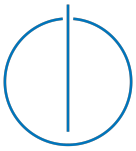
SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY – INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Information Systems

Leveraging Domain Knowledge for Class-Specific Keyword Extraction

Weixin Yan



SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY – INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Information Systems

Leveraging Domain Knowledge for Class-Specific Keyword Extraction

Nutzung von Domänenwissen für die Extraktion Klassenspezifischer Schlüsselwörter

Author:	Weixin Yan
Supervisor:	Prof. Dr. Florian Matthes
Advisors:	Stephen Meisenbacher, M.Sc.
	Tim Schopf, M.Sc.
Submission Date:	15.09.2023

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, 15.09.2023 Weixin Yan

Acknowledgments

I've been pondering over this acknowledgement section since day one of my thesis journey. In fact, I've probably spent more time thinking about this than any sane person should. And now, as the very, very, very last piece of this puzzle, I'm finally putting my gratitude into words.

Firstly, I'd like to extend my gratitude to Prof. Dr. Florian Matthes. You might not have been fully aware, but you've inadvertently contributed some valuable ideas for this work. Thank you for giving me the opportunity to undertake this thesis under your supervision.

Next up, a MASSIVE shout-out to Stephen and Tim, my dynamic duo of advisors. Our weekly meetings, filled with brainstorming, laughter, and the occasional (okay, often) overshooting of scheduled time (my apologies!), have been the highlights of this journey. Your support has been monumental, especially during times when I felt I hadn't done enough. Tim, your sense of humor and insightful ideas have added so much value to our discussions, and those occasional roasts were just the motivation I needed. And Stephen, your dedication speaks volumes. The absence of pressure, the constant encouragement, and setting up meetings even during your vacation were deeply appreciated. The fact that you consistently provide rapid and invaluable feedback, and somehow manage to remember pretty much every detail despite juggling countless tasks, is genuinely astounding (seriously, HOW??). To both of you, I am beyond grateful and honored to have had the privilege of working with you. Thanks a ton!

To the domain experts who participated in the validation process, thank you for finding time in your busy schedules and providing invaluable feedback. Your expertise made this thesis stronger. A special mention to all who took part in the survey; your feedback was pivotal in assisting our evaluation. Thank you!

Heartfelt thanks to my friends. Tony, the ever-present troubleshooter. You've witnessed the entire roller-coaster, from the thesis hunt to the final submission. I owe you countless coffees. Sarah, my emotional anchor, unofficial cheerleader, and midnight proofreader (there's your Oxford comma). Thank you for offering food and shelter in case of a meltdown (thankfully, that never happened). You always say you haven't done much, but in the grand scheme of things, you've done A TON. To Luna, my partner in crime since our elementary school days. Even though your main contribution was laughing at my struggles, having you around all these years has been a blessing in disguise. And to those who've constantly checked on my progress, teased me with the

ever-present question of "Why aren't you done yet?", and kept me in their thoughts and prayers, I cherish each one of you.

Now, to my parents. Words might fall short, but here's an attempt. You are the bedrock upon which my life stands. Your love, your support, your unwavering trust in me, and most importantly, introducing me to the faith that has been my guiding light, are the reasons I am here today. As I reflect on the past 22 years of my life, I am overwhelmed with the realization of how truly blessed I am, and indeed, all things work together for good.

As I draw this chapter of my academic journey to a close, I'm reminded of words that have been a beacon of hope throughout. Perhaps they encapsulate the essence of this entire endeavor better than I ever could:

> *For in hope we have been saved, but hope that is seen is not hope; for who hopes for what he already sees? But if we hope for what we do not see, with perseverance we wait eagerly for it.*
>
> —- Romans 8:24-25

In the spirit of hope and perseverance, I've reached the end of this journey. To everyone who's been a part of this adventure, from the bottom of my caffeinated heart, thank you. Here's to many more sleep-filled nights and a thesis that's finally complete!

Abstract

In the current data landscape, a significant portion of information remains unstructured, posing challenges to its effective utilization. The first step towards harnessing the potential of this data is to extract meaningful and domain-relevant keywords and keyphrases, which is crucial for deriving pertinent insights. While the field of keyword extraction has been extensively studied, the specific angle of integrating domain knowledge into the keyword extraction process remains less charted. Addressing this gap, this thesis introduces a systematic approach that seamlessly integrates domain expertise into a comprehensive keyword identification pipeline.

The methodology is characterized by a two-fold process: extraction and generation. Initially, keywords are extracted based on their relevance to classes within particular domains, with the process being guided using class-specific knowledge validated by domain experts. The subsequent generation process is divided into three stages: lexical substitution, synonym generation and word form generation. Each stage aims to introduce new keywords, thereby enriching the overall keyword set.

Our proposed approach demonstrates notable efficiency, significantly accelerating a manual domain-specific keyword extraction process and outperforming conventional algorithms in terms of class-specificity. Furthermore, the pipeline offers adaptability across various domains and is designed with high configurability to diverse needs. In validation of our methodology, both expert-led and automatic evaluations were conducted to assess the accuracy and relevance of the resulting keywords. The results, while already promising, highlight the methodology's amplified potential with further refinements.

Contents

1. Introduction

1.1. Motivation

The current data landscape is characterized by an overwhelming predominance of unstructured text. Unstructured data, due to its inherent irregularities and ambiguities, presents significant challenges for analysis and interpretation, particularly for AI applications that typically require structured input data with clear labels.

Data annotation becomes the obvious solution to this problem. The traditional approach to annotation involves manual classification and annotation of text by domain experts. While this method yields precise results and incorporates domain-specific knowledge, it is highly inefficient, costly, and not scalable. This motivates the need for a hybrid approach that combines Natural Language Processing (NLP) techniques with domain expertise for more efficient annotation and classification of text data.

1.1.1. Project Overview

The CreateData4AI (CD4AI) project[1] was devised to address the aforementioned challenges. It employs a pipeline that utilizes state-of-the-art NLP techniques to aid domain experts in annotating text data. This framework incorporates domain-specific knowledge directly into the data set creation process. Starting with an unstructured text corpus, the goal is to create structured and annotated data sets that are classified according to defined features. The pipeline consists of several sub-tasks, including:

1. **Keyword Extraction:** Keywords and keyphrases are extracted using unsupervised techniques based on classes or labels defined by a domain expert. Additionally, associated terms and expressions are proposed to refine the scope of the classes.

2. **Context Window Extraction:** Windows encapsulating the context are extracted surrounding the identified keywords and keyphrases. These context windows aim to capture the meaning of the keyword, providing insight into its role within the text.

3. **Context Rule Creation:** With the acquired context windows, the domain expert assesses and identifies which windows aptly capture the essence of the pre-defined classes. These chosen rules set the foundation for automated data creation.

4. **Extrapolation:** For each class, the previously created context rules are combined with NLP techniques to annotate the remainder of the unstructured text corