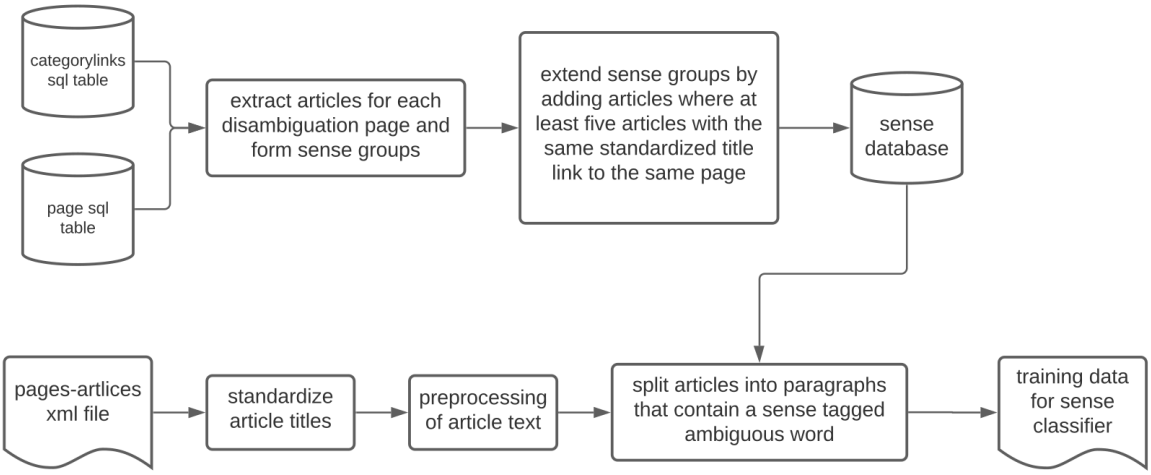


Figure 4.7.: Flowchart of the data creation process. The input files are described in Table 4.1. The sense database that is produced during the process refers to the database described in Figure 4.6. The output comprises TFRs where each row contains a paragraph that contains an ambiguous word, the correct sense label for that ambiguous word, and all the possible sense labels for that ambiguous word. The structure of the TFRs is shown in Table 4.2

4.4. Neural Model Architectures

Figure 4.8 depicts the architecture of the original fastSense model. We will refer to this model as the "hashing model" to clearly distinguish it from the other model presented in this chapter. The word tokens in the text input are hashed into integers with a deterministic hashing function provided by TensorFlow[16]. The values of the hashed tokens are embedded into low-dimensional vectors with the help of an embedding layer. The

[16]The function "tf.strings.to_hash_bucket_fast" was used. More information can be found hat https://www.tensorflow.org/api_docs/python/tf/strings/to_hash_bucket_fast. Visited on July 04, 2021.

embedded tokens are subsequently averaged in the combiner and passed through the output layer of the model. This procedure resembles the fastText method described in Subsection 2.1.2, which gave the original fastSense algorithm its name. By default, the combiner averages the tokens by dividing them by the square root of the number of tokens instead of the absolute number of tokens. The formula is given in Equation 4.1. In the formula, v_i denotes the embedding of the i-th token and n is the number of tokens in the text segment used to predict the sense of the target word. When dividing the sum of the embedded tokens by $\sqrt{n}$ instead of just n, we still keep some information about the length of the text segment that the model can use. In Chapter 5 we evaluate the model's performance and study the effect of the reduction operation by evaluating the model when using the reduction operation from equation 4.2. The model is available at `https://github.com/AndreasPadProbst/fastSense_TrainModel`. A documentation of how to use the model is given in Appendix A.

$$avg_1 = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} v_i. \tag{4.1}$$

$$avg_2 = \frac{1}{n} \sum_{i=1}^{n} v_i. \tag{4.2}$$

In this thesis, we also experiment with contextual word embeddings. Therefore, sentences containing the ambiguous target words are embedded with contextual embeddings from a pre-trained German BERT model[17]. We will refer to this model as the "fastBERT" model. The pre-trained BERT model that we use consists of 12 hidden layers comprising 109 million parameters.[18] This makes forward passes resource-intensive, which is why only a subset of the generated training and test data is used. In total, approximately 314,700 samples are taken from the original dataset, which covers approximately 32,000 senses. A stratified train-test-split of 75% and 25% is carried out. We abstain from using a validation set in this case because we have far less data available to train the fastBERT model than we had with the hashing model. When using BERT embeddings, we use sentences instead of paragraphs. A *[CLS]* and a *[SEP]* token have to be added at the beginning and at the end of the sentence, respectively. The embedding of the *[CLS]* token represents the sentence embedding. The rest of the architecture is similar to the hashing model architecture depicted in Figure 4.8, except that no token pooling has to be done and thus no combiner is used. The fastBERT model is depicted in Figure 4.9.

[17] The pre-trained BERT model can be found at https://huggingface.co/dbmdz/bert-base-german-cased and was open sourced by the Bavarian State Library. Visited on July 04, 2021.

[18] According to https://huggingface.co/transformers/v3.3.1/pretrained_models.html. Visited on July 05, 2021.

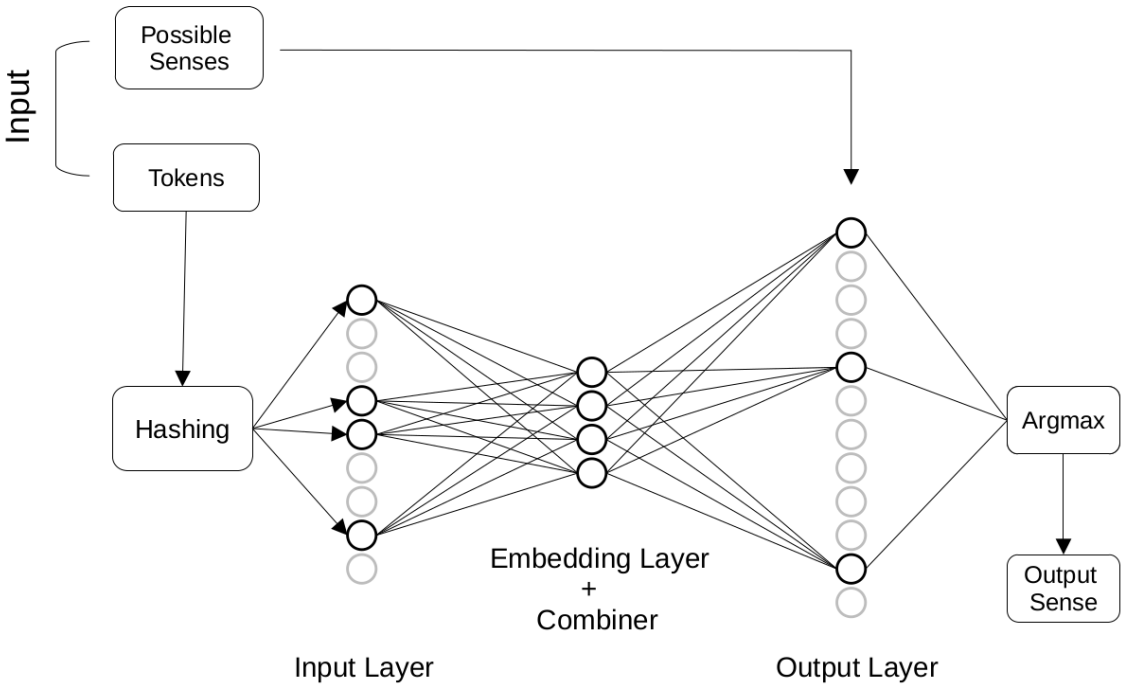


Figure 4.8.: The original fastSense architecture. We refer to this architecture as the "hashing model" to clearly distinguish it from the other model presented in this chapter. Each word token in the input text is hashed to an integer number. It passes an embedding layer to obtain a low-dimensional vector representation for each word. The embedded tokens are subsequently averaged in the combiner and passed to the output layer. The output layer uses a softmax activation function. It returns a vector of length 342,861. Each component of the vector represents a specific sense. A preprocessing step is performed where only possible senses for the target word are considered. Subsequently, the sense with the highest score is selected as the predicted sense. For the backpropagation during the training process, a gradient descent optimizer is used in combination with a cross-entropy loss function. Figure adapted from [Schulz, 2021].

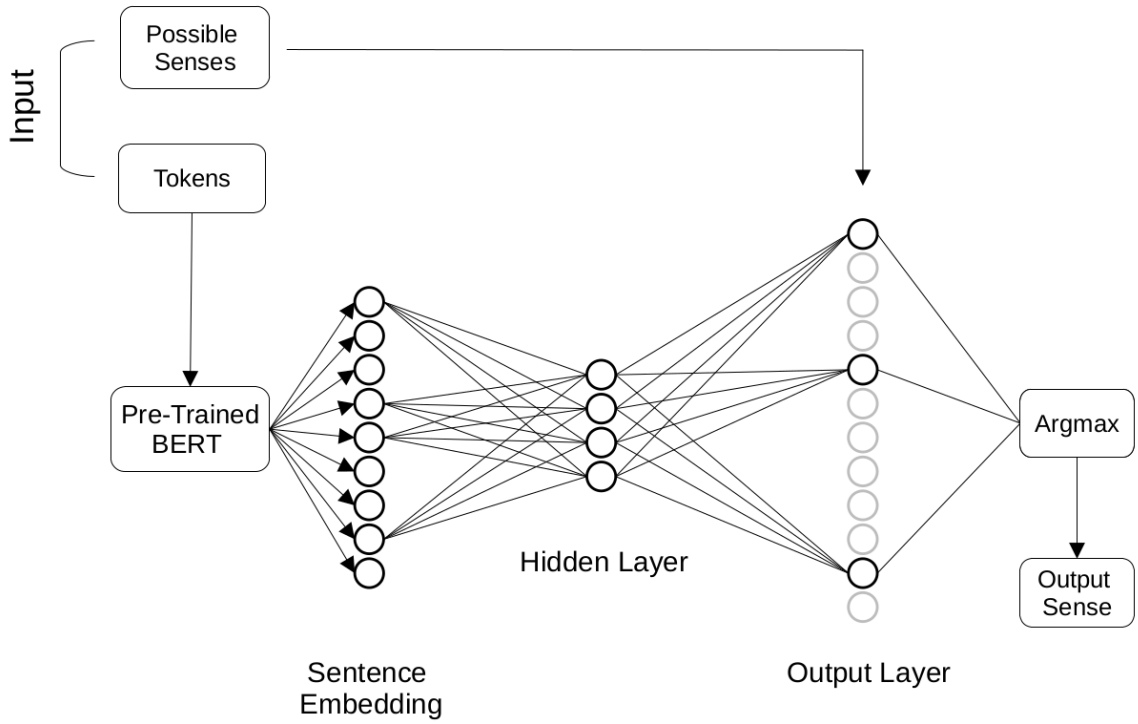


Figure 4.9.: The fastBERT model architecture. The architecture uses contextual word embeddings from BERT. A pre-trained German BERT model of type bert-base-cased was used. When using BERT for contextual embeddings, sentences were given to the model instead of whole paragraphs. The embedding of the *[CLS]* token was used to represent the embedded sentence. The hidden layer nodes had ReLU activation functions. The output layer uses a softmax activation function. It returns a vector of length 342,861. Each component of the vector represents a specific sense. A preprocessing step is performed where only possible senses for the target word are considered. Subsequently, the sense with the highest score is selected as the predicted sense. For the backpropagation, an adam optimization algorithm is used in combination with a categorical cross-entropy loss. Figure adapted from [Schulz, 2021].

Dataset	Proportion
Train Set	75%
Test Set	25%

Table 4.4.: Train/test split ratio of the generated data for the fastBERT model depicted in Figure 4.9. The test set is used to measure the model's accuracy on a dataset that it has not seen before. We abstain from using a validation set for hyper-parameter tuning because of the small amount of data that is available to us in this case. Because forward passes to create token embeddings with BERT are very resource-intensive, the whole dataset only comprises around 314,700 examples, in contrast to the 64 million examples that we passed through the hashing model. The dataset covers approximately 32,000 different senses. The data split is stratified, so each dataset has a similar sense distribution.

5. Quantitative Evaluation

5.1. Hashing Model Evaluation

In Subsection 2.2.1 we listed the different baselines and evaluation measures that are typically used when assessing stand-alone WSD systems. The "Most Frequent Sense" heuristic is a strong baseline to compare with the performance of our models. However, we do not have the most frequent sense information for the senses in our database. Due to the working mechanisms of the fastSense algorithm presented in the previous chapter, we are only able to measure the precision of our WSD model. To measure the coverage, we need information about the number of ambiguous tokens that the classifier does not detect on a given test set. Since the test set is created as part of the fastSense algorithm itself, the coverage on the test set would be 100%. We can only check the marked ambiguous tokens within the dataset generated by fastSense itself against the detected tokens by fastSense during classification, which results in a Coverage of 100%. Subsequently, it also does not make sense to measure recall and the F_1-score. For 100% coverage, the recall and F_1-score are both equal the precision. In the following, we refer to the precision as "accuracy".

We measure the accuracy of the two different fastSense architectures that we have presented in Chapter 4. We start by evaluating the performance of the hashing model that is shown in Figure 4.8. The model's accuracy is measured on the test set that we generate with the algorithm presented in Chapter 4. The train/validation/test-split ratio of the generated data is given in Table 4.3. We measure the accuracy on a "micro" and "macro" level, like it was done in [Schulz, 2021]. The "micro" accuracy is the fraction of correctly classified senses over the number of all classified senses. It is equal to the precision P introduced in Equation 2.2. It is defined in Equation 5.1.

$$micro = \frac{\text{\# correctly classified senses}}{\text{\# total classified senses}}. \tag{5.1}$$

The "macro" accuracy measures how well the model performs on average for each sense. It is a more robust measure and compensates for the fact that there are more examples for certain senses than there are for others. Consequently, senses with fewer examples get misclassified more often than senses with many examples. The skewed sense distribution in natural language is a result of Zipf's law. According to Zipf's law, the sense frequency is approximately inversely proportional to its rank in the sense frequency table of the underlying corpus. So the top-ranking senses will appear disproportionally more often than any other sense. This effect artificially increases the accuracy values calculated with Equation 5.1. The "macro" accuracy compensates this. Generally, the "macro" accuracy values will be lower than the "micro" accuracy values. The formula is given in Equation 5.2.

$$macro = \frac{1}{n} \sum_{i=1}^{n} \frac{\text{\# correctly classified instances of sense } s_i}{\text{\# total classified instances of sense } s_i}. \tag{5.2}$$

By default, the model uses the following hyper parameters:

- The square root combiner according to formula 4.2 is used.
- No dropout is used.
- The model only contains a single hidden layer with 25 nodes.
- A batch size of 32 is chosen.
- An initial learning rate of 1.0 is chosen, together with a step-based decay rate of 0.98 per 100,000 steps. The formula for the learning rate is given in Equation 5.3. In the formula, η_s is the learning rate after s training steps, η_0 is the initial learning rate, and δ is the decay rate.

$$\eta_s = \eta_0 \cdot \delta^t, \quad \text{where} \quad t = \frac{s}{100000}. \tag{5.3}$$

In addition to that, the input text that is fed into the model has the following format by default:

- Whole paragraphs are used for training, validation and testing of the model.
- Words in the paragraph get tokenized and lemmatized.
- The tokens are made up of 1-grams.
- The tokens do not contain POS information.

Moving forward, we study the effects on the model's performance when changing one or more of the default settings. A summary of the model's parameter study is given in Table 5.1.

Figure 5.1 shows the different micro and macro accuracy values of the hashing model for different n-grams as a function of the training epochs. From the figure, one notices immediately that the macro accuracy is a more robust accuracy measure with fewer random fluctuations between epochs. Using 1-grams results in the highest micro accuracy and using 3-grams results in the lowest micro accuracy. When looking at the macro accuracy, the exact opposite is the case. There, 1-grams perform the worst, and 3-grams achieve the best results. Another interesting finding is that the micro accuracy for 3-grams decreases slightly after 4 epochs and saturates at around 86.5% accuracy. The other accuracy values show an upward trend with increasing epochs and asymptotically get closer to their maximum value.

Figure 5.2 shows the micro and macro accuracies of the hashing model when lemmatizing and not lemmatizing word tokens. The micro accuracy for the unlemmatized input is performing slightly better in the beginning, but the model scores a higher accuracy with the lemmatized input after 10 epochs. The macro accuracy is similar for both cases.

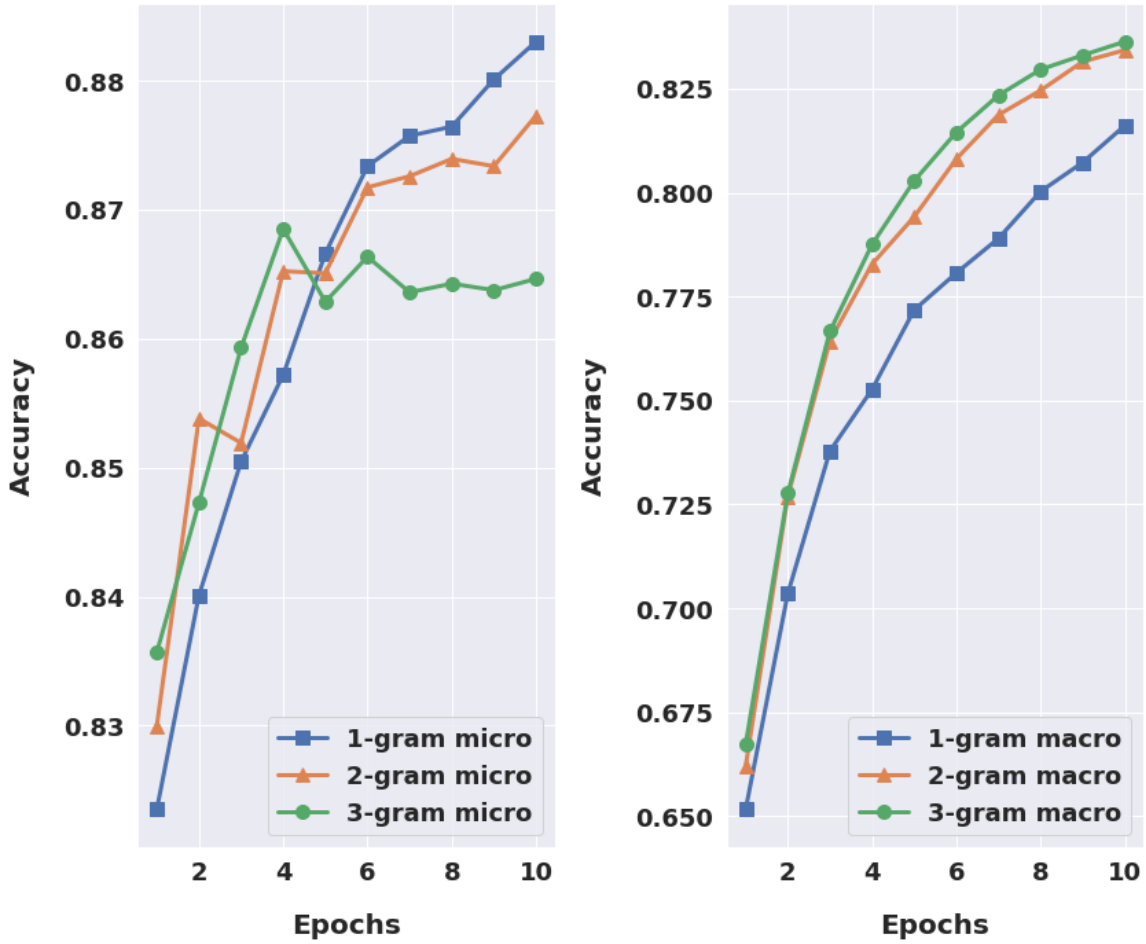


Figure 5.1.: Micro (left) and macro (right) accuracies of the hashing model for different n-gram size variations. The accuracies are plotted as a function of the number of epochs.

Figure 5.3 shows the different micro and macro accuracy values of the hashing model when using either sentences or paragraphs. In both cases and for almost all epochs, the model achieves similar accuracy scores.

Figure 5.4 shows the different micro and macro accuracy values of the hashing model when using either sentences or paragraphs and additionally provide POS information for each token. The POS information is appended at the end of each token. The accuracy values when using paragraphs or sentences are similar, both on a micro and macro level. The micro accuracies when using POS information are more stable than the ones without POS information (see Figure 5.3). There are less fluctuations between epochs. When comparing the macro accuracies for the cases with and without POS information, the models using paragraphs perform similarly. However, the sentence based model performs slightly worse when using POS information.

Figure 5.5 depicts the model's performance for different dropout rates. Both the micro and

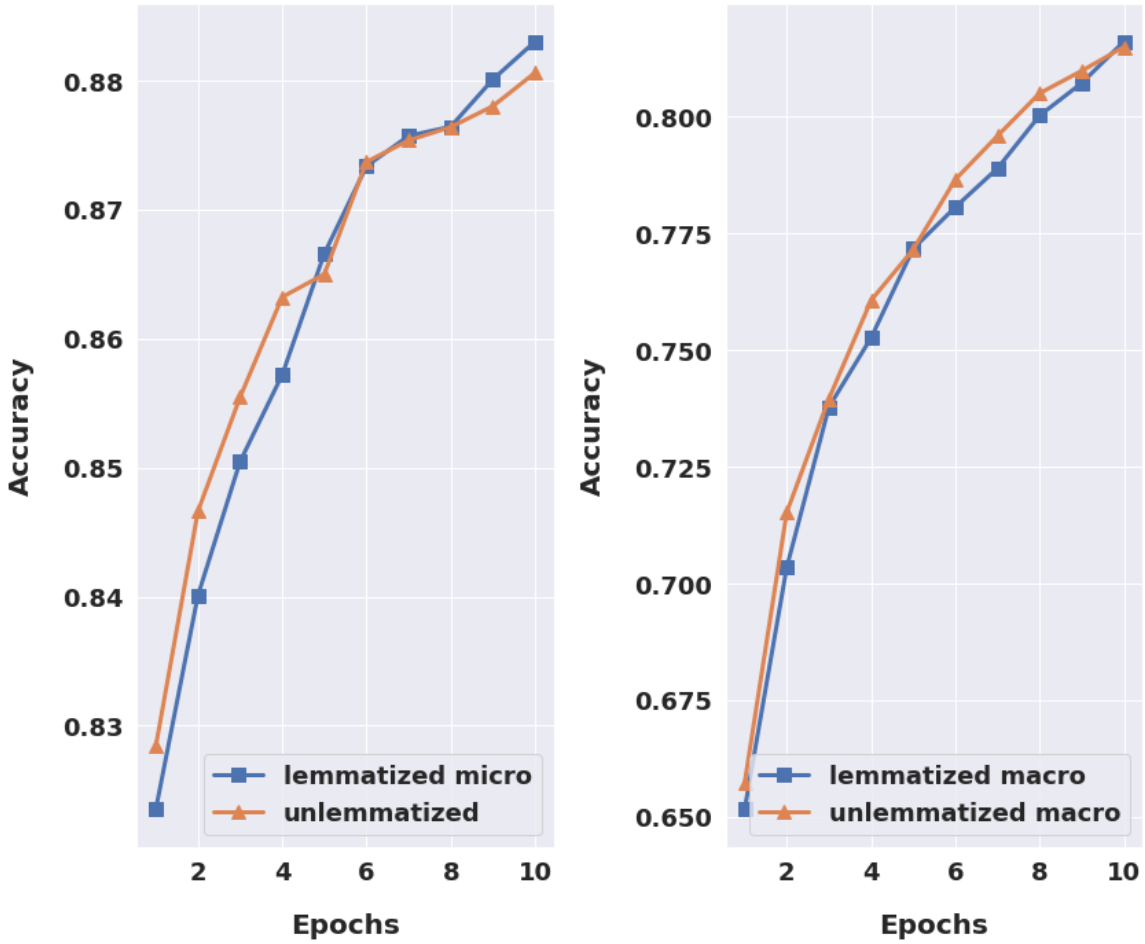


Figure 5.2.: Micro (left) and macro (right) accuracies of the hashing model when lemmatizing and not lemmatizing word tokens. The accuracies are plotted as a function of the number of epochs.

macro accuracies display similar behaviour. The higher the dropout rate, the worse the model performs. This might be because we only use 25 neurons in our hidden layer. The previous results show that the model does not suffer from overfitting and thus, dropout is not needed and counterproductive.

Figure 5.6 displays the accuracies for some other variations of the default settings. We test the model's performance when instead of the default reduction operation, we use the reduction operation from equation 4.1. We also test the model's performance using two layers with 25 neurons each instead of just a single hidden layer. We also study the effects of a hidden layer of double the size, namely 50 neurons, has on the performance. What is immediately noticeable is that the model using two hidden layers performs much worse than the others. This is probably the result of overfitting. The number of stored weights increase exponentially with the depth of the network. The model that uses a standard average pooling performs slightly worse than when the square root combiner is used.

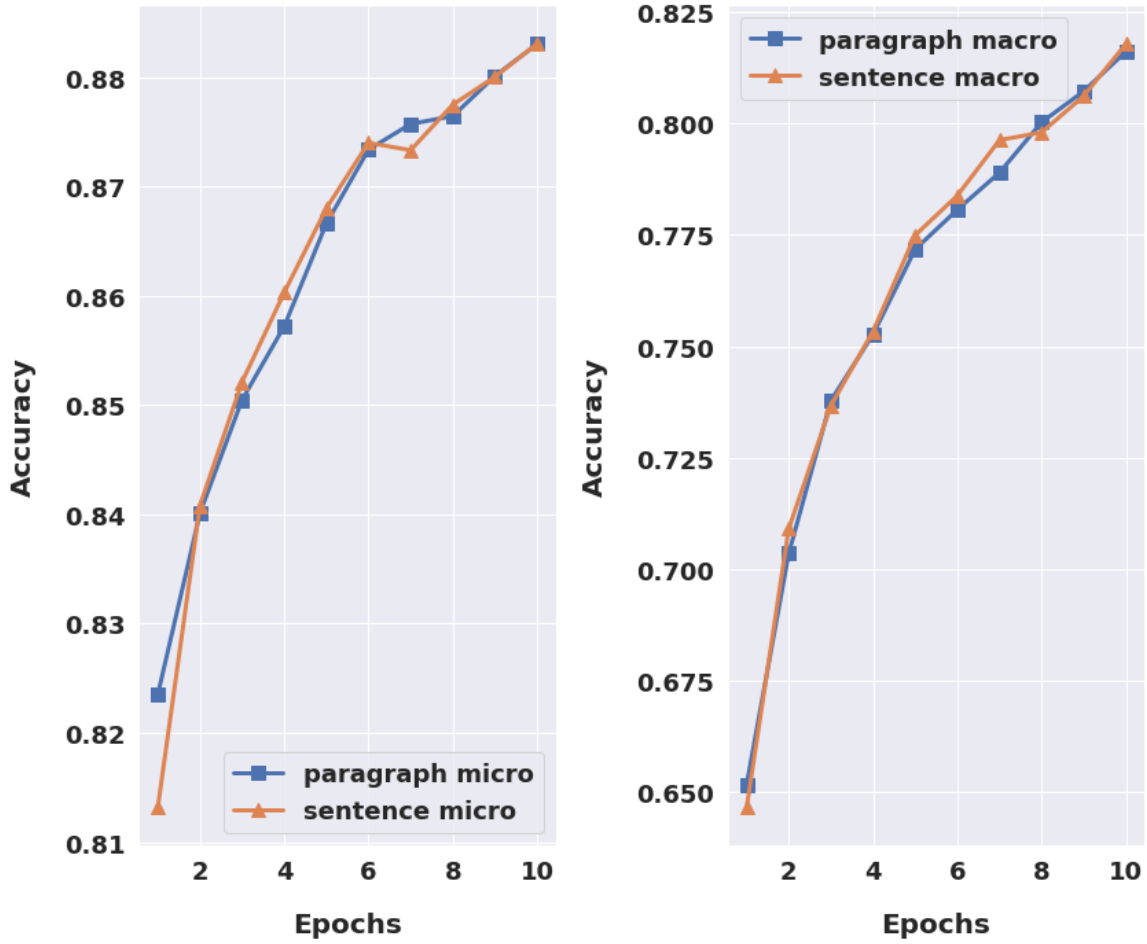


Figure 5.3.: Micro (left) and macro (right) accuracies of the hashing model when using either sentences or whole paragraphs. The accuracies are plotted as a function of the number of epochs. When using sentences instead of paragraphs, the train, validation, and test set only contain sentences and vice versa.

With almost 90% accuracy, the model with 50 hidden layer nodes achieves the best results from all the model variations in this section.

5.2. FastBERT Evaluation

In this section, we measure the accuracy of our fastBERT model. By default, we use the same settings as we did with the hashing model. However, fastBERT uses no combiner, tokens are not lemmatized, and sentences are used to create embeddings instead of paragraphs. The fastBERT model uses an Adam optimizer with the default parameters from Keras[1] Because BERT embeddings are resource-intensive to generate, we only measure the micro accuracy from equation 5.1. Not enough different senses are contained in the dataset

[1]The parameters can be looked up at `https://keras.io/api/optimizers/`. Visited on July 06, 2021.

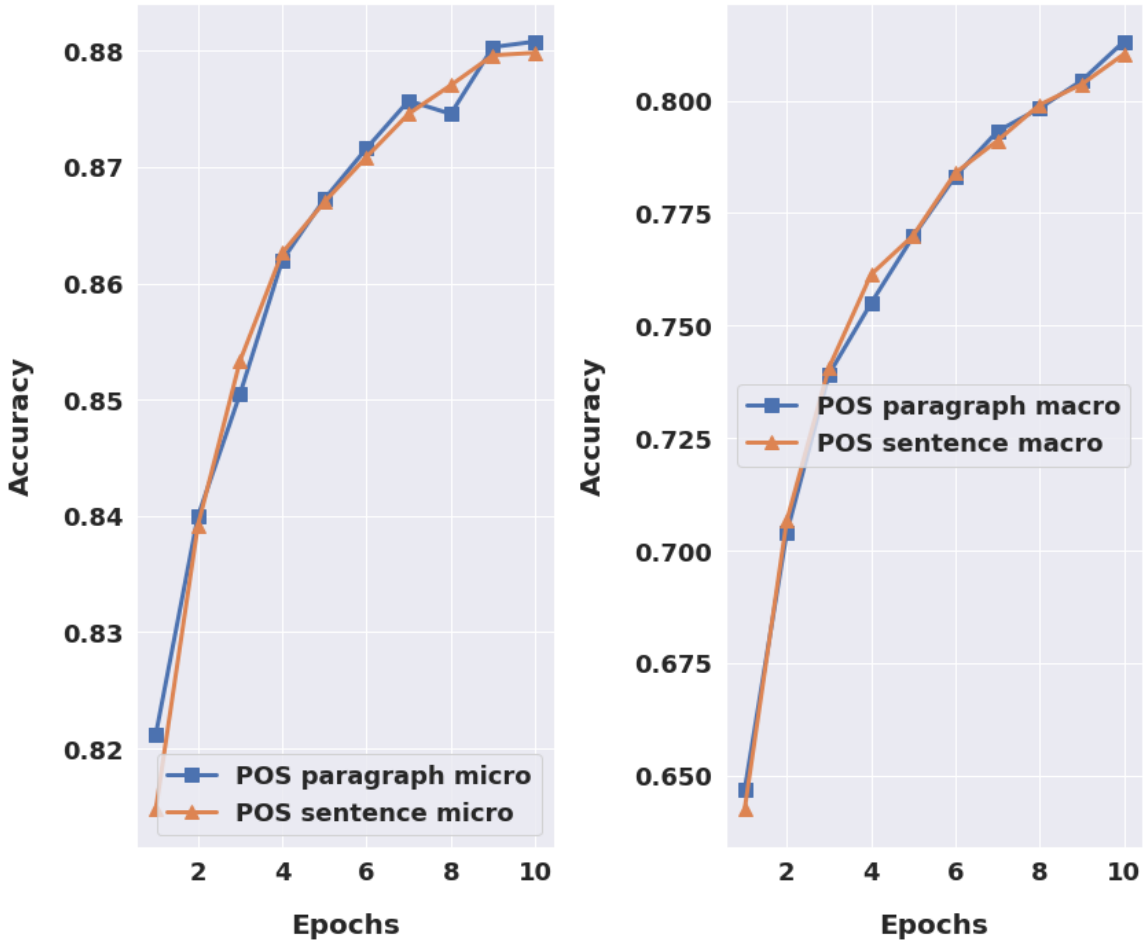


Figure 5.4.: Micro (left) and macro (right) accuracies of the hashing model when using either sentences or whole paragraphs together with POS information for each token. The accuracies are plotted as a function of the number of epochs. When using sentences instead of paragraphs, the train, validation, and test set only contain sentences and vice versa.

to get a good measure of the macro accuracy. We take the embedding of the *[CLS]* token as a representation of the sentence. This is in line with the research literature that studies the performance of BERT models for classification tasks, for example [Sun et al., 2019]. In the literature, the embedding of the *[CLS]* token is usually chosen to represent the whole input sequence. The accuracies that we measure with our generated test set is presented in Figure 5.7. The train/test ratio is given in Table 4.4. We notice that more nodes lead to a better classifier performance. However, after 3 or 4 epochs, the model accuracy begins to decrease. This result is consistent with reports from the literature. For example, [Zhu, 2019] reports a performance cap within 3 epochs. The authors of the paper [Devlin et al., 2019], where BERT was first presented, also recommend between 2 and 4 epochs. The fine-tuning of an already pre-trained model requires fewer epochs than when training a model from scratch.

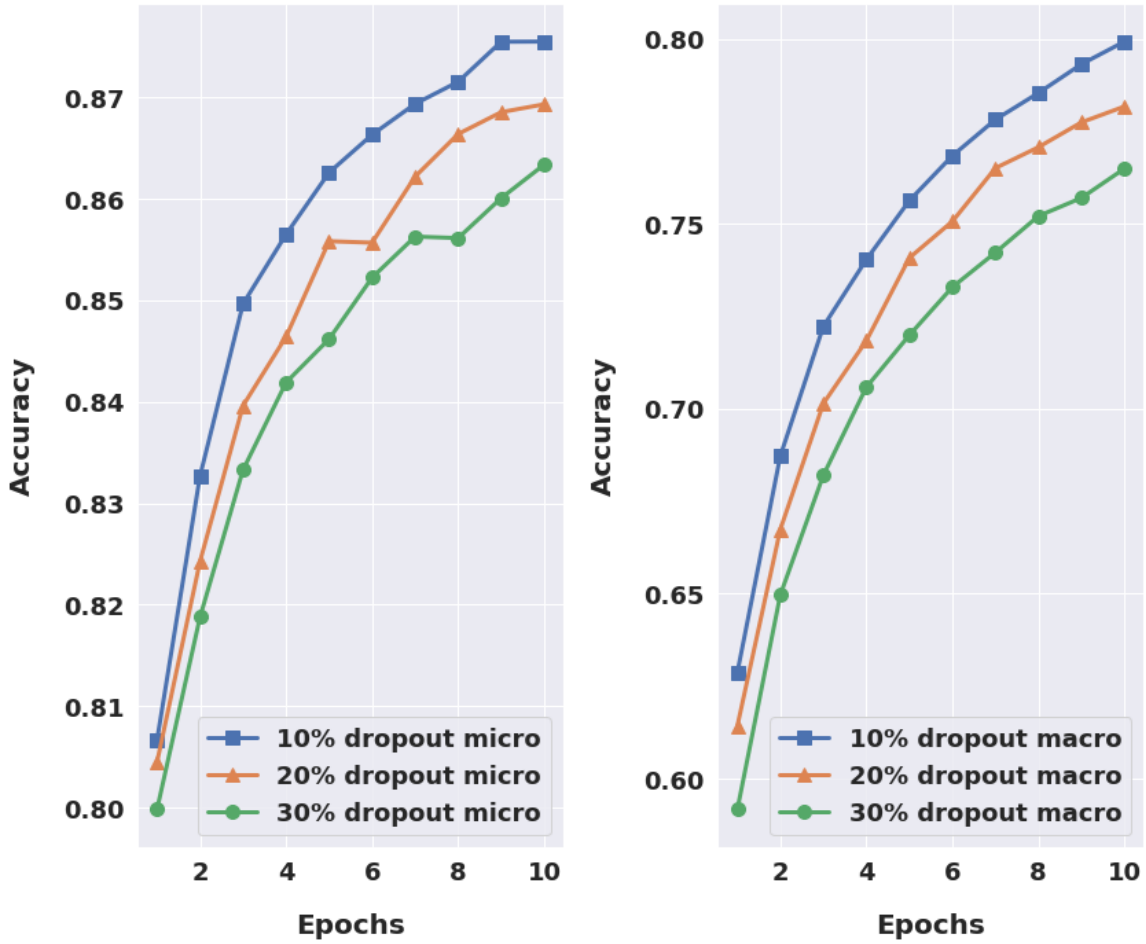


Figure 5.5.: Micro (left) and macro (right) accuracies of the hashing model when varying the dropout probability. The accuracies are plotted as a function of the number of epochs. Dropout is used only during the training phase. During the validation, and testing phase, all neurons are active, and dropout is disabled.

On the other hand, it is remarkable that when using a 20% dropout rate, the model's performance asymptotically approaches a maximum accuracy value instead of decreasing. For the fastBERT model, dropout has the opposite effect than for the hashing model in Section 5.1. The model that uses 25 nodes and a 20% dropout rate approaches a maximum accuracy of approximately 89%. That is the best performance out of the trained fastBERT models. Even though the fastBERT model was trained with less than 1% of the training samples the hashing model was trained with, it achieves similarly good accuracy scores. A summary of the parameter study of the fastBERT model on the generated test set is provided in Table 5.2.

We also evaluate the trained fastBERT model on the manually tagged legal dataset described in Section 3.3. For this purpose, we remove the examples that are marked as exceptional cases. We further remove all examples for senses not contained in the training

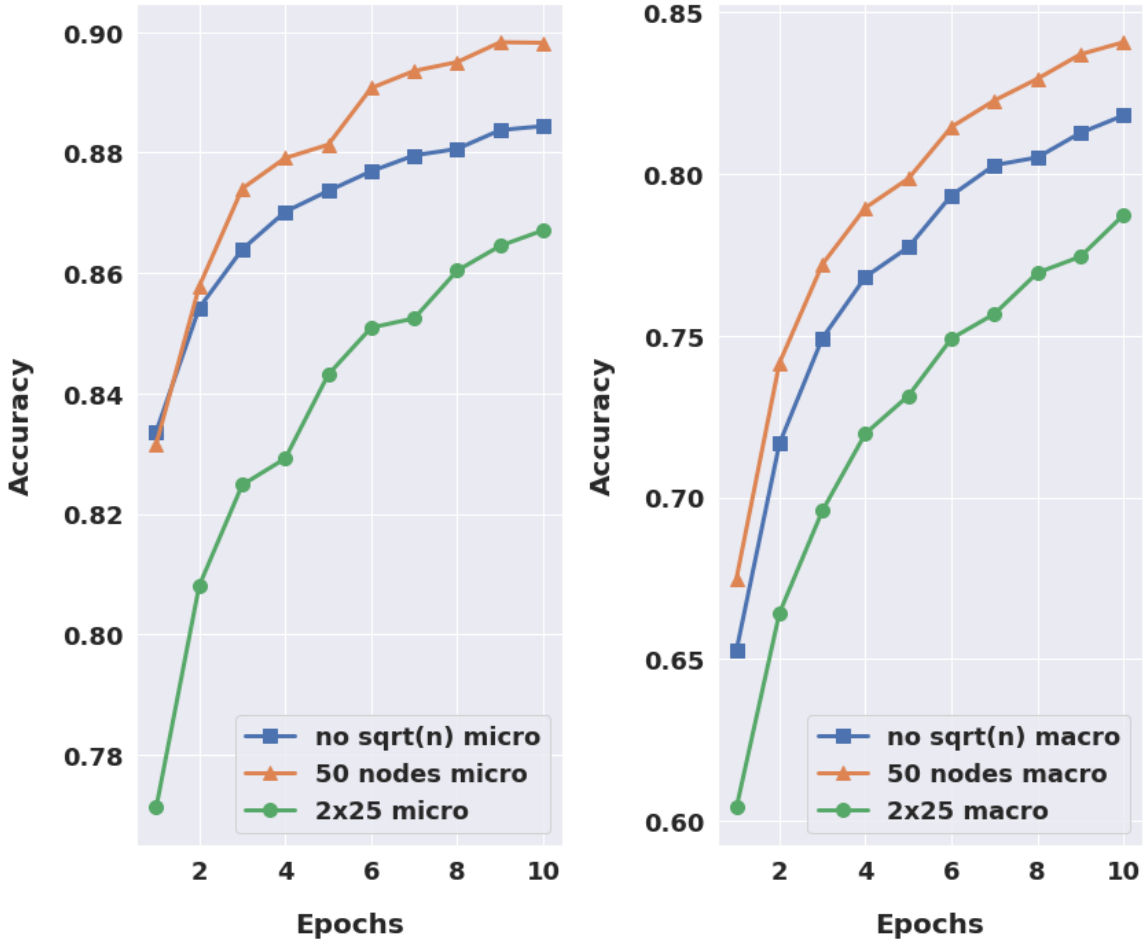


Figure 5.6.: Model's performance for different variations of the default settings. We test the model's performance when instead of the default reduction operation, we use the reduction operation from equation 4.1. We also test the model's performance when using two layers with 25 neurons each instead of just a single hidden layer. We also study the effects of a hidden layer of double the size, namely 50 neurons, has on the performance.

set when we trained the fastBERT model. This leaves us with **991** manually tagged examples to measure the performance of our model. A summary of the parameter study of the trained fastBERT model on the manually tagged legal dataset is given in Table 5.3.

Figure 5.8 shows the performance of fastBERT on the legal test set. In general, a downward trend with increasing epochs is visible. This means that the models start to overfit and lose their ability to generalize on different text types. This probably also has to do with the fact that legal texts are very different from the text contained in Wikipedia. Another reason for the worse performance on the test set is that senses are an abstract representation of a word's context and are subjective to a certain degree. This is evident from the low inter-annotator agreement score presented in Section 3.3. Oftentimes there is some degree of

Lemmatized	NG	POS	n	S	L	N	B	DO	η_0	δ	Micro	Macro
✓	1				1	25	32	0	1	0.98	88.30%	81.61%
✓	2				1	25	32	0	1	0.98	87.73%	83.43%
✓	3				1	25	32	0	1	0.98	86.85%	83.63%
	1				1	25	32	0	1	0.98	88.06%	81.50%
✓	1			✓	1	25	32	0	1	0.98	88.30%	81.79%
✓	1				1	25	32	10%	1	0.98	87.55%	79.93%
✓	1				1	25	32	20%	1	0.98	86.93%	78.17%
✓	1				1	25	32	30%	1	0.98	86.34%	76.49%
✓	1	✓		✓	1	25	32	0	1	0.98	87.98%	81.03%
✓	1	✓			1	25	32	0	1	0.98	88.08%	81.31%
✓	1		✓		1	25	32	0	1	0.98	88.44%	81.81%
✓	1				1	50	32	0	1	0.98	**89.83%**	**84.08%**
✓	1				2	25x25	32	0	1	0.98	86.71%	78.73%

Table 5.1.: Parameter study of the hashing model. Depicted are the accuracies of all trained models. The following abbreviations in the table header are explained in the following: **NG** = n-gram size, **POS** = part of speech information was used, **S** = if sentences are used (instead of paragraphs), **L** = number of hidden layers, **N** = number of nodes in the hidden layers, **B** = batch size, and **DO** = dropout rate. η_0 and δ are parameters of the learning rate given in equation 5.3. n indicates whether the pooling function from Equation 4.2 was used instead of the one from Equation 4.1. "Micro" is the maximum micro accuracy achieved by the model within 10 epochs. "Macro" is the maximum macro accuracy achieved by the model within 10 epochs. The model using a hidden layer with 50 nodes achieves both the best micro and macro accuracy out of all the trained models.

overlap between the contexts that are captured in the sense labels.

S	L	N	B	DO	Micro
✓	1	12	32	0	84.23%
✓	1	25	32	0	84.78%
✓	1	50	32	0	85.23%
✓	1	100	32	0	85.63%
✓	2	50x25	32	0	85.11%
✓	1	12	32	20%	87.02%
✓	1	25	32	20%	**88.59%**

Table 5.2.: Parameter study of the fastBERT model on the generated test set. The following abbreviations in the table header are explained in the following: **L** = number of hidden layers, **N** = number of nodes in the hidden layers, **B** = batch size, and **DO** = dropout rate. "Micro" is the maximum micro accuracy achieved by the model within 10 epochs. Due to the availability of less training data, we only measure the micro accuracy, as not enough examples for each sense are contained in the test set to calculate the macro accuracy. The model using a hidden layer with 25 nodes and 20% dropout achieves the best micro accuracy out of all the trained models.

S	L	N	B	DO	Micro
✓	1	12	32	0	68.62%
✓	1	25	32	0	72.75%
✓	1	50	32	0	68.72%
✓	1	100	32	0	**74.47%**
✓	2	50x25	32	0	73.16%
✓	1	12	32	20%	72.86%
✓	1	25	32	20%	73.76%

Table 5.3.: Parameter study of the fastBERT model on the manually tagged legal test set. The following abbreviations in the table header are explained in the following: **S** = sentences instead of paragraphs were used to create embeddings, **L** = number of hidden layers, **N** = number of nodes in the hidden layers, **B** = batch size, and **DO** = dropout rate. "Micro" is the maximum micro accuracy achieved by the model within 10 epochs. Due to the availability of less training data, we only measure the micro accuracy, as not enough examples for each sense are contained to calculate the macro accuracy. The model using a hidden layer with 100 nodes achieves the best micro accuracy out of all the trained models.

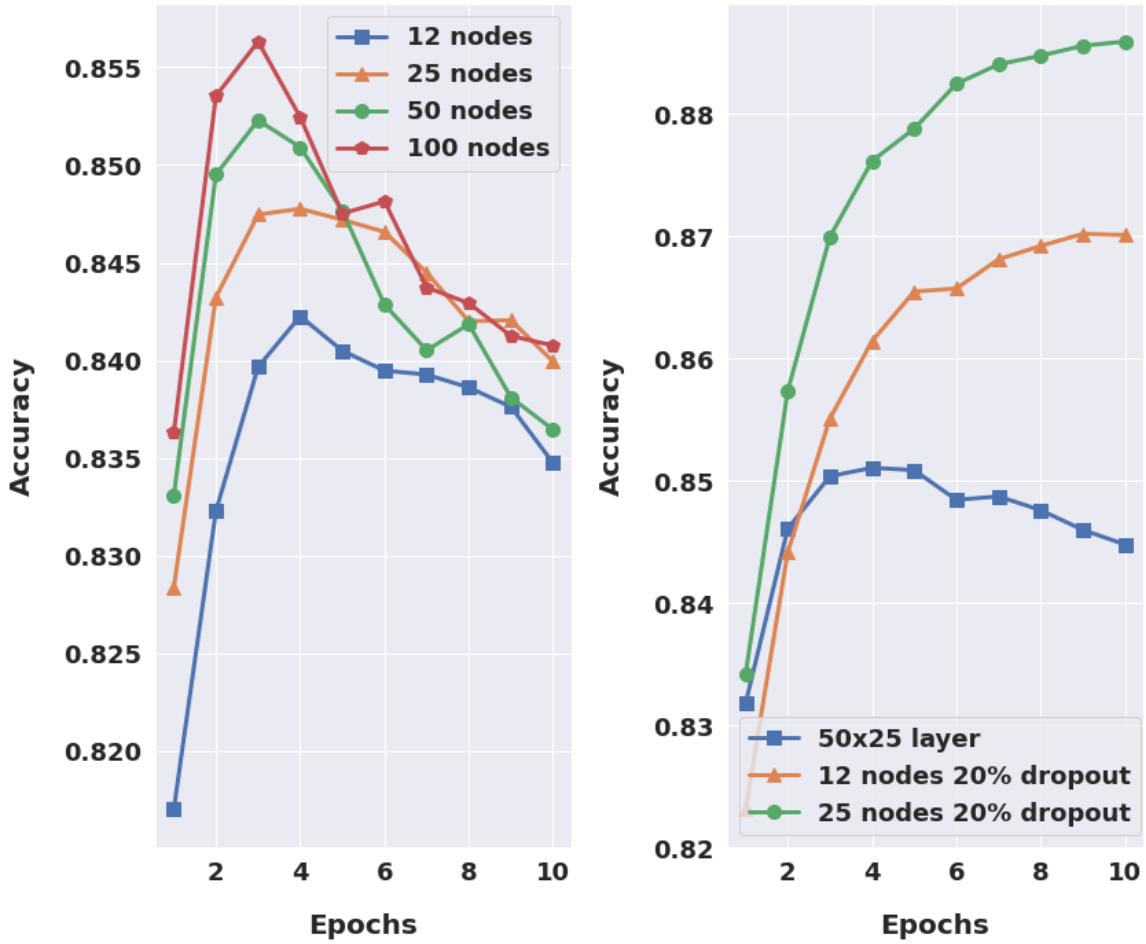


Figure 5.7.: Performance evaluation of the fastBERT model on the generated test set from Chapter 4. When using the fastBERT model, sentences are used instead of paragraphs. Due to the availability of less training data, we only measure the micro accuracy, as not enough examples for each sense are contained in the test set to calculate the macro accuracy. We evaluate the model for different hidden layer sizes, a different number of hidden layers and different dropout rates.

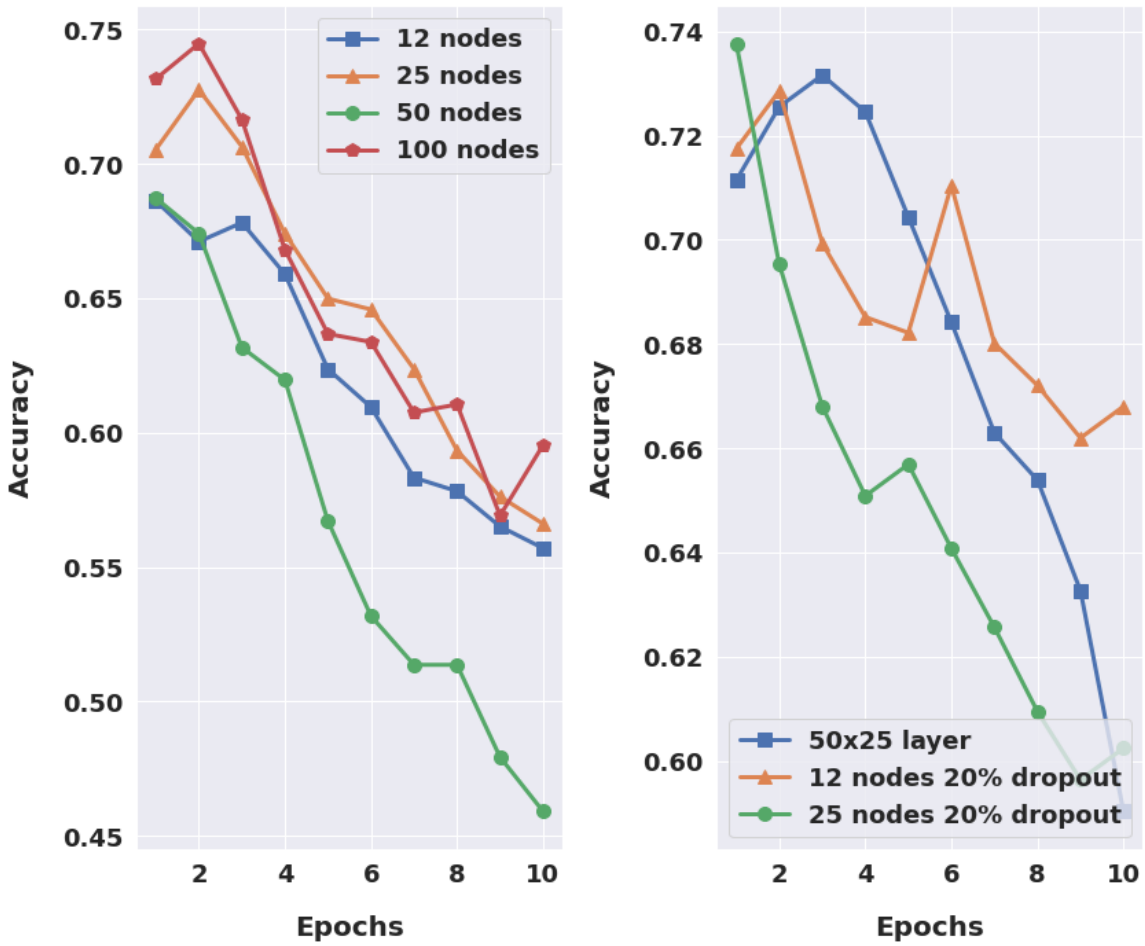


Figure 5.8.: Performance evaluation of the fastBERT model on the manually tagged legal test set. The performance is tested on 991 manually tagged examples from the legal test set described in 3.3. A downward trend with increasing epochs is visible, which is a sign of overfitting.

6. Qualitative Evaluation

To assess the quality of the different senses for ambiguous words that appear frequently throughout the legal corpus, **11** interviews were conducted with lawyers from different age groups and different areas of specialization. Alternatively, a legal research task could have been simulated with lawyers by setting up a simple search engine connected to the sense database and letting them research for a particular legal case or topic while using the sense filter. However, doing so would have provided hardly any additional benefit and thus did not justify the time needed to set up such a system. Furthermore, the choice of legal databases that lawyers use depends on their areas of expertise. Because we only have data from the legal publisher Dr. Otto Schmidt, it would have been challenging to find suitable candidates for the evaluation because this publisher focuses mainly on content for German company law and tax law. Even though the legal experts interviewed for this thesis specialized in various legal areas, none of them uses the database from the publisher Dr. Otto Schmidt. Finding a suitable task for them to research on within this publisher's data would most likely not have been possible, or in the very least, would not have been a representative example of a genuine legal research task.

The interview guideline used for the qualitative evaluation of the results can be found in Appendix D. The interview consisted of three parts. In the first part, personal information like age and IT affinity, and their level of experience as lawyers are determined. In the second part, their general search behaviour was analysed. The third part was about assessing the implemented word sense disambiguator and how they would like to interact with a sense filter.

6.1. General Search Behaviour

Figure 6.1 shows the age distribution of the interviewed legal experts. Most of the interviewed legal experts were in their thirties. A relatively large proportion of the interviewed people do not use filters regularly. The legal practitioners who rarely use filters either use precise legal keywords or narrow down search results by combining multiple legal keywords. Figure 6.2 depicts the filter usage behaviour of the interviewed people. When asked about how the interviewees would assess their own IT skills, the majority claimed to have very good IT skills compared to the average legal practitioner. Figure 6.3 shows how the lawyers rated their IT skills. Interestingly, only 2 of the interviewed people said that they only have a fundamental understanding of technology.

When asked about the average time per week they spend on conducting legal research, all lawyers mentioned that this highly fluctuates. There are weeks where they do nothing but conduct legal research, but there are also weeks where they do not engage in legal

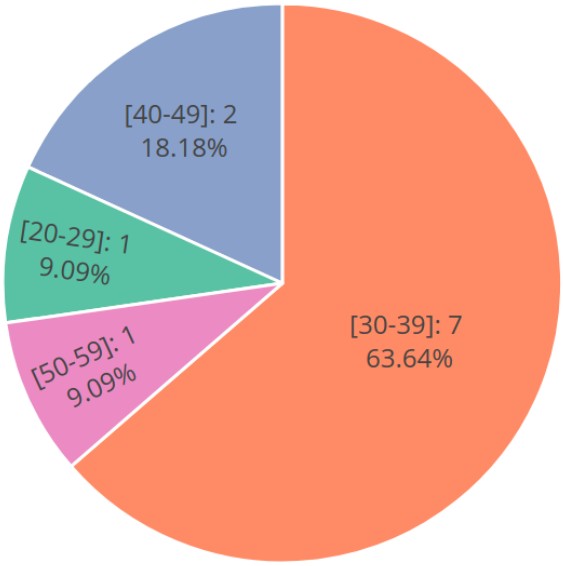


Figure 6.1.: Age distribution of the people interviewed for the qualitative evaluation.

research at all. On average, the interviewees estimated that they spend **9.27** hours per week conducting legal research, with a standard deviation of **8.89** hours per week. While some of the interviewed people mentioned that they only spend 1 hour on legal research each week, others estimated that they spend 20 or 30 hours per week. This results in a large standard deviation around the average value. The people who spend 20 or 30 hours per week on legal research were either working on their PhD or were working as legal counsellors.

6.2. Sense Filter Evaluation

To qualitatively evaluate the sense filter, the interviewed legal experts were asked to judge the quality of the distinguished senses of some ambiguous words. For this purpose, four of the most frequent ambiguous German words from the legal corpus provided by the publisher Dr. Otto Schmidt were taken. These words were *"Verfahren"*, *"Berufung"*, *"Revision"*, and *"Anlage"*. The possible senses can be looked up from the provided table in the interview guideline in Appendix D. Before judging the quality of the provided senses, the interviewees were asked which meaning first comes to mind for each of these four words. The distribution of the associated meanings can be looked up in Figure 6.4. Up to this point, the legal experts have not yet seen the possible senses in our sense database, so the answers that they provided tended to be more specific. The answers that the lawyers gave were mapped to one of the possible meanings from our sense database. For example, if the first associated meaning that came to mind for the word "Anlage" was the word "Parkanlage" (en. park), then that would count towards "Bauliche_Anlage" (en. structural installation), since "Bauliche_Anlage" is a more general term. All the interviewed people associated the word "Berufung" with "Berufung_(Recht)" (a legal remedy to appeal a court judgement).

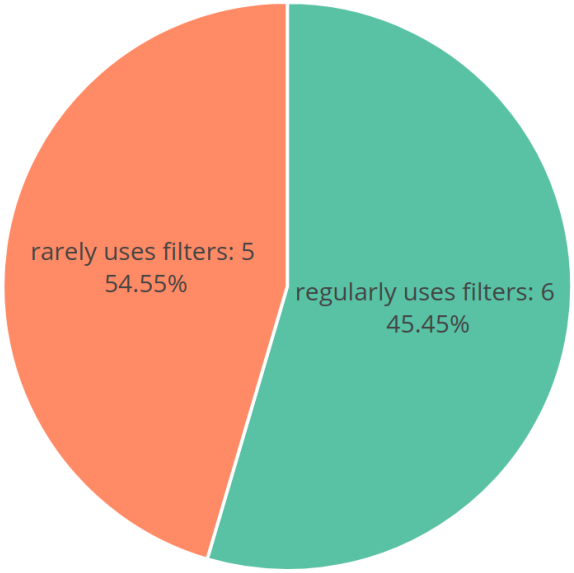


Figure 6.2.: Filter usage behaviour among the interviewed legal experts.

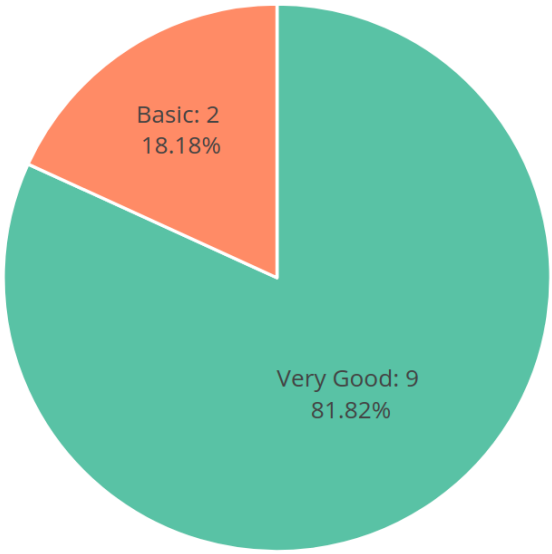
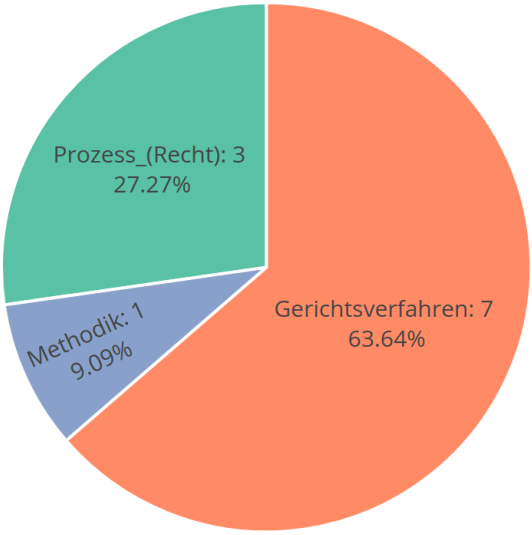
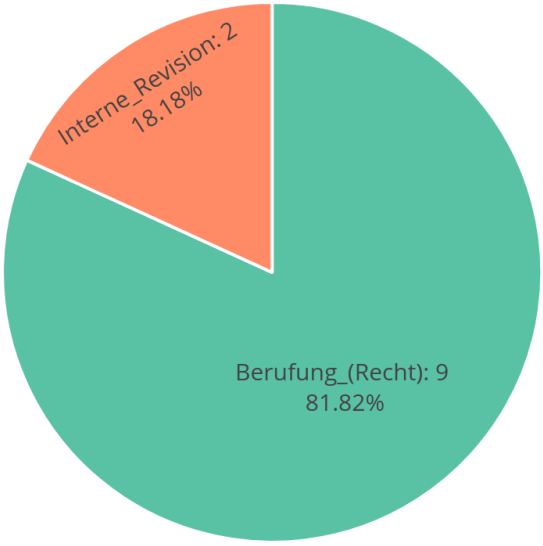


Figure 6.3.: IT skills of the interviewed legal experts when compared to the average lawyer.

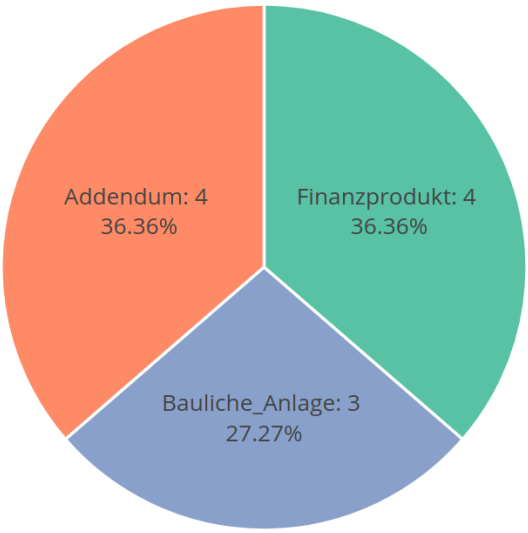


(a) Verfahren

(b) Berufung



(c) Revision



(d) Anlage

Figure 6.4.: Distribution of the associated meanings for four of the most frequent words in the legal corpus provided by the legal publisher Dr. Otto Schmidt. The lawyers were asked which meaning first comes to mind when they hear one of these words. The answers that the lawyers gave were mapped to one of the possible meanings. For example, if the first meaning that came to mind for the word "Anlage" was the word "Parkanlage", then that would count towards "Bauliche_Anlage", since "Bauliche_Anlage" is a more general term. A list of the possible meanings is given in the interview guideline for the qualitative evaluation in Appendix D.

The different meanings of the words were communicated to the lawyers by taking the last part of the corresponding Wikipedia article URL for that specific word sense. For example, "Berufung" in the sense of "Rechtsmittel" is described by the Wikipedia article https://de.wikipedia.org/wiki/Berufung_(Recht). We would then take the last part of the URL, namely "Berufung_(Recht)", to communicate this specific sense with the legal experts. Since we have saved the Wikipedia URL for each word sense that we distinguish in our sense database, this is the most straightforward way to communicate specific senses with the would-be users of our sense-filter.

In the following, we wanted to find out if this approach leads to sense labels that contain enough semantic information for the lawyers to know which sense is meant with each title. When the legal experts were asked to judge the different senses in our sense database that we can distinguish for those words, the feedback was mixed. A common issue that almost all lawyers had was that some of the listed meanings are not completely distinguishable from each other, and the semantics of some of the possible senses overlap with each other. For example, the difference between "Verfahrenstechnik" (en. process engineering) and "Fertigungsverfahren" (en. manufacturing process) was not always clear to the lawyers. In some cases, the sense labels do not contain enough semantic information to sufficiently describe the meaning. For example, the difference between "Gerichtsverfahren"(en. court proceeding) and "Prozess_Recht" (en. litigation) was not clear enough. According to the German Wikipedia, "Gerichtsverfahren" describes the judicial review of facts, and "Prozess_Recht" refers to a court litigation initiated by a lawsuit. Because word meanings are subjective, the senses that Wikipedia distinguishes are not necessarily intuitive for everyone. Also, the difference between "Addendum" and "Addendum_Schriftwesen" was not clear for the lawyers. In this case, there is no difference between those two senses. In fact, they both refer to the exact same meaning, but due to internal redirections within Wikipedia, the URLs `https://de.wikipedia.org/wiki/Addendum` and `https://de.wikipedia.org/wiki/Addendum_(Schriftwesen)` both link to the same article. Such errors likely lead to confusion for the users of our sense filter. Another thing that was frequently remarked is that the senses are not within the same level of hierarchy. For example, "Verwaltungsverfahren" (en. administrative proceeding) is a sub-term of "Gerichtsverfahren".

When asked about suggestions for improvement, common remarks were that the fineness of the senses was too coarse-grained since the lawyers would probably not use many of the senses that do not related to anything law related. It was mentioned that it would be helpful to have a dynamic slider to manually adjust the fineness of the possible senses. As of now, we are only able to distinguish very general terms within our sense inventory. Especially lawyers who use precise keywords instead of filters remarked that a sense filter would probably not provide many benefits if they are already familiar with a legal area. Examples of very precise legal terms that the interviewed lawyers used recently were "Treuhandgläubiger", "Produktbeobachtungspflicht", and "Inkassodienstleistung". However, for beginners or lawyers who are unfamiliar with a particular legal field and thus do not know the right keywords, a sense filter would be helpful.

When asked about the accuracy requirements of the sense filter the majority said that, instead of a minimum precision, it is much more important that the benefits and the added

value for them must be noticeable. If given a choice between filtering for relevant results or excluding specific topics, results were mixed, as both methods would have their own advantages and disadvantages. Preferences were also mixed when they were given the choice between receiving many results that also contain irrelevant results, or very few results containing highly relevant results but where some relevant results might be missing. Lastly, when asked about whether they would prefer to interact with the sense filter already during query input in the form of a dropdown menu where they can select each word's sense, or filter results after a first search round, the majority preferred the first option.

7. Conclusion and Outlook

7.1. Summary

The majority of lawyers are unsatisfied with the relevance of retrieved information from legal databases. The overall goal of this thesis was to aid legal practitioners in legal research tasks by providing them with a word sense filter to speed up research. The idea was to increase the relevance of retrieved information by allowing them to filter for the intended meaning of an ambiguous word that they enter into the search bar of a legal database. For this purpose two neural net-based word sense classification models were presented to automatically sense-tag legal corpora. An effective way to deal with data sparseness for German sense annotated data has been implemented. Extensive quantitative and qualitative evaluations of the models were conducted. The research results are summarized based on the defined research questions posed in Section 1.3.

What algorithms already exist to automatically classify word senses (knowledge-based, unsupervised, and supervised)?

Chapter 2 described the theoretical fundamentals needed to understand the intuition and working mechanisms behind different word sense disambiguation algorithms. After introducing word embedding algorithms and explaining how their vector representation encodes context information, standard baselines and evaluation measures for WSD tasks were presented. The "Most Frequent Sense" heuristic was introduced as a strong baseline, as it is hard to beat by WSD systems due to the skewed sense frequency. The baseline can be viewed as a knowledge-based sense classifier because it only needs the most frequent sense information for each ambiguous word. Other knowledge-based WSD algorithms that were presented were the Lesk algorithm and its different variations. Besides the knowledge-based algorithms, we gave examples of unsupervised WSD algorithms that only need unannotated raw text corpora. Two unsupervised approaches were presented: a clustering-based approach and a graph-based approach.

In contrast to unsupervised methods, supervised methods need large amounts of sense-tagged data. Seven different supervised algorithms that can be used for WSD were presented: the decision list, decision trees, naïve Bayes, memory-based machine learning methods, support vector machines, AdaBoost learning methods, and neural networks. In this thesis we used a neural network-based approach.

What possibilities are there to compensate for the lack of German sense annotated (legal) data?

Resources for the training of supervised German WSD models were listed at the end of Chapter 2, together with their advantages and disadvantages. Available German datasets for WSD are manually sense labelled and do not contain enough training samples for each sense. Their sense inventory is limited and comprises a small vocabulary. Because no dataset of suitable size is publicly available, we automatically generated a sense annotated training corpus and a sense inventory that contains a great variety of ambiguous words. We did so by using the disambiguation pages of Wikipedia. The implementation of the algorithm is described in Chapter 4. Our sense database contains sense groups for 148,615 ambiguous words. In total, 342,861 different senses are contained within those sense groups, which amounts to an average of 6.02 possible senses per sense group. In total, the whole dataset comprises paragraphs with 64,274,208 sense-tagged ambiguous words. To compensate for the lack of German sense annotated legal data, we manually sense tagged legal text from the legal publisher Dr. Otto Schmidt with senses from our generated sense inventory.

How do WSD algorithms perform on German (legal) text?

Chapter 4 introduced two different models for sense classification, the "hashing model" and the "fastBERT" model. The results of the qualitative performance evaluation and parameter studies are given in Chapter 5. Because of the working mechanisms of our data generation algorithm and sense classifier, we are only able to measure the accuracy of the model. We measure the "micro" and "macro" accuracy of the hashing model on regular German text taken from Wikipedia articles. The model consistently achieves accuracy scores between 75% and 90% for the accuracy measures, depending on the settings and model hyperparameters. The accuracy values of the fastBERT model on regular German text consistently lie between 81% and 90%, depending on the chosen hyperparameters. The performance decreases after three to four epochs. When using dropout, the performance significantly increases, as it counters the performance decrease with increasing epochs. When assessing the model's performance on a manually sense tagged legal corpus, the performance drastically decreases with increasing epochs. The initial accuracy values after one epoch consistently lie between 68% and 75%.

How do legal experts judge the usefulness of our word sense filter?

Two interview rounds were conducted for this thesis, one at the beginning of the thesis and one at the end. The purpose of the first interview round was to assess how useful a sense filter would be for legal experts and which requirements a sense filter needs to fulfil. The results of these interviews are analysed in Section 3.1. The interviewed lawyers were not satisfied with the retrieved results of German legal databases that they currently use. While a sense filter for legal terms would not benefit them in their research tasks, a sense filter for natural language terms would be a beneficial addition to already existing filters. In the second interview round, the quality of the distinguishable word senses

was assessed. The results are presented in Chapter 6. Feedback and suggestions for improvement were collected, which give room for future work refinements. In summary, the fineness and granularity of the distinguished word senses need to be adjustable to the legal expertise of the user.

7.2. Critical Reflection and Limitations

Quantitative Evaluation

Because of the skewed sense distribution due to Zipf's law and the fact that senses with fewer training examples get misclassified more often by our model than senses with many training examples, the standard accuracy score would be artificially increased. To compensate for this, we introduced a "macro" accuracy score that measures how well the model performs on average for each individual sense. With this, we get more stable and representative performance measures from our model.

However, since we do not use the listed senses from the "See also" section from Wikipedia's disambiguation pages and only add senses with a certain minimum amount of training examples into our sense inventory, some sense groups in our sense database contain only one example. This side effect slightly increases the measured accuracy values. On the other hand, some sense groups in our sense database include one or more entries from Wikipedia articles of people with the same last name as the ambiguous word. This effect leads to a decrease in the model's accuracy since some sense groups contain more sense entries to chose from which leads to more possibilities for misclassification.

The performance of the "fastBERT" model on the manually sense tagged legal corpus achieved worse results than for the automatically generated sense tagged corpus from Wikipedia. A reason for this is the subjectiveness of word senses. In section 3.3 we evaluated the inter-rater reliability of the sense annotations for our legal corpus to be 69.6%. The semantics from some word senses overlap and are not clearly distinguishable. It is not always clear which sense to classify a word with.

Qualitative Evaluation

We assessed the quality of the distinguishable senses of our model independently from a specific legal corpus. However, to be more precise in our evaluation, we have to better understand the search behaviour of legal practitioners. For example, some people usually search for "Fundstellen" (en. references) instead of using keywords. Subsequently, they would rate a sense filter as less helpful because it does not align with their search pattern. Legal experts who are already familiar with a legal field tend to use highly specialized keywords that either do not need sense disambiguation or where the disambiguated word senses need to be very fine-grained. They would then also judge a sense filter with senses taken from Wikipedia as less useful, as the senses listed in Wikipedia's disambiguation pages are too coarse-grained. The possibilities to generate a sense inventory with very fine-grained senses for specialized keywords are limited by the availability of appropriate encyclopedic resources.

Another thing that might lead to less accurate qualitative evaluation results is that people tend to remember things differently or leave out details that they judge as insignificant when asked about how they do certain tasks. What would complement the interview results would be to observe them doing a legal research task in real-time.

7.3. Possible Improvements and Future Work Outlook

To improve the performance of the sense classifiers, senses that come from people's last names should be removed from the sense database. The manual sense label distribution in Figure 3.12 indicates that a large proportion of possible senses for the ambiguous words is not contained in the sense database. A solution for this could be to extend the sense database with other sources than Wikipedia, for example with the German Wiktionary or with GermaNet. Since the interviewed legal experts wished to filter for more fine-grained senses, it makes sense to look for other German encyclopedic resources that distinguish between more senses. Also, senses that are semantically more distinguishable should be taken since senses are subjective to a certain degree. The model's performance suffers from the low inter-rater reliability score for the manually tagged legal dataset. The meaning of senses can overlap, and it is not always clear which sense to choose.

In addition to that, more extensive experiments regarding the model's settings and hyper-parameters can be conducted to find the best-performing combination of parameters. It can also be researched how different embedding choices influence the model. When using the "fastBERT" model, we used the embedding of the *[CLS]* token to represent our text sequence. Other possibilities would be to just take the contextual word embedding of the token that should be sense-classified. Alternatively, the output of the hidden states can be averaged or pooled together to capture the semantic content of the input sequence.

Appendix

A. Technical Documentation

In the following a technical information and documentation regarding the implementation of the algorithms for this thesis are given. The implementation is heavily based on the implementation described in [Schulz, 2021] and comprises two parts: an algorithm for generating sense annotated data, and another part that implements and trains the sense classifier.

A.1. Data Creation

The implementation for the creation of WSD training data can be downloaded from `https://github.com/AndreasPadProbst/fastSense_CreateData`. This package creates a WSD Trainings Corpus from the disambiguation pages of the German Wikipedia. The main structure of this code is taken from `https://github.com/texttechnologylab/fastSense`(accessed: 20.02.2021) and modified to be able to parse the German version of the Wikipedia. Also, the original code uses the CoreNLP Java library from the Stanford NLP Group. As not all desired modules are available for the German language, like a lemmatizer, the spaCy library is used here instead.

Setup

The implementation was developed with **Python 3.6** and the following additional packages were used:

- mwparserfromhell = 0.6
- spacy = 3.0 (along with the "de_core_news_lg" pipeline version 3.0)
- tensorflow = 2.5

Required Data

Following dumps need to be downloaded from https://dumps.wikimedia.org/dewiki/ (accessed: 21.02.2021):

- dewiki-*-categorylinks.sql.gz
- dewiki-*-page.sql.gz
- dewiki-*-pages-articles.xml.bz2

How to Use

Step 1:

To extract the relevant senses, corresponding page links, sense groups, and tokens, the **cli_wiki_extract.py** script has to be executed with the following command line arguments:

--dump	Path to the bz2-compressed XML dump of the German Wikipedia (dewiki-*-pages-articles.xml.bz2).
--page_table	Path to gzip-compressed SQL dump of page table (dewiki-*-page.sql.gz).
--categorylinks_table	Path to gzip-compressed SQL dump of categorylinks table (dewiki-*-categorylinks.sql.gz).
--intermediate	Path to folder where intermediate information about tokens and links will be stored.
--db	Path to database where information about the senses and sense groups is stored.

A shell command example for the code is given below:

```
# python cli_wiki_extract.py --page_table ./data/dewiki-latest-page.sql.gz --cat
egorylinks_table ./data/dewiki-latest-categorylinks.sql.gz --db ./data/sql3_db/f
astSense.db --intermediate_output ./data/intermediate --dump ./data/dewiki-lates
t-pages-articles.xml.bz2
```

Step 2:

To export the extracted data into a format more suitable for training, the **cli_wiki_export.py** script has to be executed with the following command line arguments:

--intermediate	Path to the folder where intermediate files from previous step are stored.
--db	Path to database from the previous step.
--output	Path to output folder for training data.
-f	Comma-separated options for output in the order [Name], [N-Gram Size], [Caseless], [Ignore Punctuation], [Add PoS Tags], [Uses Lemma], [Uses Sentences] (e.g. -f p_out,1,0,0,1,1,0). With these options it can be specified whether the tokens should be lemmatized, if n-grams should be formed, if a PoS tag should be added to the tokens, if case and punctuations should be ignored and if information from the whole paragraph or just the

sentence should be used. When a certain n-gram size is chosen, all possible n-grams for that size and smaller sizes will be generated.

A shell command example for the code is given below:

```
# python cli_wiki_export.py --intermediate ./data/intermediate --db ./data/sql3_
db/fastSense.db --output ./data/export_output -f p_out,1,0,0,1,1,0
```

Remarks

- Depending on the used hardware, both scripts can take over 10 hours each to run.
- The script creates over 64 million training samples in TensorFlow Record format.

A.2. Model Training

The implementation for the creation of WSD training data can be downloaded from `https://github.com/AndreasPadProbst/fastSense_TrainModel`. This package implements and trains a linear sense classifier on the data generated with the algorithm from Section A.1.

Setup

The implementation of this part was developed with **Python 3.9** and the following additional packages were used:

- tensorflow = 2.5

How to Use

To train the sense classifier, the **cli_train.py** script has to be executed with the following command line arguments:

--data	Path to the folder containing the training, validation and test data generated with the model from Section A.1.
--models_dir	Path to the folder where the trained model should be saved. If the folder already contains an existing model, the script will continue training the model.
--final_models_dir	Path to the folder where the trained models will be saved after each epoch. Each model is placed in a timestamped subfolder, together with information on the used parameters to train that model.

--jobs	Path to a JSON file containing the parameters that should be used to train the model.

A shell command example for the code is given below:

```
# python cli_train.py --data ./training_data --models_dir ./saved_model --final_
models_dir ./final_model_output --data ./train_jobs.json
```

An example for a *train_jobs.json* file that contains the default parameters of the model specifications is given below.

```
{
  "dataset_name": "p_out",
  "model_name": "my_model",
  "params": {

    "hash_bucket_size": 10000000,
    "embedding_size": 25,
    "use_sqrtn_combiner": true,
    "clip_gradients": false,
    "learning_rate": 1.0,
    "decay_rate": 0.98,
    "decay_steps": 100000,
    "hidden_layer_sizes": [25],
    "dropout_keep_prob": 1.0
  },
  "target": {
  "metric_key": "loss",
  "flip_sign_of_metric": true,
  "end_if_slope_less_than": 0.0,
  "epochs_to_avg_over": 10,
  "test_after_epochs": 1
  },
  "batch_size": 32,
  "train_sets": ["train"],
  "test_sets": ["dev", "test"]
  }
}
```

The following keys in the JSON file need to be specified:

dataset_name	Name of the folder containing the output of the *ned-wiki-export.py* script.
model_name	Name of the trained model. If a model with the same name already exists, it gets trained further.
params	Contains hyperparameter specifications of the neural network. The hyperparameter names should be self explanatory.
metric_key	The metrics based on which the model should be optimized.
flip_sign_of_metric	Is true if a "smaller than" comparison of the metric should be used.
end_if_slope_less_than	The training will be stopped if the slope of the used metric is smaller than the specified value.
epochs_to_avg_over	Averages the value based on the specified number of previous epochs.
test_after_epochs	The number of epochs that the model should be trained on before results get validated with the test set.

B. Expert Interview Guideline

Interview Leitfaden – Juristische Recherche (45+ min)

Leitende Forschungsfrage: Ist die Unterscheidung zwischen Wort Sinnhaftigkeit im Recht wichtig, und welche Anwendungsmöglichkeiten sind für Juristen attraktiv?

Name:

Datum:

Zeit Beginn:

Zeit Ende:

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). Fokus auf deutschen Rechtsbegriffen • Aktueller Stand: Bewertung der Relevanz + Suche nach Anwendungsmöglichkeiten für Juristen	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	
Wie schätzen Sie Ihre IT Kenntnisse ein?	
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	
Welche Tools / Datenbanken verwenden Sie und warum?	
Können Sie mir beispielhaft die einzelnen Schritte zeigen wie Sie bei der Suche vorgehen? Inwiefern unterscheiden sich die Suchbegriffe die Sie eingeben von e.g. privat motivierten Google Suchen (e.g. München Wetter, ...)	
Wie stark grenzen Sie die Suchergebnisse ein, bzw. wie viele Ergebnisse schauen Sie sich an und wie genau schauen Sie sich diese an?	
Welche Features sind für Sie bei der Einschränkung des Suchraums am hilfreichsten, und wo gibt es Ihrer Meinung nach noch Mängel? **Falls Mängel:** Haben Sie nach Lösungen / Workarounds gesucht um dieses Problem möglichst gut in den Griff zu bekommen?	
Bei welchem Schritt bei der Suche besteht Ihrer Meinung nach das größte Beschleunigungspotential?	

WORD SENSES IM RECHT	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Gibt es Beispiele für Wörter mit unterschiedlichen Senses im Recht (innerhalb eines Rechtsgebiet und rechtsgebietsübergreifend) Evtl. Beispiel: Gewalt, Berufungsgrund **Wenn Ja:** Wie relevant schätzen Sie die Möglichkeit zur Sinnunterscheidung deutscher Rechtsbegriffe vor?	
Würde eine Einschränkung der Suchergebnisse nach Senses von nicht-juristischen Fachbegriffen helfen?	
Fallen Ihnen Anwendungsfälle ein wo eine Kontextunterscheidung der Suchergebnisse hilfreich wäre?	
Würden Vorschläge von verwandten Suchbegriffen helfen? (Bsp. Google Bildersuche)	
Würde eine Synonymsuche helfen?	
Fällt Ihnen spontan noch ein Feature ein, welches mithilfe von Kontexterkennen, Synonymen, Wortbedeutungen oder dergleichen die juristische Recherche für Sie verbessern könnte?	
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Kontext bereits bei Query Eingabe, oder als Zusatzinformationen bei den Ergebnissen?)	
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

C. Expert Interview Results

Interview Leitfaden – Juristische Recherche (45+ min)

Leitende Forschungsfrage: Ist die Unterscheidung zwischen Wort Sinnhaftigkeit im Recht wichtig, und welche Anwendungsmöglichkeiten sind für Juristen attraktiv?

Name:

Datum: 14.12.2020

Zeit Beginn: 10:00

Zeit Ende: 12:05

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie Vorkommen (Beispiel Bank, Berufungsgrund). Fokus auf deutschen Rechtsbegriffen • Aktueller Stand: Bewertung der Relevanz + Suche nach Anwendungsmöglichkeiten für Juristen	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	55, Rechtsanwalt
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	27 Jahre
Wie schätzen Sie Ihre IT Kenntnisse ein?	Anfänger, UND-ODER Verknüpfungen sind das Limit
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Arbeitsrecht Gesellschaftsrecht, Erbrecht, Berufsrecht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Kommentare und Großkommentare
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	Starke Schwankung, 5 bis 7 Stunden pro Woche. Es gibt aber durchaus Wochen, wo er nichts anderes macht als recherchieren.
Welche Tools / Datenbanken verwenden Sie und warum?	Beck Online und Juris Arbeitet noch viel mit Printmedien Google bei neueren Sachen, e.g. Corona
Können Sie mir beispielhaft die einzelnen Schritte zeigen wie Sie bei der Suche vorgehen? Inwiefern unterscheiden sich die Suchbegriffe die Sie eingeben von e.g. privat motivierten Google Suchen (e.g. München Wetter, ...)	Bearbeitet keine Fälle in Rechtsgebieten die er gar nicht kennt. Benutzt Kommentare und Urteile um in einen Fall einzusteigen. Suchstrategie: Schaut zuerst in Handbuch oder kleinem Kommentar nach ob er etwas zu dem Thema findet. Sucht sich dann die richtigen Schlagwörter heraus und sucht mit denen in einem Großkommentar weiter. Hat sehr gutes Verständnis für Stichwortsuche (überlegt sich auch welche analogen Begriffe / Synonyme es geben könnte). Schränkt dann den Pool an Schlagwörtern noch mehr ein und sucht dann in Fachdatenbanken, e.g. Beck Online. Gleich mit einem großem Suchmedium (Großkommentar, Datenbank) anzufangen wäre zu Diffus, da zu viele Ergebnisse zurückgeliefert werden. Erst wenn er sich sicher genug fühlt in einem Rechtsgebiet, fängt er an online zu recherchieren.
Wie stark grenzen Sie die Suchergebnisse ein, bzw. wie viele Ergebnisse schauen Sie sich an und wie genau schauen Sie sich diese an?	-

Welche Features sind für Sie bei der Einschränkung des Suchraums am hilfreichsten, und wo gibt es Ihrer Meinung nach noch Mängel? **Falls Mängel:** Haben Sie nach Lösungen / Workarounds gesucht um dieses Problem möglichst gut in den Griff zu bekommen?	Mängel: Bevorzugt Printmedien, da UND-ODER Verknüpfungen zu schwer zu benutzen sind. Fachdatenbanken sind für eine „ich fang einfach mal an zu suchen“ Suche nicht zu gebrauchen, da Ergebnisse zu diffus.
Bei welchem Schritt bei der Suche besteht Ihrer Meinung nach das größte Beschleunigungspotential?	Problem: Er bekommt viele nicht relevanten Ergebnisse, selbst wenn er weiß wo er suchen muss und sogar den genauen Dokumentnamen eingibt. Bei der Relevanzbewertung der Ergebnisse besteht noch viel Potential.
WORD SENSES IM RECHT	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Gibt es Beispiele für Wörter mit unterschiedlichen Senses im Recht (innerhalb eines Rechtsgebiet und rechtsgebietsübergreifend) Evtl. Beispiel: Gewalt, Berufungsgrund **Wenn Ja:** Wie relevant schätzen Sie die Möglichkeit zur Sinnunterscheidung deutscher Rechtsbegriffe vor?	Im Recht gibt es viele Wörter mit unterschiedlichen Bedeutungen. Bsp: Urlaub Urlaubssenat bei Bundesarbeitsgericht hat eine andere Auffassung von Urlaub als der Senat der sich mit Arbeitsentgelt beschäftigt. D.h. Kontextunterscheidung sehr wichtig. Urlaub kann mehrere Kontexte haben, e.g. unterschiedliche Bedeutung im Kontext von Arbeitsentgelt oder Sozialversicherungsrecht. Urteile zu Urlaub lesen sich alle gleich, aber kommen zu unterschiedlichen Ergebnissen je nachdem welcher Kontext angewendet wird. z.B. EuGH: sieht Urlaub z.B. mehr als Schutzmaßnahme um Gesundheit des Mitarbeiters zu erhalten. Zurückzuführen auf unterschiedliche Meinungsstränge während der Entwicklung der Gesetze. Es wäre toll wenn man Gesetze nach diesen Meinungssträngen kategorisieren kann.
Würde eine Einschränkung der Suchergebnisse nach Senses von nicht-juristischen Fachbegriffen helfen?	-
Fallen Ihnen Anwendungsfälle ein wo eine Kontextunterscheidung der Suchergebnisse hilfreich wäre?	Ja (siehe oben)
Würden Vorschläge von verwandten Suchbegriffen helfen? (Bsp. Google Bildersuche)	Ja + Erweiterung mihilfe Überbegriffen (Hypernyme)
Würde eine Synonymsuche helfen?	Ja
Fällt Ihnen spontan noch ein Feature ein, welches mithilfe von Kontexterkennen, Synonymen, Wortbedeutungen oder dergleichen die juristische Recherche für Sie verbessern könnte?	Automatische Subsumtion wäre hilfreich
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Kontext bereits bei Query Eingabe, oder als Zusatzinformationen bei den Ergebnissen?)	Abhängig von der Anzahl der potentiellen Ergebnisse
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Aggressive Verkaufstaktik von Beck nervig. Nicht nur gute Recherchefähigkeiten sind als Anwalt wichtig, sondern psychologische / menschlichen Aspekte spielen eine wichtige Rolle. Es geht darum herauszufinden, was der Mandant am dringendsten braucht und was seine konkreten Probleme sind. Es geht darum Wünsche des Mandanten zu erfüllen und wie man ihn optimal vertreten kann. Mit den einzelnen Parteien im Interesse des Mandanten sprechen erfordert viel Geschick. EinTelefonat mit "mächtigen" Entscheidungsträgern kann einen Fall schnell beenden. Argumentation mit Rechtsnormen führen meistens dazu, dass die Gegenseite mit Rechtsnormen zurückschlägt. Seine Erfahrung: Junge Juristen haben schlechtere Suchstrategien, können schlechter juristisch und terminologisch strukturiert arbeiten.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Juristische Recherche (45+ min)

Leitende Forschungsfrage: Ist die Unterscheidung zwischen Wort Sinnhaftigkeit im Recht wichtig, und welche Anwendungsmöglichkeiten sind für Juristen attraktiv?

Name:

Datum: 04.01.2021

Zeit Beginn: 11:00

Zeit Ende: 12:30

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie Vorkommen (Beispiel Bank, Berufungsgrund). Fokus auf deutschen Rechtsbegriffen • Aktueller Stand: Bewertung der Relevanz + Suche nach Anwendungsmöglichkeiten für Juristen	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	29, Rechtsreferendarin
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	5+ Jahre Erfahrung im Rechtsbereich
Wie schätzen Sie Ihre IT Kenntnisse ein?	Sehr technikaffin, kann in Python programmieren und hat diverse technische Fortbildungen besucht
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	IT Recht / Datenschutzrecht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Entscheidungen Kommentare Rezensionen von Bücher Besprechungen und Anmerkungen Normen
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	starke Schwankungen, im Schnitt ca. 6 Stunden
Welche Tools / Datenbanken verwenden Sie und warum?	Beck Online (sehr häufig) Juris (weil mehr Funktionen in bestimmten Bereichen. Besser zum Recherchieren von Entscheidungen + es sind Besprechungen/Anmerkungen zur Entscheidungen aufgelistet.) Gerichtsdatenbanken vom BGH (man kann sich Reminder setzen) Google
Können Sie mir beispielhaft die einzelnen Schritte zeigen wie Sie bei der Suche vorgehen? Inwiefern unterscheiden sich die Suchbegriffe die Sie eingeben von e.g. privat motivierten Google Suchen (e.g. München Wetter, ...)	Hauptunterschied zwischen juristischen Recherche und normaler Goolge Suche: bei Google Suche gibt es meist eindeutiges Ergebnis Schaut als erstes ob es schon etwas zu der Suche gibt. Überlegt sich mögliche Suchbegriffe die verkettet werden. Schaut sich dann die erste Seite der Suchergebnisse an und bewertet die Ergebnisse. Passt dann ggf. Suchbegriffe an. Filtert die Entscheidungen ggf. nach Gericht (e.g. BGH) und Datum. Verwendet auch logische Verknüpfungen („und“, „oder“) .
Wie stark grenzen Sie die Suchergebnisse ein, bzw. wie viele Ergebnisse schauen Sie sich an und wie genau schauen Sie sich diese an?	Fängt zuerst breit an und lässt sich von den Ergebnissen inspirieren, und sortiert dann erst im Nachhinein.

Welche Features sind für Sie bei der Einschränkung des Suchraums am hilfreichsten, und wo gibt es Ihrer Meinung nach noch Mängel? **Falls Mängel:** Haben Sie nach Lösungen / Workarounds gesucht um dieses Problem möglichst gut in den Griff zu bekommen?	Verwendet hauptsächlich Sortierungsfeatures, e.g. Sortierung nach Relevanz oder Datum. Ergebnisse sollten gekennzeichnet sein wenn man sie sich schon angesehen hat. Benachrichtigungen per Email geben lassen für neue bestimmte Inhalte (aber sehr Fehleranfällig) Viele nicht relevante Ergebnisse (einerseits wegen Mehrdeutigkeit, falsche Suchbegriffe wenn man in ein neues Gebiet einsteigt, gleiche Begriffe in unterschiedlichen Rechtsgebieten haben unterschiedliche Bedeutung) Oft wegen Rechtsbehelfsbelehrung werden irrelevante Ergebnisse geliefert Copy-Paste Funktion von Urteil in schönem Format wäre wünschenswert und erhöht die Weiterverarbeitbarkeit.
Bei welchem Schritt bei der Suche besteht Ihrer Meinung nach das größte Beschleunigungspotential?	-
WORD SENSES IM RECHT	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Gibt es Beispiele für Wörter mit unterschiedlichen Senses im Recht (innerhalb eines Rechtsgebiet und rechtsgebietsübergreifend) Evtl. Beispiel: Gewalt, Berufungsgrund **Wenn Ja:** Wie relevant schätzen Sie die Möglichkeit zur Sinnunterscheidung deutscher Rechtsbegriffe vor?	Jura Sprache sind Deutsche Wörter mit einer ganz anderen Bedeutung als in der Alltagssprache. Übersetzung schwierig: Juristische Sprache in Alltagssprache 2 unterschiedliche Arten von Mehrdeutigkeiten: Mehrdeutig im juristischen Kontext und in der natürlichen Sprache (letzteres zu unterscheiden ist wichtiger): z.B. das Wort „Last“ kann bedeuten Darlegungslast oder Traglast. Gibt ihrer Meinung unzählige Begriffe wo dieses Problem der Sinnunterscheidung besteht.
Würde eine Einschränkung der Suchergebnisse nach Senses von nicht-juristischen Fachbegriffen helfen?	Ja (siehe oben)
Fallen Ihnen Anwendungsfälle ein wo eine Kontextunterscheidung der Suchergebnisse hilfreich wäre?	Ja (siehe oben)
Würden Vorschläge von verwandten Suchbegriffen helfen? (Bsp. Google Bildersuche)	Ja, aber in geringerem Maße. Wenn sie einen Begriff eingibt dann meint sie genau diesen speziellen Begriff.
Würde eine Synonymsuche helfen?	Ja, aber in geringerem Maße
Fällt Ihnen spontan noch ein Feature ein, welches mithilfe von Kontexterkennen, Synonymen, Wortbedeutungen oder dergleichen die juristische Recherche für Sie verbessern könnte?	-
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Kontext bereits bei Query Eingabe, oder als Zusatzinformationen bei den Ergebnissen?)	Kontextmenü bei den Suchergebnissen wäre hilfreich, wo man sieht in welchen Kontexten kommen Wörter vor. Am besten bekommt man die Sinnhaftigkeitsinformation spätestens mit den Ergebnissen als Zusatzinformation (e.g. mittels Schlagwörtern), damit man sich nicht erst die Ergebnisse anschauen muss, aber auch schon bei der Eingabe in den Suchschlitz möglich. Man soll mehr mit optischen Sachen arbeiten, Kennzeichnung mit unterschiedlichen Farben, da das Auge dadurch schneller filtern kann.
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Aggressive Upselling-Taktik von Beck Online ist nervig Was auch häufig vorkommt: am Ende von Entscheidungen gibt es Rechtsbehelfsbelehrungen. In den Belehrungen stecken bestimmte Wörter und Normen. Die Suchmaschine wertet ebenfalls die Rechtsbehelfsbelehrung aus und liefert irrelevante Entscheidungen zurück, wo die Suchbegriffe nur in der Rechtsbehelfsbelehrung vorkommen.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Juristische Recherche (45+ min)

Leitende Forschungsfrage: Ist die Unterscheidung zwischen Wort Sinnhaftigkeit im Recht wichtig, und welche Anwendungsmöglichkeiten sind für Juristen attraktiv?

Name:

Datum: 12.01.2021

Zeit Beginn: 16:00

Zeit Ende: 17:00

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). Fokus auf deutschen Rechtsbegriffen • Aktueller Stand: Bewertung der Relevanz + Suche nach Anwendungsmöglichkeiten für Juristen	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	26, Head of IT (Schnittstelle Legal Tech) in einer NGO
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	-
Wie schätzen Sie Ihre IT Kenntnisse ein?	Sehr gute IT Kenntnisse
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	IT-Recht, Strafrecht, Datenschutzrecht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Normen Kommentare Urteile
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	Zwischen 5 bis 10 Stunden
Welche Tools / Datenbanken verwenden Sie und warum?	Beck Online (wegen Preis und weil es umfänglicher ist als z.B. Juris) Mängel: viele nicht relevante Suchergebnisse, die Suchmaschine bei Beck Online ist nicht gut
Können Sie mir beispielhaft die einzelnen Schritte zeigen wie Sie bei der Suche vorgehen? Inwiefern unterscheiden sich die Suchbegriffe die Sie eingeben von e.g. privat motivierten Google Suchen (e.g. München Wetter, ...)	Schaut zuerst ob es ein ähnliches Urteil schon gibt, am Anfang ohne etwas einzuschränken, sucht dann ohne Filterfunktionen (Kombination Begriff und einschlägiger Paragraph) Suche ähnelt bei ihr einer Google Suche, Kombination von Stichwörtern
Wie stark grenzen Sie die Suchergebnisse ein, bzw. wie viele Ergebnisse schauen Sie sich an und wie genau schauen Sie sich diese an?	Schaut sich in der Regel drei Seiten Ergebnisse an Wenn sie nach bestimmten Inhalten sucht, geht sie in die Ergebnisse rein und sucht Begriffe über die Browser Suchfunktion
Welche Features sind für Sie bei der Einschränkung des Suchraums am hilfreichsten, und wo gibt es Ihrer Meinung nach noch Mängel? **Falls Mängel:** Haben Sie nach Lösungen / Workarounds gesucht um dieses Problem möglichst gut in den Griff zu bekommen?	Filter nach Rechtsgebiet und Datum. Ihrer Meinung deckt Beck Online alle notwendigen Filteroptionen für ab.

Bei welchem Schritt bei der Suche besteht Ihrer Meinung nach das größte Beschleunigungspotential?	Es wäre wünschenswert wenn die Suchfunktion mit der Zeit lernt, e.g. wenn die Person fast immer im Strafrecht sucht, werden nur Ergebnisse aus dem Strafrecht angezeigt. Z.B. auch berücksichtigen, wie Personen mit ähnlich Suchverläufen verfahren sind.
WORD SENSES IM RECHT	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Gibt es Beispiele für Wörter mit unterschiedlichen Senses im Recht (innerhalb eines Rechtsgebiet und rechtsgebietsübergreifend) Evtl. Beispiel: Gewalt, Berufungsgrund **Wenn Ja:** Wie relevant schätzen Sie die Möglichkeit zur Sinnunterscheidung deutscher Rechtsbegriffe vor?	Es gibt mehrdeutige Rechtsbegriffe, aber: Sinnunterscheidung bei Rechtsbegriffen wenig hilfreich, da sich der Sinn oft schon aus der Dokumentenart oder dem Rechtsgebiet ergibt. Z.B. der Sinn für vom Wort „Gewalt“ in einem Urteil welches an einem Strafgericht erlassen wurde ist bereits eindeutig.
Würde eine Einschränkung der Suchergebnisse nach Senses von nicht-juristischen Fachbegriffen helfen?	Mit Sicherheit, teilweise kann man den Kontext durch Dokumentenart filtern, aber bei Wörtern der natürliche Sprache nur begrenzt. Sinnhaftigkeitsfilter von Wörtern wäre hilfreich.
Fallen Ihnen Anwendungsfälle ein wo eine Kontextunterscheidung der Suchergebnisse hilfreich wäre?	Spontan nicht.
Würden Vorschläge von verwandten Suchbegriffen helfen? (Bsp. Google Bildersuche)	Ja, vor allem wen man gar nicht weiß wonach man sucht. Vermutlich sehr interessant für Studierende, da diese die juristische Bezeichnung oftmals nicht wissen. Die passenden juristischen Fachbegriffe zu finden ist manchmal ein eigener Rechercheakt an sich.
Würde eine Synonymsuche helfen?	Wäre hilfreich, wenn synonyme Schlagwörter angezeigt werden.
Fällt Ihnen spontan noch ein Feature ein, welches mithilfe von Kontexterkennen, Synonymen, Wortbedeutungen oder dergleichen die juristische Recherche für Sie verbessern könnte?	Im Urteilsbereich wäre es sehr hilfreich wenn man sehen würde, ob ein Urteil in einer höheren Gerichtsinstanz schon behandelt wurde. Lebensstruktur eines Urteils sollte man transparent nachverfolgen können.
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Kontext bereits bei Query Eingabe, oder als Zusatzinformationen bei den Ergebnissen?)	Nach der Query Eingabe. Zuerst sollen alle möglichen Ergebnisse angezeigt werden, damit man dann sieht welche Ergebnisse herausfliegen wenn man den Sense Filter verwendet.
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Anmerkung: Beck Online hat zu viele Filtermöglichkeiten, man kommt gar nicht dazu sich Gedanken darüber zu machen wie man am geschicktesten filtert. Sie kann sich vorstellen, dass ein Sense Filter für viele sehr hilfreich wäre.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Juristische Recherche (45+ min)

Leitende Forschungsfrage: Ist die Unterscheidung zwischen Wort Sinnhaftigkeit im Recht wichtig, und welche Anwendungsmöglichkeiten sind für Juristen attraktiv?

Name:

Datum: 22.12.2020

Zeit Beginn: 17:00

Zeit Ende: 18:00

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie Vorkommen (Beispiel Bank, Berufungsgrund). Fokus auf deutschen Rechtsbegriffen • Aktueller Stand: Bewertung der Relevanz + Suche nach Anwendungsmöglichkeiten für Juristen	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	44, Anwalt
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	16 Jahre
Wie schätzen Sie Ihre IT Kenntnisse ein?	Gut, arbeitet momentan an Projekt zur automatischen Vertragsgenerierung und kann ein bisschen Programmieren.
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Vergaberecht, Einkaufsrecht (Kaufrecht, Schuldrecht, Vertragsrecht) Sein typischer Arbeitsalltag (Rechtsabteilung Deutsche Bahn): Hilft beim Einkauf von Betriebsfahrzeugen, kümmert sich um alles was Schienenfahrzeuge angeht (Einkauf, Mangelansprüche, etc…). 50% seiner Arbeitszeit sind Besprechungen / Organisation und deren Vorbereitung. Rechtliches Prüfen nimmt 1/3 seiner Arbeitszeit ein.
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Arbeitet am meisten mit Kommentaren, und geht dann darüber manchmal in Rechtsprechungen.
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	Sehr unterschiedlich, aber gemittelt über das Jahr eine Stunde. Viele Sachen wiederholen sich und man kennt sie nach der Zeit.
Welche Tools / Datenbanken verwenden Sie und warum?	Beck Online und Google. Google eignet sich gut um ein erstes Gefühl zu bekommen was man braucht. Suchmaschine in Google ist besser als die von Beck und eignet sich daher besser, um sich an ein Thema heranzutasten.
Können Sie mir beispielhaft die einzelnen Schritte zeigen wie Sie bei der Suche vorgehen? Inwiefern unterscheiden sich die Suchbegriffe die Sie eingeben von e.g. privat motivierten Google Suchen (e.g. München Wetter, ...)	Seine rechtlichen Google Suchen unterscheiden sich wenig von privat motivierten Google Suchen. Z.B. wenn er etwas über Regelungen für die Zulassung von Schienenfahrzeugen wissen will, gibt er ein „Zulassung Schienenfahrzeug“ und hantelt sich damit voran bis zum richtigen Fachbegriff („Inbetriebnahmegenehmigung“). Bei Beck: Problem: Wenn man die richtigen Stichwörter nicht genau kennt, kommt man gar nicht zu relevanten Ergebnissen. Probiert dann mit Wörtern herum (Zulassung, Genehmigung, Projektantenstellung, …).
Wie stark grenzen Sie die Suchergebnisse ein, bzw. wie viele Ergebnisse schauen Sie sich an und wie	„So viele wie nötig“. Wenn er die Schlagwörter kennt, dann reicht es 3 oder 4 Ergebnisse genauer anzuschauen.

genau schauen Sie sich diese an?	Wenn er nicht genau weiß wonach er suchen muss, schaut er sich zuerst einmal 10 bis 12 Ergebnisse bevor er einige Anhaltspunkte hat. Orientiert sich hauptsächlich an den Überschriften, Datum und Normketten. Benutzt keine Filter / Features, da es zu aufwändig ist (ausser Datum, wegen Rechtsreformen, z.B. 2016 wurde Vergaberecht komplett neu aufgesetzt). Zum Filtern muss er schon ca. Wissen wo er suchen muss.
Welche Features sind für Sie bei der Einschränkung des Suchraums am hilfreichsten, und wo gibt es Ihrer Meinung nach noch Mängel? **Falls Mängel:** Haben Sie nach Lösungen / Workarounds gesucht um dieses Problem möglichst gut in den Griff zu bekommen?	Hilfreiches Feature: Man sollte Wörter der natürlichen Sprache verwenden können um in juristischen Fachdatenbanken zu suchen. Hilfe bei der Abstraktion wäre auch hilfreich. Beck Suchfunktion ist nur gut, wenn man die richtigen Stichwörter hat. Größter Mangel: ohne das genaue Stichwort keine brauchbaren Ergebnisse.
Bei welchem Schritt bei der Suche besteht Ihrer Meinung nach das größte Beschleunigungspotential?	-
WORD SENSES IM RECHT	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Gibt es Beispiele für Wörter mit unterschiedlichen Senses im Recht (innerhalb eines Rechtsgebiet und rechtsgebietsübergreifend) Evtl. Beispiel: Gewalt, Berufungsgrund **Wenn Ja:** Wie relevant schätzen Sie die Möglichkeit zur Sinnunterscheidung deutscher Rechtsbegriffe vor?	Ja, Bsp: „Nachforderung“ einerseits im Vergaberecht, dem BGB oder ZPO. Manche Begriffe haben Fundstellen aus vielen verschiedenen Rechtsgebieten. Sinnunterscheidung von Rechtsbegriffen könnte für Anfänger / ungeübte Juristen Sinn machen.
Würde eine Einschränkung der Suchergebnisse nach Senses von nicht-juristischen Fachbegriffen helfen?	Es könnte helfen. Es kommt manchmal vor dass Wörter der natürlichen Sprache in unerwünschten Kontexten vorkommt, da würde so etwas helfen.
Fallen Ihnen Anwendungsfälle ein wo eine Kontextunterscheidung der Suchergebnisse hilfreich wäre?	-
Würden Vorschläge von verwandten Suchbegriffen helfen? (Bsp. Google Bildersuche)	Ja
Würde eine Synonymsuche helfen?	Ja
Fällt Ihnen spontan noch ein Feature ein, welches mithilfe von Kontexterkennen, Synonymen, Wortbedeutungen oder dergleichen die juristische Recherche für Sie verbessern könnte?	-
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Kontext bereits bei Query Eingabe, oder als Zusatzinformationen bei den Ergebnissen?)	Bereits im Suchschlitz.
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Mehr mit Grafiken arbeiten, Ergebnisse visuell/optischer darstellen, nicht so textbasiert.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Juristische Recherche (45+ min)

Leitende Forschungsfrage: Ist die Unterscheidung zwischen Wort Sinnhaftigkeit im Recht wichtig, und welche Anwendungsmöglichkeiten sind für Juristen attraktiv?

Name:

Datum: 16.12.2020

Zeit Beginn: 13:00

Zeit Ende: 13:35

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Nein
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie Vorkommen (Beispiel Bank, Berufungsgrund). Fokus auf deutschen Rechtsbegriffen • Aktueller Stand: Bewertung der Relevanz + Suche nach Anwendungsmöglichkeiten für Juristen	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	35, Rechtsanwältin
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	2 ½ Jahre als Rechtsanwältin
Wie schätzen Sie Ihre IT Kenntnisse ein?	Anwenderkenntnisse (Anfänger, Basic)
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Patentrecht, Vertretung von Unternehmen vor Gericht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Normen (mit geringer Bandbreite). Kommentare, Aufsätze, Rechtsprechungen
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	im Schnitt 3 Stunden pro Woche (Arbeitet momentan Teilzeit, 6h wenn sie Vollzeit arbeiten würde). Starke Schwankungen. Als Anfänger mehr, mit der Zeit ist man geübter und kennt die Normen/Schlagwörter und braucht weniger lang zum Recherchieren.
Welche Tools / Datenbanken verwenden Sie und warum?	Beck Online, Wolters Kluwer, Patentdatenbanken
Können Sie mir beispielhaft die einzelnen Schritte zeigen wie Sie bei der Suche vorgehen? Inwiefern unterscheiden sich die Suchbegriffe die Sie eingeben von e.g. privat motivierten Google Suchen (e.g. München Wetter, ...)	Wenn man das Stichwort kennt, sucht sie zuerst in Standardwerken (e.g. Standardkommentar BGB, Palandt, Hardcover Werke). Wenn sie tiefer Suchen möchte, schaut sie zu den Paragraphen ausführliche Kommentare an (e.g. Münchener Kommentar, da dieser sehr ausführlich ist). Ansonsten klickt sie sich durch Entscheidungen durch. Bei juristischer Recherche benutzt sie engere Fachbegriffe. Bei Beck Online: Arbeitet auch mit logischen Verknüpfungen Verwendet Filter Anders als bei Google suche kann man oft nicht sofort einschätzen, ob Ergebnisse relevant sind oder nicht.
Wie stark grenzen Sie die Suchergebnisse ein, bzw. wie viele Ergebnisse schauen Sie sich an und wie genau schauen Sie sich diese an?	-

Welche Features sind für Sie bei der Einschränkung des Suchraums am hilfreichsten, und wo gibt es Ihrer Meinung nach noch Mängel? **Falls Mängel:** Haben Sie nach Lösungen / Workarounds gesucht um dieses Problem möglichst gut in den Griff zu bekommen?	Schlecht bei Beck Online: Ergebnisse passen manchmal gar nicht zu dem was man finden möchte, der Kontext wird nicht beachtet. Man muss aber trotzdem auf die Ergebnisse klicken, nur um dann festzustellen dass sie nicht relevant sind. Bei Kommentaren bei Beck: Jeder Absatz / jede Untergliederung muss einzeln aufgerufen/aufgeklappt werden. Man muss die Dokumente manuell aufrollen.
Bei welchem Schritt bei der Suche besteht Ihrer Meinung nach das größte Beschleunigungspotential?	Bei der Relevanzbestimmung der Ergebnisse
WORD SENSES IM RECHT	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Gibt es Beispiele für Wörter mit unterschiedlichen Senses im Recht (innerhalb eines Rechtsgebiet und rechtsgebietsübergreifend) Evtl. Beispiel: Gewalt, Berufungsgrund **Wenn Ja:** Wie relevant schätzen Sie die Möglichkeit zur Sinnunterscheidung deutscher Rechtsbegriffe vor?	Ja, Anfechtung: im BGB und im Insolvenzrecht. Es gibt durchaus Mehrdeutigkeiten im Recht. Vorgefertigte Kontexte nach denen man Filtern könnte wäre praktisch.
Würde eine Einschränkung der Suchergebnisse nach Senses von nicht-juristischen Fachbegriffen helfen?	Ja, aber mögliches Problem: normaler Sprachgebrauch und juristischer Sprachgebrauch nicht immer deckungsgleich.
Fallen Ihnen Anwendungsfälle ein wo eine Kontextunterscheidung der Suchergebnisse hilfreich wäre?	Ja, siehe oben.
Würden Vorschläge von verwandten Suchbegriffen helfen? (Bsp. Google Bildersuche)	Vorschläge von verwandten Schlagwörtern hilfreich
Würde eine Synonymsuche helfen?	Vorschläge von verwandten Schlagwörtern hilfreich
Fällt Ihnen spontan noch ein Feature ein, welches mithilfe von Kontexterkennen, Synonymen, Wortbedeutungen oder dergleichen die juristische Recherche für Sie verbessern könnte?	Spontan nicht
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Kontext bereits bei Query Eingabe, oder als Zusatzinformationen bei den Ergebnissen?)	Am hilfreichsten schon bei der Eingabe in den Suchschlitz, das würde einen Arbeitsschritt sparen.
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Tut vor der Suche abstrahieren, es wäre hilfreich sich den Abstraktionsschritt sparen zu können. Wirtschaftlicher Druck seitens der Mandanten, Suche muss schnell gehen.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Juristische Recherche (45+ min)

Leitende Forschungsfrage: Ist die Unterscheidung zwischen Wort Sinnhaftigkeit im Recht wichtig, und welche Anwendungsmöglichkeiten sind für Juristen attraktiv?

Name:

Datum: 23.12.2020

Zeit Beginn: 18:00

Zeit Ende: 19:00

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie Vorkommen (Beispiel Bank, Berufungsgrund). Fokus auf deutschen Rechtsbegriffen • Aktueller Stand: Bewertung der Relevanz + Suche nach Anwendungsmöglichkeiten für Juristen	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	Alter nicht genannt (geschätzt 29) Werdegang: 4+ Jahre Erfahrung im Rechtsbereich
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	2 Jahre
Wie schätzen Sie Ihre IT Kenntnisse ein?	Gute Kenntnisse, Erfahrung im IT Bereich
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Handelsrecht, Vertragsrecht, Gesellschaftsrecht, Internationales Recht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Keine Urteile, Muster, Kommentare
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	Variiert stark, im Schnitt 5 Stunden pro Woche
Welche Tools / Datenbanken verwenden Sie und warum?	Beck Online Haufe Bevorzugt Beck Online, sie mag die Übersicht bei Beck Online und das Design gefällt ihr gut. Auch Gewohnheitssache, da sie Beck Online als erstes Tool benutzt hatte.
Können Sie mir beispielhaft die einzelnen Schritte zeigen wie Sie bei der Suche vorgehen? Inwiefern unterscheiden sich die Suchbegriffe die Sie eingeben von e.g. privat motivierten Google Suchen (e.g. München Wetter, ...)	Stichwortartige Suche, e.g. „aufschiebende Bedingung Vertragsrecht“, mit juristischen Fachbegriffen Muss 2 bis 3 mal die Suche anpassen bis die Ergebnisse passen Benutzt Filter in Beck Online Wenn sie nicht vertraut ist mit einem Thema, schaut sie sich Kommentare an (bevorzugt Münchener Kommentar)
Wie stark grenzen Sie die Suchergebnisse ein, bzw. wie viele Ergebnisse schauen Sie sich an und wie genau schauen Sie sich diese an?	-

Welche Features sind für Sie bei der Einschränkung des Suchraums am hilfreichsten, und wo gibt es Ihrer Meinung nach noch Mängel? **Falls Mängel:** Haben Sie nach Lösungen / Workarounds gesucht um dieses Problem möglichst gut in den Griff zu bekommen?	Filter nach Dokumentart und Datum. Suche soll mehr personalisiert werden, Suchintention soll automatisch erkannt werden.
Bei welchem Schritt bei der Suche besteht Ihrer Meinung nach das größte Beschleunigungspotential?	-
WORD SENSES IM RECHT	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Gibt es Beispiele für Wörter mit unterschiedlichen Senses im Recht (innerhalb eines Rechtsgebiet und rechtsgebietsübergreifend) Evtl. Beispiel: Gewalt, Berufungsgrund **Wenn Ja:** Wie relevant schätzen Sie die Möglichkeit zur Sinnunterscheidung deutscher Rechtsbegriffe vor?	Ja, Rechtsbegriffe haben unterschiedliche Bedeutungen in unterschiedlichen Rechtsgebieten.
Würde eine Einschränkung der Suchergebnisse nach Senses von nicht-juristischen Fachbegriffen helfen?	-
Fallen Ihnen Anwendungsfälle ein wo eine Kontextunterscheidung der Suchergebnisse hilfreich wäre?	Ja, eine Kontexterkennung würde helfen, z.B. beim Suchen im Bereich Corporate Law kommen viele Ergebnisse aus dem Familienrecht, es wäre schön wenn man das filtern könnte.
Würden Vorschläge von verwandten Suchbegriffen helfen? (Bsp. Google Bildersuche)	Ja
Würde eine Synonymsuche helfen?	-
Fällt Ihnen spontan noch ein Feature ein, welches mithilfe von Kontexterkennen, Synonymen, Wortbedeutungen oder dergleichen die juristische Recherche für Sie verbessern könnte?	-
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Kontext bereits bei Query Eingabe, oder als Zusatzinformationen bei den Ergebnissen?)	Schon beim Eingeben in den Suchschlitz.
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Interface sollte übersichtlich gehalten werden, lieber 5 Features die einfach zum Anpassen sind als dass Interface dann Überladen ist.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

D. Qualitative Evaluation Guideline

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum:

Zeit Beginn:

Zeit Ende:

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	
Wie schätzen Sie Ihre IT Kenntnisse ein?	
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden Sie Filter?	
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt? Welche Verbesserung würden Sie sich wünschen?	
Fallen Ihnen Begriffe ein, nach denen Sie in letzter Zeit gesucht haben, bzw. nach denen Sie häufiger Suchen? Präsentation beispielhafter Filtermöglichkeiten.	

Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

E. Qualitative Evaluation Results

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum: 06.07.2021

Zeit Beginn: 11:00

Zeit Ende: 11:30

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	36, Rechtsberaterin
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	6 Jahre
Wie schätzen Sie Ihre IT Kenntnisse ein?	Sehr gut
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Internationales Privatrecht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Urteile
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	20 Stunden / Woche im Schnitt
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden sie Filter?	Internationale Fachdatenbanken, ist mehr in Englischen Fachdatenbanken unterwegs. Benutzt kaum Filter
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	Verfahren: Zivilrechtliches Verfahren Berufung: Rechtsmittel Revision: Rechtsmittel Anlage: Kapitalanlage
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt? Welche Verbesserung würden Sie sich wünschen?	Verfahren: Sinnhaftigkeiten klar, aber Unterschied zwischen 2 und 5, und 4 und 7 ist nicht eindeutig erkennbar. Berufung: alle Sinnhaftigkeiten klar. Revision: Kennt nicht alle Begriffe, würde Begriffe die sie nicht kennt googlen oder als Filter auswählen und schauen was dabei rauskommt. Anlage: Sinnhaftigkeiten sind klar, aber Unterschied zwischen 4 und 5 ist nicht klar. Verbesserungsvorschläge: Filter spezifischer auf Rechtsbereiche mit feinerer Aufspaltung Nur Relevante Senses anzeigen Senses ordnen nach Relevanz
Fallen Ihnen Begriffe ein, nach denen Sie in letzter Zeit gesucht haben, bzw. nach denen Sie häufiger	Hat in letzter Zeit nur Englische Begriffe gesucht.

Suchen? Präsentation beispielhafter Filtermöglichkeiten.	
Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	Mehrwert muss erkennbar sein. Wenn irrelevante Ergebnisse zurückgeliefert werden macht es ihr nichts aus, da sie lieber zu viele Ergebnisse hat als zu wenig.
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	Lieber die relevante Ergebnisse anzeigen. Sieht in irrelevanten Ergebnissen auch eine Chance sich weiterzubilden.
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	Lieber zu viele relevante Ergebnisse als zu wenig.
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	Spontan keine Einfälle.
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	Zusatzinformationen zu den Ergebnissen.
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Würde diesen Filter nutzen um Zusatzinformationen zu Begriffen zu Suchen um sich weiterzubilden welche verschiedenen rechtlichen Auslegungen ein Begriff haben könnte.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum: 01.07.2021

Zeit Beginn: 10:00

Zeit Ende: 11:00

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	35, Rechtsanwalt
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	7 Jahre Rechtsanwalt
Wie schätzen Sie Ihre IT Kenntnisse ein?	Sehr gut
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Gesellschaftsrecht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Handbücher, Kommentare, Urteile
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	2 Stunden in der Woche, aber stark fluktuierend Wenn er ein Gutachten erstellen muss, dann 20-30 Stunden pro Woche
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden sie Filter?	Beck Online, Juris, Deutsche Notarinstitut Filter: arbeitet weniger mit Filter sondern legt sich die Schlagwörter so zurecht, dass er die richtigen Inhalte findet Verwendet zwar Beck online aber findet es nicht gut. Interface źu überladen, Interface ist für alle das gleiche (sollte mehr individualisert werden) und es gibt zu viel noise. Das Lesen ist unangenehm, druckt es sich lieber aus zum lesen. Der einzige Grund warum er es verwendet ist wegen dem Content, da Beck online so ziemlich alle Quellen abdeckt.
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	Verfahren: Gerichtsverfahren, Verwaltungsverfahren Berufung: Rechtsmittel Revision: Rechtsmittel 2. Instantz Anlage: Anhang, Gebäude Anlage (spielt bei MA eine Rolle)
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt?	Verfahren: eine Multi Selection wäre Interessant bei den Filtern Wie brauchbar die Filter sind hängt auch davon ab, ob der Jurist eine normative oder faktische Suche anstrebt. Berufung: Senses klar Revision: Der 1. Sense wäre vermutlich in 9 von 10 Suchanfragen relevant, Anlage: Der Ausdruck “Addendum” ist für Juristen evtl. nicht sofort klar

Welche Verbesserung würden Sie sich wünschen?	Verbesserungsvorschläge: Filter nach Autor hilfreich, der Autor impliziert die Bedeutung und das Interesse (Subjektivität) von dem zugrunde liegenden Rechtstext (oder ist zumindest ein starker Indikator). Die Senses sollten ein bisschen Abstrakter (e.g. Berufung (Rechtsmittel)) und konkreter sein, er wünscht sich mehr Aufspaltung
Fallen Ihnen Begriffe ein, nach denen Sie in letzter Zeit gesucht haben, bzw. nach denen Sie häufiger Suchen? Präsentation beispielhafter Filtermöglichkeiten.	Sucht häufig eher nach konkreten Paragraphen
Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	Mehrwert muss erkennbar sein Juristen denken nicht in Wahrscheinlichkeiten, und wenn dann wären sie nur mit 100% Genauigkeit zufrieden Es kommt stark darauf an wie man das Tool den Juristen Präsentiert, e.g. der Filter ist nur eine Art Suchassistent, dann spielt die Fehlerquote eine geringere Rolle
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	Beides wichtig, aber etwas auszuschließen ist im Zweifel leichter, da er oftmals ziemlich genau weiß was er nicht braucht
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	Kommt drauf an wie gut er sich auskennt Wenn er vertraut ist mit Gebiet, dann möchte die er lieber wenige knackige Ergebnisse. In einem neuem Gebiet lieber zu viele Ergebnisse
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	Ist vermutlich vorgekommen, aber kein spezifisches Beispiel im Moment
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	Am liebsten als Netz, Graphennetzwerk angezeigt und visualisiert Wie hängen Begriffe zusammen Mehr als eine Art Spinnennetz mit Knoten, als eine hierarchiesche Struktur Unterschied zu linearen Breadcrumbs, lieber Netz mit allen möglichen Verbindungen
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Priorisierung der Sinnhaftigkeiten wichtig Reihenfolge der Sinnhaftigkeiten wichtig, in welcher Reihenfolge werden sie dem User angezeigt.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum: 29.06.2021

Zeit Beginn: 14:00

Zeit Ende: 15:00

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	30, Rechtsreferendarin
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	5+ Jahre im Rechtsbereich
Wie schätzen Sie Ihre IT Kenntnisse ein?	Sehr gute IT Kenntnisse
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Zivilrecht, viele Rechtsgebiete im Referendariat
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Urteile (am meisten), Kommentare, Aufsätze
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	3 Stunden pro Woche, stark fluktuierend, manchmal Tage wo nichts anderes macht
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden sie Filter?	Beck Online, Juris, Datenbank Bayern Recht Filter nach Datum, Gericht, Hierarchie der Gericht, Dokumenttyp
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	Verfahren: Gerichtsprozess Berufung: 511 ZPO, Berufung in Zivilprozess und Strafprozess als Rechtsmittel Revision: Rechtsmittel nach der Berufung Anlage: Schriftsatz, Emissionsschutzrechtliche Anlage, Gebäudeanlage
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt? Welche Verbesserung würden Sie sich wünschen?	Anmerkung bei Verfahren: Unterschied zwischen 4 und 6 ist nicht ganz klar, könnten synonym zueinander sein 7 ist ein Unterbegriff von 4, d.h. hierarchische Gliederung ist nicht einheitlich Bei Berufung und Revision sind Senses klar. Bei Anlage ist Unterschied zwischen “Addendum” und “Addendum_(Schriftsatz)” nicht klar. Verbesserungsvorschläge: Hierarchien noch feiner gliedern, die Feindifferenzierung ist noch nicht spezifisch genug (Feindifferenzierung indem Senses mehr aufgesplittet werden in spezifischere Senses)
Fallen Ihnen Begriffe ein, nach denen Sie in letzter Zeit gesucht haben, bzw. nach denen Sie häufiger Suchen?	Nicht spontain

Präsentation beispielhafter Filtermöglichkeiten.	
Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	1 von 4 ist für sie eine relativ hohe Fehlerquote es kommt auf die Qualität der übrigen Ergebnisse an, aber würde mindestens eine Genauigkeit von 90% erwarten In der allgemeinen Lebenswelt sind Suchergebnisse immer mehr perfektioniert worden, e.g. Google, daher Leute sind gute Suchmaschinen wie Google gewohnt. Die Jurawelt hängt noch hinterher
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	Filter der relevante Ergebnisse zurückgeliefert wäre ihr lieber
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	Zu wenige Ergebnisse sind gefährlich Aus Haftungsgründen und im Zweifel hätte sie lieber zu viel Ergebnisse
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	Spontan nicht
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	Lieber schon bei Eingabe in den Suchschlitz
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Bei so einer Arbeit ist die konstante Validierung mit Nutzern sehr wichtig, d.h. Austausch und Nutzertests mit Juristen sollten regelmäßig gemacht werden.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum: 05.07.2021

Zeit Beginn: 09:00

Zeit Ende: 09:30

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	33, Rechtsanwalt
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	4 Jahre
Wie schätzen Sie Ihre IT Kenntnisse ein?	Sehr gut
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	IT Recht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Kommentare, Muster
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	30% seiner Zeit, 12 Stunden/ Woche
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden sie Filter?	Beck Online, Juris benutzt sie wie Google, ohne Filter
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	Verfahren: Methodik Berufung: Rechtsmittel Revision: Überprüfung, Rechtsmittel Anlage: Vertragsanlage
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt? Welche Verbesserung würden Sie sich wünschen?	Verfahren: Überschneidungen bei 1 und 3, 4 und 6 sind sehr ähnlich Berufung: Senses klar Revision: Die meisten Senses klar Anlage: Unterschied zwischen 4 und 5 nicht klar Verbesserungsvorschläge: Filter machen nur Sinn wenn sie trennscharf sind Filter sollte automatisch die Suchintention erkennen, möchte keine aktiven Filter verwenden
Fallen Ihnen Begriffe ein, nach denen Sie in letzter Zeit gesucht haben, bzw. nach denen Sie häufiger	Produktbeobachtungspflicht

Suchen? Präsentation beispielhafter Filtermöglichkeiten.	
Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	Mehrwert muss erkennbar sein. Filter muss intuitiv sein.
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	Hätte lieber Filter der nicht relevante Themenblöcke ausschließt
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	Möchte beides, braucht relevante Ergebnisse, aber ihm ist auch vollständigkeit wichtig.
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	Ja oft. Bsp: Datenrecht und Datenschutzrecht.
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	Beides gleich sinnvoll, solange es gut und intuitiv präsentiert ist.
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum: 30.06.2021

Zeit Beginn: 17:00

Zeit Ende: 18:00

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	56, Rechtsanwalt
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	27 Jahre
Wie schätzen Sie Ihre IT Kenntnisse ein?	Anwenderkenntnisse
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Arbeitsrecht, Gesellschaftsrecht, Erbrecht, Berufsrecht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Die ganze Bandbreite
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	3 bis 5 Stunden pro Woche. Aber mit starken Fluktuationen.
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden sie Filter?	Juris und Beck Online verwendet viele verschiedene Filter, je nach Bedarf
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	Verfahren: ZPO, Verfahrensordnung, Vorgehensweise Berufung: ZPO, Erbrecht Revision: Rechtsmittel, Umkehrung eines Zustands Anlage: Parkanlage, Vermögensanlage
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt? Welche Verbesserung würden Sie sich wünschen?	Verfahren: alle Sinnhaftigkeiten für ihn klar, hat erkannt dass Prozess sich auf Streitiges Verfahren bezieht Berufung: alle Bedeutungen klar Revision: kann etwas anfangen mit 1, 3, 4, 5, 6 Bei den Bedeutungen die er nicht kennt würde er sich beispielhaft Ergebnisse anschauen. Er findet es gut dass auch Bedeutungen die er nicht kennt angezeigt werden, da diese ihn auf neue Ideen bringen können. Anlage: Sinnhaftigkeiten klar für ihn, hätte gedacht dass Addendum eine Verallgemeinerung von Addendum_(Schriftwesen) ist. Verbesserungsvorschlag: spontan kein Verbesserungsvorschlag

Fallen Ihnen Begriffe ein, nach denen Sie in letzter Zeit gesucht haben, bzw. nach denen Sie häufiger Suchen? Präsentation beispielhafter Filtermöglichkeiten.	Nur sehr spezielle Begriffe e.g. Treuhandgläubiger, Treuhändiger Kommen nicht in der Sense DB vor
Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	Würde einen Sinnhaftigkeitsfilter verwenden, gerade am Anfang, Einstieg Wenn 1 von 4 Ergebnisse falsch: Würde Sinnhaftigkeitsfilter immer noch verwenden Verwendet immer ein paar verschiedene Möglichkeiten
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	Findet die Idee mit dem Ausschließen sehr gefährlich Für ihn ist die Gefahr, etwas weg filtern was man brauchen würde zu groß Würde lieber mit Filter arbeiten der relevante Ergebnisse anzeigt.
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	Bei zu viel: leidensfähig Bei zu wenig: ärgerlicher Tendenz zu viel als zu wenig Abhängig von wie viel Ergebnisse ihm zurückgeliefert werden
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	Ja, kommt regelmäßig vor Er ist viel im Zivilrecht unterwegs, dort gibt es viele Begriffe die auch im Verwaltungsrecht auftauchen, aber mit einer anderen Bedeutung Doppeldeutigkeiten tauchen immer wieder auf Wörter bei denen ein Sinnhaftigkeit Filter besonders sinnvoll wäre: Hängt ab wie eng man den Begriff suchen Suchverhalten hängt von der Tiefe der Suche ab
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	Schon während der Suchschlitzeingabe
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Würde das ganze am liebsten mit richtigem visuellen Prototyp testen Hätte so einen Filter gerne nicht nur im juristischen Bereich, sondern in allen Bereichen (e.g. bei normalen Google Suchen) Die Feinstruktur der Senses soll so gewählt werden dass es möglichst vollständig ist und alle Senses abgedeckt werden.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum: 29.06.2021

Zeit Beginn: 11:00

Zeit Ende: 11:55

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Nein
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	36, Rechtsanwältin
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	5+ Jahre im Rechhsbereich
Wie schätzen Sie Ihre IT Kenntnisse ein?	Anwenderkenntnisse
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Patentrecht, geschäftliche Rechtsstreitigkeiten
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Normen, Kommentare, Fachartikel (Zeitschriften), Urteile
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	Hohe Fluktuation, 4 Stunden im Schnitt pro Woche
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden sie Filter?	Beck Online, Juris, Wolters Kluwer Verwendet Filter: UND-ODER Verknüpfung und Filterung nach Dokumentenart
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	Verfahren: Gerichtsverfahren, oder ein interner Geschäftlicher Prozess. Abhängig vom Rechtsgebiet (e.g. denkt an patentrechtliche Anmeldeverfahren, wenn sie im Patentrecht unterwegs ist) Berufung: wenn man denselben Prozess eine Instanz höher bringt Revision: wenn man einen Prozess in die Revisionsinstanz bringt, e.g. zum BGH Anlage: Gebäudeanlagen, Geldanlagen
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt? Welche Verbesserung würden Sie sich wünschen?	Bei „Verfahren“: Versteht den Unterschied zwischen „Gerichtsverfahren“ und „Prozess_Recht“ nicht. „Prozess_Recht“ zielt speziell auf Zivilprozesse ab, d.h. sie würde es noch weiter spezifizieren damit man weiß dass es sich um einen Zivilprozess handelt. Wort nicht mit sich selber erklären (e.g. Verfahren nicht als eigenen Sense aufführen) Nicht alle Senses sind rechtlich relevant. Kennt nicht immer alle Senses. Es kommen durchaus Senses vor, an die sie zuvor gar nicht gedacht hat.

	Bei Senses die sie nicht kennt würde sie sich die Filterergebnisse anschauen um eine Vorstellung zu bekommen, was mit diesem Sense gemeint ist. Bei dem Wort „Anlage“ ist der Unterschied zwischen „Addendum_(Schriftwesen)“ und „Addendum“ nicht klar. Manche Senses haben wahrscheinlich gar keine Relevanz für juristische Texte. Verbesserungsvorschläge: Auf den Kontext achten und basierend darauf die Reihenfolge anpassen, juristisch relevantere Bedeutungen als erstes anzeigen. Nur Senses die auch wirklich in der Datenbank vorkommen sollen angezeigt werden.
Fallen Ihnen Begriffe ein, nach denen Sie in letzter Zeit gesucht haben, bzw. nach denen Sie häufiger Suchen? Präsentation beispielhafter Filtermöglichkeiten.	Anfechtung: Im Bürgerlichen Gesetz gibt es eine Anfechtung, und im Insolvenzrecht. Haben miteinander nichts zu tun,
Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	Sie würde den Filter trotzdem noch verwenden, 3 von 4 ist für sie eine gute Quote.
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	Filter zum Anzeigen relevanter Ergebnisse besser, aber beides wäre nützlich. Wenn man Themenblöcke ausschließen könnte, wäre es auch sehr hilfreich.
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	Kommt darauf an, wenn wenig Zeit ist und sie sich einen Überblick verschaffen will hat sie lieber 5 knackige Treffer. Wenn sie Rechtfrage detailiert aufbereiten muss, dann lieber zu viel. Kommt ganz auf die Zielsetzung an.
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	Ja (Anfechtung, siehe oben)
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	Schon bei Suchschlitzeingabe, da ein Arbeitsschritt dadurch gespart wird.
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Priorisierung sehr wichtig, rechtliche Bedeutung sollte als erstes erscheinen / aufgelistet werden
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum: 01.07.2021

Zeit Beginn: 16:15

Zeit Ende: 16:45

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	26, Volljuristin
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	Seit 2014 recherchiert sie regelmäßig
Wie schätzen Sie Ihre IT Kenntnisse ein?	Sehr gut, überdurchschnittlich
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Öffentliches Recht, Verwaltungsrecht, Verfassungsrecht, Datenschutzrecht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Aufsätze, Entscheidungen und Kommentare
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	15 Stunden im Schnitt, fluktuiert stark
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden sie Filter?	Juris und Beck Online verwendet Filter, bei Juris nach Textbaustein, nach Dokumententyp, Zeiträume Bei Beck Online eher weniger, hat das Gefühl die Zuordnung bei den Filtern stimmt nicht ganz
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	Verfahren: Gerichtsverfahren Berufung: Rechtsmittel Revision: Rechtsmittel Anlage: Kapitalanlage
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt? Welche Verbesserung würden Sie sich wünschen?	Verfahren: klar, aber der Unterschied zwischen Gerichtsverfahren und Prozess_(Recht) verschwimmt ein bisschen Berufung: Würde sich eine feinere Aufspaltung von Berufung_(Recht) wünschen Revision: Kennt nicht alle Begriffe, aber würde die die sie nicht kennt im Zweifel mit auswählen und auch nachschauen was sie bedeuten (googeln) Anlage: Alle Begriffe klar Verbesserungsvorschläge: Genauere Aufspaltung der Sinnhaftigkeiten Unterkategorien nach Gesetzen, da Begriffe in den unterschiedlichen Gesetzen unterschiedliche Definitionen haben Erklärungstexte mit Definition von dem Sense
Fallen Ihnen Begriffe ein, nach denen Sie in letzter	Verhältnismäßigkeitsgrundsatz, Formerfordernisse, schriftlich

Zeit gesucht haben, bzw. nach denen Sie häufiger Suchen? Präsentation beispielhafter Filtermöglichkeiten.	sehr spezielle Begriffe, sind nicht in der Sense DB „schriftlich“ ist in der DB, aber nicht fein genug aufgespalten
Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	Wenn sie Suchbegriffe eingibt, dann suchen Fachdatenbanken nur nach Ergebnissen mit ganz genau diesem Wort, das ist nervig Würde den Filter in einem ersten Arbeitsschritt benutzt wenn dieser schon viele irrelevante Ergebnisse herausfiltert
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	Nein, Filter der gezielt die richtigen Ergebnisse zurückliefert wäre besser
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	Lieber zu viel als zu wenig
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	Spontan keine Beispiele, aber gefühlt kommt es bei ihr oft vor
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	Am liebsten während der Suchbegriff Eingabe
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Löst Mehrdeutigkeit momentan mit zusätzlichen Keywords als Workaround Mehr Toleranz für Tippfehler
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum: 02.07.2021

Zeit Beginn: 17:00

Zeit Ende: 18:00

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	45, Rechtsanwalt
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	16 Jahre
Wie schätzen Sie Ihre IT Kenntnisse ein?	Sehr Gut
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Vergaberecht, Einkaufsrecht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Kommentare, Urteile, Aufsätze
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	1 Stunde / Woche, starke Fluktuation
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden sie Filter?	Beck Online normalerweise keine Filter, höchstens nach Kommentare filtern Man muss relativ genau wissen wonach man sucht, d.h. Schlagwörter sind schon so genau dass er nicht mehr filtern muss
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	Verfahren: Gerichtsverfahren Berufung: Rechtsmittel, persönliche Berufung Revision: Revisionsabteilung (Compliance Abteilung), Rechtsmittel Anlage: Vertragsanlage (Addendum), Geldanlage
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt? Welche Verbesserung würden Sie sich wünschen?	Sinnhaftigkeiten sind zu breit gestreut für ihn, da er immer sehr spezifisch sucht. Daher wäre der Filter nur ein begrenzter Mehrwert für ihn. Kann sich unter den meisten Begriffen etwas vorstellen. Der Kosten/Nutzen Aspekt spielt für ihn eine große Rolle, er hätte nicht die Zeit sich vertraut mit Begriffen zu machen die er nicht kennt Verbesserungsvorschläge: Hilft wenn man sich nicht auskennt in einem Gebiete, e.g. als Student Für normale Juristen müsste man eine Ebene tiefer gehen und feiner Aufspaltungen vornehmen Meisten Juristen Risikoscheuer
Fallen Ihnen Begriffe ein, nach denen Sie in letzter Zeit gesucht haben, bzw. nach denen Sie häufiger	Hatte zuletzt nach „Präqualifikation“ gesucht

Suchen? Präsentation beispielhafter Filtermöglichkeiten.	
Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	2 unterschiedliche Suchen für ihn: er hat keine Ahnung oder er weiß genau wo er hin will Wenn er schon weiß wonach er sucht ist er schon auf einer sehr tiefen Detailebene, da wäre ein Filter weniger hilfreich Wenn er sich nicht auskennt, würde eine Aufspaltung helfen. Aber kommt bei ihm nicht oft vor, da er sehr spezialisiert ist.
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	Lieber filtern nach den gesuchten Ergebnissen
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	Bei allgemeiner Suche lieber mehr als wenig Wenn spezifisch nach was sucht, lieber wenige knackige Ergebnisse
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	Es passiert ab und zu, aber spontan fällt ihm nichts ein
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	Lieber schon bei der Eingabe in den Suchschlitz
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Wikipedia ist vermutlich nicht juristisch spezifisch genug, die Datenqualität ist nicht juristisch spezifisch genug. Die Grundidee hätte viel Potential, viele ähnliche Fälle werden in Rechtsabteilungen mehrmals bearbeitet, das könnte man sich sparen. Es tauchen immer wieder die gleichen Fragestellungen auf, hohes Einsparpotential hierbei.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum: 01.07.2021

Zeit Beginn: 13:00

Zeit Ende: 14:00

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	32, Doktorand
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	Schon ca. 9 Jahre mit juristischer Recherche beschäftigt
Wie schätzen Sie Ihre IT Kenntnisse ein?	Sehr gut
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Völkerrecht, Internationales Menschenrecht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Kommentare, Gerichtsurteile
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	30 Stunden pro Woche (viel, da er momentan promoviert)
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden sie Filter?	Beck Online viele verschiedene Datenbank, da er viel im internationalen Recht unterwegs ist: Google Scholars (weniger juristisch), Connected Papers, Max Planck Encyclopedias of International Law Versucht schon im vornherein nach sehr spezifischen Begriffen zu suchen
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	Verfahren: Gerichtsverfahren Berufung: Rechtsmittel Revision: Rechtsmittel, Rechtskraft Anlage: Geldanlage
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt? Welche Verbesserung würden Sie sich wünschen?	Verfahren: Senses sind klar, aber hatte eine andere Auffassung von Prozess (Recht) als Wikipedia Wikipedia ist nicht so genau mit den Definitionen wie das Recht Berufung: Bedeutungen klar Revision: Kennt nicht alle Wörter, wenn er ein Wort nicht kennen würde, dann würde er es googeln. Anlage: Bedeutungen klar Verbesserungsvorschläge:

	kurze Erläuterung zu jedem Sense sollte Funktionieren für detailliertere Suchanfragen Verwandte Begriffe anzeigen Anmerkung: Das sind Begriffe nach denen er normalerweise nicht sucht, die Begriffe sind so allgemein dass man immer spezifischere Begriffe sucht. Er sucht von vornherein schon spezifischere Begriffe die trennschärfer sind.
Fallen Ihnen Begriffe ein, nach denen Sie in letzter Zeit gesucht haben, bzw. nach denen Sie häufiger Suchen? Präsentation beispielhafter Filtermöglichkeiten.	Spontan fällt nichts ein, da hauptsächlich im Völkerrecht im internationalen Recht unterwegs ist (also viel in englischen Rechtsdatenbanken nachschaut)
Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	Mehrwert muss erkennbar sein Würde nicht auf die Prozentwerte / Genauigkeit achten
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	Nein, würde lieber die relevanten Ergebnisse zurückgegeben bekommen
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	Lieber wenige Ergebnisse und dafür aber knackige. Schaut sich normalerweise eh nur die ersten 10 Ergebnisse an bevor er die Suchanfrage anpasst.
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	Ja ständig während seiner Dissertation Er beschäftigt sich mit Menschenrechte in seiner Dissertation und sucht oft nach „Cultural Rights“. „economic, social and cultural rights“ als Phrase kommt aber sehr oft in literatur vor, d.h. werden viele irrelevante Ergebnisse zurückgegeben. Im spezielen „Cultural Rights“ ist mühsam zu suchen Manchmal auch Ergebnisse zu „Agricultural“ zurückgeliefert
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	Beides wäre gut, aber wenn er sich entscheiden müsste wäre ihm ein Dropdown Menü lieber.
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Datengrundlage macht sicher den größten Faktor aus Nicht so sehr auf Performance Score achten sondern auf die Ergebnisse Interface sollte möglichst simpel sein Testen mit unterscheidungskräftigeren Begriffen, niemand sucht nach Wörern wie “Verfahren” und “Berufung” etc…
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum: 06.07.2021

Zeit Beginn: 15:00

Zeit Ende: 15:45

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	Ja
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	31, wissenschaftlicher Mitarbeit
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	2 ½ Jahre
Wie schätzen Sie Ihre IT Kenntnisse ein?	Sehr gut
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Urheberrecht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Fachaufsätze, Kommentare
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	Hat sich viele Suchagenten gebaut 1 Stunden / Woche im Schnitt
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden sie Filter?	Beck Online Filtert hauptsächlich nach Dokumenttyp
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	Verfahren: ZPO, Prozessrecht Berufung: Rechtsmittel, Prozessrecht Revision: Rechtsmittel, Prozessrecht Anlage: Kapitalanlage
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt? Welche Verbesserung würden Sie sich wünschen?	Verfahren: Prozess ist für ihn Überbegriff von Gerichtsverfahren oder Verwaltungsverfahren, Hierarchiebenen einbauen Unterschiede nicht trennscharf, e.g. zwischen 4 und 6 Berufung: Differenzierung klar Revision: Semantische Überschneidungen bei einigen Begriffen Kennt er Begriffe nicht, würde er beispielhaft testen welche Ergebnisse der Filter zurückliefern wüde Anlage: Unterschied zwischen 4 und 5 unklar Wenn der Unterschied unklar ist, würde er beides auswählen

	Verbesserungsvorschläge: Relevanzsortierung der Sinnhaftigkeiten Sensitivitätsfilter, möchte die Feinheit der Sinnhaftikeiten selber wählen können Sensitivitätsfilter über Schieberegler von 0 (stumpf) bis 1(sehr fein strukturiert), so sich die Ergebnisse in „real time“ updaten. Seite soll nicht neu laden.
Fallen Ihnen Begriffe ein, nach denen Sie in letzter Zeit gesucht haben, bzw. nach denen Sie häufiger Suchen? Präsentation beispielhafter Filtermöglichkeiten.	Legal Tech, Inkassodienstleistung, Prozessfinanzierung, Rechtskauf, einzelne Aktenzeichen und Paragraphen
Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	Kann es verschmerzen, solange er zu viele Ergebnisse bekommt als zu wenige Ergebnisse.
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	Lieber irrelevante Themen ausschließen, da er dadurch mehr kontrollieren kann.
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	Lieber zu viele Ergebnisse als zu wenig.
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	Rechtsdienstleisutngsgesetz (RDG) Bekommt Ergebnisse für Rettungsdienstgesetz.
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	Möchte eine Ergebnisliste, Blacklist Möchte Überblick haben über was herausgefiltert wird Möchte sicher sein dass keine relevanten Ergebnisse an ihm vorbei gehen
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Es könnte Leute dazu verleiten, Verantwortung aus gemütlichkeit anzugeben. Wünscht sich Verknüpfung mit Artikel und Paragraphen.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

Interview Leitfaden – Qualitative Evaluation (45+ min)

Leitende Forschungsfrage: Wie schätzen Juristen die präsentierten Filtermöglichkeiten nach Wort Sinnhaftigkeit ein?

Name:

Datum: 30.06.2021

Zeit Beginn: 14:00

Zeit Ende: 15:20

Fragen	Antwort/ Kommentar
GESPRÄCHSEINSTIEG	
Sind Sie damit Einverstanden wenn das Gespräch aufgezeichnet wird (alles anonym und nur für interne Zwecke)?	
Auf Dauer hinweisen	
persönliche Vorstellung + Studium Vorstellung Masterarbeit: • Ziel: Verbesserung juristische Recherche • aktuelle Idee ist automatische Klassifikation von Wörtern nach deren Sinngehalt/Bedeutung bzw. Kontext in dem sie vorkommen (Beispiel Bank, Berufungsgrund). • Aktueller Stand: Qualitative Evaluation der Ergebnisse	
PERSÖNLICHE FRAGEN	
Alter, Beruflicher Werdegang	40, Legal Tech Unternehmerin
Wie viele Jahre arbeiten Sie schon in Ihrem Beruf?	Seit 11 Jahren im juristischen Bereich
Wie schätzen Sie Ihre IT Kenntnisse ein?	Sehr gut
JURISTISCHE RECHERCHE ALLGEMEIN	
Mit welchen Rechtsgebieten beschäftigen Sie sich am meisten?	Gesellschaftsrecht, Stiftungsrecht, Wirtschaftsrecht
Welche Art von Dokumenten schauen Sie sich am meisten an und bei welcher Gelegenheit? E.g. Einstieg in neues Rechtsgebiet über Kommentare?	Komplette Bandbreite
Wie viel Zeit nimmt die Juristische Recherche in Anspruch (pro Woche)?	10 Stunden pro Woche im Durchschnitt
Suchen Sie in juristischen Fachdatenbanken (welche?), und verwenden sie Filter?	RIS, EUR-Lex, RDB, Lexis Nexis, manchmal Google Verwendet Datenbanken zum Nachschlagen, wenn sie weiß wonach sie suchen muss Filter: Quellentypus, welches Gericht, Zeit, Dokumententypus
EVALUATION SENSE FILTER	Davor: Erklärung was ist mit Word Sense gemeint Disambiguieren welche Bedeutung ein Wort in bestimmtem Kontext hat
Wörter können mehrdeutig sein. Wenn sie an das Wort „Verfahren“ denken, welche Bedeutung kommt Ihnen in den Sinn? Selbe mit Berufung, Revision, Anlage.	Verfahren: Gerichtsverfahren, Prozesse Berufung: Rechtsmittel, (persönliche) Erfüllung Revision: Rechtsmittel Anlage: Gebäudeanlage, Anhang
Vorstellung beispielhafter Filtermöglichkeiten bei mehrdeutigen Wörtern (siehe Tabelle unten). Wie würden Sie so einen Filter bewerten? Ist die Kommunikation der Senses gut genug, wissen sie bei den genannten Beispielen um welche Senses es sich konkret handelt? Welche Verbesserung würden Sie sich wünschen?	Verfahren: Sinnhaftigkeiten nicht klar genug und nicht fein strukturiert genug Die Wörter sind nicht spezifisch genug damit eine klare juristische Abgrenzung klar ist, Bedeutungen würden sich überlappen. Wikipedia verwendet die Worte nicht mit der gleichen Präzision wie im Recht üblich. Berufung: Nicht alle Bedeutungsklassen sind sinnvoll. Bedeutungen die Wikipedia unterscheidet sind nicht unbedingt Bedeutungen die im Recht mit der gleichen Feinheit unterschieden werden müssen. Revision: Manche Wörter kennt sie nicht, bei vielen Kategorien wüsste sie nicht was gemeint ist.

	Anlage: Unterschied zwischen “Addendum” und „Addendum_(Schriftwesen)“ nicht klar Bei 1 bis 3 und 6 wüsste sie welche Ergebnisse sie erwarten kann, beim Rest nicht. Verbesserungen: Präzisere Themengebiete Lieber weniger Kategorien anstatt zu viele aber dafür mit einer klaren Abgrenzung. Es muss genug „Distanz“ zwischen den Bedeutungen sein damit damit eine klare Bedeutungsdifferenzierung möglich ist. (E.g. Unterschied zwischen „Bauliche_Anlage“ und „Addendum_(Schriftsatz)“ eindeutig klar, aber z.B. zwischen „Gerichtsverfahren“ und „Prozess_(Recht)“ nicht eindeutig erkennbar, da die damit assoziierten Bedeutungen zu nahe beieinander liegen.) Unterscheidung Österreichisches Recht und Deutsches Recht
Fallen Ihnen Begriffe ein, nach denen Sie in letzter Zeit gesucht haben, bzw. nach denen Sie häufiger Suchen? Präsentation beispielhafter Filtermöglichkeiten.	Stiftungsrecht, Parteirecht (nicht in Sense DB aufgeführt, schon sehr speziell).
Angenommen der Sense Filter hätte eine Genauigkeit von 75%, d.h. in 1 von 4 Fällen ist eine Sinnhaftigkeit falsch zugeordnet, was dazu führen kann dass irrelevante Ergebnisse zurückgeliefert werden, oder relevante Ergebnisse nicht angezeigt werden. Würden Sie den Filter immer noch verwenden? **Falls Nein:** Welche Genauigkeit muss mindestens erreicht werden damit Sie ihn verwenden würden?	Schwierig für sie zum sagen, würde es zuerst testen müssen, Es kommt weniger auf die Prozentwerte an, sondern der konkrete Mehrwert soll spürbar sein. Die Präzision ist wahrscheinlich weniger ausschlaggebend als wie sehr die assoziierten Sinnhaftigkeiten mit den tatsächlichen Sinnhaftigkeiten übereinstimmen, da Sinnhaftigkeit immer subjektiv ist.
Wäre ein Filter der irrelevante Ergebnisse ausschließt, dafür aber mit einer höheren Genauigkeit, Ihnen lieber.	Beides wäre gut, beides hat seinen Sinn
Bei einer Suche, ist es Ihnen lieber wenn sie zu wenig oder zu viel Ergebnisse präsentiert bekommen?	lieber zu viel als zu wenig, lieber mehr false positives
War Mehrdeutigkeit ein Problem für Sie in vergangenen Suchen? Wenn ja, fallen Ihnen spontan Wörter ein wo ein Sinnhaftigkeit Filter besonders sinnvoll wäre?	Spontan fällt ihr kein Beispiel ein. Wenn man spezialisiert ist, dann ergänzt man Suchbegriffe mit einem Keyword wo man aus Erfahrung weiß dass dann die gewünschten Ergebnisse kommen
Wie sieht für Sie die gewünschte Interaktion mit dem Tool aus? (E.g. Hinweise auf Sinn bereits bei Eingabe des Suchbegriffs oder als Zusatzinformationen bei den Ergebnissen?)	Kommunikation der Sinnhaftigkeiten schon während Suchbegriffeingabe
ABSCHLUSS	
Hätten Sie noch einige Hinweise / Anmerkungen / Vorschläge? Gibt es ihrer Meinung nach etwas wichtiges was ich vielleicht noch nicht angesprochen habe?	Starke Unterschiede zwischen österreichischem und deutschem Recht. Z.B. das Wort Stiftung hat ganz andere Bedeutung und rechtliche Implikationen.
Hätten Sie noch andere interessante Kontakte für mich mit denen ich über dieses Thema sprechen kann?	

List of Figures

1.1. Motivational use case. 3
1.2. Design science research framework for information systems. 5

2.1. Illustrations of the CBOW and Skip-gram model. 10
2.2. Architecture of the linear fastText Classifier. 11
2.3. Word vectors of contextual embeddings. 12
2.4. Visualization of a WSD process workflow. 13
2.5. Local graph cluster. 18
2.6. Neural network with three hidden layers. 21

3.1. Four possible WSD use cases. 26
3.2. Example of a German legal information retrieval task. 27
3.3. Use case diagram for the interaction with a minimalistic legal database that uses word sense filters. 28
3.4. Court type distribution. 30
3.5. Year distribution. 30
3.6. Distribution of legal decision types. 31
3.7. Sense distribution in the legal corpus. 32
3.8. Token correlation. 33
3.9. Sense distribution of the document sampled test set. 35
3.10. Sense distribution test set. 37
3.11. Court type distribution test set. 38
3.12. Label distribution of the manually tagged test set. 39

4.1. Graphical overview of the training data generation process to train a sense classifier. 42
4.2. Creating a manually sense annotated legal corpus. 43
4.3. Process of fine-tuning sense classifier and creating a sense annotated legal corpus. 44
4.4. Working mechanism of fastSense. 45
4.5. UML class diagram of the preprocessor and parser. 48
4.6. Entity-relationship diagram of the sense database. 49
4.7. Flowchart of the data creation process. 50
4.8. The hashing model architecture. 52
4.9. The fastBERT model architecture. 53

5.1. Micro and macro accuracies of the hashing model for different n-gram size variations. 57

5.2. Micro and macro accuracies of the hashing model when lemmatizing and not lemmatizing word tokens. 58
5.3. Micro and macro accuracies of the hashing model when using either sentences or whole paragraphs. 59
5.4. Micro and macro accuracies of the hashing model when using either sentences or whole paragraphs together with POS information for each token. . 60
5.5. Micro and macro accuracies of the hashing model when varying the dropout probability. 61
5.6. Model's performance for different variations of the default settings. 62
5.7. Performance evaluation of the fastBERT model. 65
5.8. Performance evaluation of the fastBERT model on the manually tagged legal test set. 66

6.1. Age distribution of the people interviewed for the qualitative evaluation. . 68
6.2. Filter usage behaviour among the interviewed legal experts. 69
6.3. IT skills of the interviewed legal experts when compared to the average lawyer. 69
6.4. Distribution of the first associated meaning for each of the ambiguous words. 70

List of Tables

3.1. Token statistics of the text segments in the legal dataset. 29
3.2. Overview over the resources that are used for this thesis. 40

4.1. Description of the required Wikipedia dumps. 47
4.2. Structure of the TFRs that are produced by the exporter. 49
4.3. Train/validation/test split ratio of the generated data for the hashing model. 50
4.4. Train/test split ratio of the generated data for the fastBERT model. 54

5.1. Parameter study of the hashing model. 63
5.2. Parameter study of the fastBERT model on the generated test set. 64
5.3. Parameter study of the fastBERT model on the manually tagged legal test set. 64

Bibliography

Eneko Agirre and Philip Edmonds. *Word Sense Disambiguation: Algorithms and Applications*, chapter 1, pages 11–12. Springer Publishing Company, Incorporated, 1st edition, 2007. ISBN 1402068700.

Ahmed H. Aliwy and Hawraa A. Taher. Word Sense Disambiguation: Survey Study. *Journal of Computer Science*, 15(7):1004–1011, Jul. 2019. doi: 10.3844/jcssp.2019.1004.1011. URL `https://thescipub.com/abstract/jcssp.2019.1004.1011`.

Satanjeev Banerjee and Ted Pedersen. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. volume 2276, pages 136–145, 02 2002. ISBN 978-3-540-43219-7. doi: 10.1007/3-540-45715-1_11.

Yehoshua Bar-Hillel. The Present Status of Automatic Translation of Languages. volume 1 of *Advances in Computers*, pages 91–163. Elsevier, 1960. doi: https://doi.org/10.1016/S0065-2458(08)60607-5. URL `https://www.sciencedirect.com/science/article/pii/S0065245808606075`.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. volume 1, pages 238–247, 06 2014. doi: 10.3115/v1/P14-1023.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model. 08 2014.

Mohamed Ben Aouicha, Mohamed Ali Hadj Taieb, and Hania Marai. *WordNet and Wiktionary-Based Approach for Word Sense Disambiguation*, pages 123–143. 01 2018. ISBN 978-3-319-90286-9. doi: 10.1007/978-3-319-90287-6_7.

Trevor Bench-Capon, Michał Araszkiewicz, Kevin Ashley, Katie Atkinson, Floris Bex, Filipe Borges, Daniele Bourcier, Paul Bourgine, Jack G. Conrad, Enrico Francesconi, Thomas F. Gordon, Guido Governatori, Jochen L. Leidner, David D. Lewis, Ronald P. Loui, L. Thorne Mccarty, Henry Prakken, Frank Schilder, Erich Schweighofer, Paul Thompson, Alex Tyrrell, Bart Verheij, Douglas N. Walton, and Adam Z. Wyner. A History of AI and Law in 50 Papers: 25 Years of the International Conference on AI and Law. *Artif. Intell. Law*, 20(3):215–319, September 2012. ISSN 0924-8463. doi: 10.1007/s10506-012-9131-x. URL `https://doi.org/10.1007/s10506-012-9131-x`.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2017.

Pranjal Borah, Gitimoni Talukdar, and Arup Boruah. Approaches for Word Sense Disam-

biguation - A Survey. *International Journal of Recent Technology and Engineering*, 3:135–138, 03 2014.

Facundo Bre, Juan Gimenez, and Víctor Fachinotti. Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks. *Energy and Buildings*, 158, 11 2017. doi: 10.1016/j.enbuild.2017.11.045.

Samuel Broscheit, A. Frank, D. Jehle, Simone Paolo Ponzetto, Danny Rehl, A. Summa, Klaus Suttner, and Saskia Vola. Rapid Bootstrapping of Word Sense Disambiguation Resources for German. In *KONVENS*, 2010.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Salvatore Caserta and Mikael Rask Madsen. The Legal Profession in the Era of Digital Capitalism: Disruption or New Dawn? *Laws*, 8(1), 2019. ISSN 2075-471X. doi: 10.3390/laws8010001. URL `https://www.mdpi.com/2075-471X/8/1/1`.

Arindam Chatterjee. A Survey of Sense Annotation, Eye-tracking and Wordnet Linking approaches till June, 2012. 2012.

Ping Chen, Wei Ding, Chris Bowes, and David Brown. *A Fully Unsupervised Word Sense Disambiguation Method Using Dependency Knowledge.* January 2009. doi: 10.3115/1620754.1620759.

Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104. URL `https://doi.org/10.1177/001316446002000104`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Beate Dorow and Dominic Widdows. Discovering Corpus-Specific Word Senses. *EACL '03: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics*, 04 2003. doi: 10.3115/1067737.1067753.

Gerard Escudero. Machine Learning Techniques for Word Sense Disambiguation. 01 2006.

J. R. Firth. A synopsis of linguistic theory 1930-55. 1952-59:1–32, 1957.

Yoav Freund and Robert E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1997.1504. URL `https://www.sciencedirect.com/science/article/pii/S002200009791504X`.

William Gale, Kenneth Ward Church, and David Yarowsky. Estimating Upper and Lower Bounds on the Performance of Word-Sense Disambiguation Programs. In *Proceedings of the 30th Annual Meeting on Association for Computational Linguistics*, ACL '92, page 249–256, USA, 1992. Association for Computational Linguistics. doi: 10.3115/981967.981999. URL https://doi.org/10.3115/981967.981999.

Reinhard Geissbauer, Stefan Schrauf, Philipp Berttram, and Farboud Cheraghi. Digital Factories 2020: Shaping the future of manufacturing, April 2017. URL https://www.pwc.de/de/digitale-transformation/digital-factories-2020-shaping-the-future-of-manufacturing.pdf.

Anton Geist. *Rechtsdatenbanken und Relevanzsortierung*. PhD thesis, Universität Wien, 2016.

Julio Gonzalo, M. Verdejo, Irina Chugur, and Juan Manuel Cigarran Recuero. Indexing with WordNet synsets can improve Text Retrieval. *CoRR*, 09 1998.

Birgit Hamp and Helmut Feldweg. GermaNet - a Lexical-Semantic Net for German. In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, 1997. URL https://www.aclweb.org/anthology/W97-0802.

Zellig Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954. doi: 10.1007/978-94-009-8467-7_1. URL https://link.springer.com/chapter/10.1007/978-94-009-8467-7_1.

Wahed Hemati, Tolga Uslu, and Alexander Mehler. TextImager: a Distributed UIMA-based System for NLP. 01 2016.

Verena Henrich. *Word Sense Disambiguation with GermaNet*. PhD thesis, 01 2015.

Verena Henrich and Erhard Hinrichs. A Comparative Evaluation of Word Sense Disambiguation Algorithms for German. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 576–583, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/164_Paper.pdf.

Verena Henrich, Erhard Hinrichs, and Tatiana Vodolazova. Semi-Automatic Extension of GermaNet with Sense Definitions from Wiktionary. In *Proceedings of 5th Language & Technology Conference (LTC 2011), Poznań, Poland, November 2011, pp. 126-130.*, 2011.

Verena Henrich, Erhard Hinrichs, and Tatiana Vodolazova. WebCAGe: a web-harvested corpus annotated with GermaNet senses. pages 387–396, 04 2012.

Alan Hevner, Salvatore T. March, and Jinsoo Park. Design Science in Information Systems Research. *Management Information Systems Quarterly*, 28:75–, 03 2004.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification, 2016.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. Zero-shot Word Sense Disambiguation using Sense Definition Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, Florence, Italy,

July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1568. URL https://www.aclweb.org/anthology/P19-1568.

Joerg Landthaler, Bernhard Waltl, Dominik Huth, Daniel Braun, Florian Matthes, Christoph Stocker, and Thomas Geiger. Extending Thesauri Using Word Embeddings and the Intersection Method. 06 2017.

Stephen A. Lastres. Rebooting Legal Researchin a Digital Age. The Research Intelligence Group (TRiG), 2012. URL http://www.lexisnexis.com/documents/pdf/20130806061418_large.pdf.

Yoong Keok Lee and Hwee Tou Ng. An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, page 41–48, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118693.1118699. URL https://doi.org/10.3115/1118693.1118699.

Michael Lesk. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, page 24–26, New York, NY, USA, 1986. Association for Computing Machinery. ISBN 0897912241. doi: 10.1145/318723.318728. URL https://doi.org/10.1145/318723.318728.

Omer Levy, Yoav Goldberg, and Ido Dagan. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015. doi: 10.1162/tacl_a_00134. URL https://www.aclweb.org/anthology/Q15-1016.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic Knowledge and Transferability of Contextual Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1112. URL https://www.aclweb.org/anthology/N19-1112.

Qi Liu, Matt J. Kusner, and Phil Blunsom. A survey on contextual embeddings, 2020.

Virpi Lyytikäinen, Pasi Tiitinen, and Airi Salminen. Challenges for European legal information retrieval. 01 2000.

Tamara Martín and Rafael Berlanga-Llavori. A clustering-based Approach for Unsupervised Word Sense Disambiguation. *Procesamiento del Lenguaje Natural*, 49:49–56, 09 2012.

Michael Matuschek. *Word Sense Alignment of Lexical Resources*. PhD thesis, Technische Universität, Darmstadt, 2015. URL http://tuprints.ulb.tu-darmstadt.de/4355/.

Rada Mihalcea. Unsupervised Large-Vocabulary Word Sense Disambiguation with Graph-Based Algorithms for Sequence Data Labeling. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, page 411–418, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1220575.1220627. URL https://doi.org/10.3115/1220575.1220627.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, 2013.

George Miller, Claudia Leacock, Randee Tengi, and Ross Bunker. A semantic concordance. pages 303–308, 01 1993. doi: 10.3115/1075671.1075742.

George A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, November 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748.

George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991. doi: 10.1080/01690969108406936. URL https://doi.org/10.1080/01690969108406936.

Roberto Navigli. Word Sense Disambiguation: A Survey. *ACM Comput. Surv.*, 41, 02 2009. doi: 10.1145/1459352.1459355.

Roberto Navigli and Simone Ponzetto. BabelNet: Building a Very Large Multilingual Semantic Network. pages 216–225, 09 2010.

Marc Opijnen and Cristiana Santos. On the concept of relevance in legal information retrieval. *Artificial Intelligence and Law*, 25:65–87, 03 2017. doi: 10.1007/s10506-017-9195-8.

Simone Papandrea, Alessandro Raganato, and Claudio Delli Bovi. SupWSD: A Flexible Toolkit for Supervised Word Sense Disambiguation. 09 2017. doi: 10.18653/v1/D17-2018.

Ted Pedersen. A Decision Tree of Bigrams is an Accurate Predictor of Word Sense. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001. URL https://www.aclweb.org/anthology/N01-1011.

Judita Preiss, Jon Dehdari, Josh King, and Dennis Mehay. Refining the most frequent sense baseline. 06 2009. doi: 10.3115/1621969.1621973.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

Alok Ranjan Pal and Diganta Saha. Word Sense Disambiguation: A Survey. *International Journal of Control Theory and Computer Modeling*, 5(3):1–16, Jul 2015. ISSN 2249-1155. doi: 10.5121/ijctcm.2015.5301. URL http://dx.doi.org/10.5121/ijctcm.2015.5301.

Ronald L. Rivest. Learning Decision Lists. *Mach. Learn.*, 2(3):229–246, November 1987. ISSN 0885-6125. doi: 10.1023/A:1022607331053. URL https://doi.org/10.1023/A:1022607331053.

Jahn-Takeshi Saito, Joachim Wagner, Graham Katz, Philip Reuter, Michael Burke, and Sabine Reinhard. Evaluation of GermaNet: Problem Using GermaNet for Automatic Word Sense Disambiguation. pages 14–29, 01 2002.

Mark Sanderson. Word Sense Disambiguation and Information Retrieval. 09 1996. doi: 10.1007/978-1-4471-2099-5_15.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation. volume 34, 04 2020. doi: 10.1609/aaai.v34i05.6402.

Clemens Schulz. Automatische disambiguation für das englische, 2021.

Hinrich Schütze. Automatic Word Sense Discrimination. *Comput. Linguist.*, 24(1):97–123, March 1998. ISSN 0891-2017.

Christopher Stokoe, Michael Oakes, and John Tait. Word Sense Disambiguation in Information Retrieval Revisited. pages 159–166, 01 2003. doi: 10.1145/860435.860466.

Chi Sun, Xipeng Qiu, Yige Xu, and X. Huang. How to Fine-Tune BERT for Text Classification? *ArXiv*, abs/1905.05583, 2019.

Heike Telljohann, Erhard Hinrichs, and Ra Ubler. The Tüba-D/Z Treebank: Annotating German with a Context-Free Backbone. *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, 07 2004.

A. M. Turing. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236): 433–460, 10 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL `https://doi.org/10.1093/mind/LIX.236.433`.

Tolga Uslu, Alexander Mehler, Daniel Baumartz, and Wahed Hemati. FastSense: An Efficient Word Sense Disambiguation Classifier. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL `https://www.aclweb.org/anthology/L18-1168`.

Özlem Uzuner, Boris Katz, and Deniz Yuret. Word Sense Disambiguation for Information Retrieval. page 985, 01 1999.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

Stephen F. Weiss. Learning to disambiguate. *Information Storage and Retrieval*, 9 (1):33–41, 1973. ISSN 0020-0271. doi: https://doi.org/10.1016/0020-0271(73)90005-3. URL `https://www.sciencedirect.com/science/article/pii/0020027173900053`.

David Yarowsky. One Sense per Collocation. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, page 266–271, USA, 1993. Association for Computational Linguistics. ISBN 1558603247. doi: 10.3115/1075671.1075731. URL `https://doi.org/10.3115/1075671.1075731`.

David Yarowsky and Radu Florian. Evaluating Sense Disambiguation across Diverse Parameter Spaces. *Nat. Lang. Eng.*, 8(4):293–310, December 2002. ISSN 1351-3249. doi: 10.1017/S135132490200298X. URL `https://doi.org/10.1017/S135132490200298X`.

Zhi Zhong and Hwee Tou Ng. Word Sense Disambiguation Improves Information Retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Lin-*

guistics (Volume 1: Long Papers), pages 273–282, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P12-1029`.

Jiehan Zhu. SQuAD Model Comparison. 2019.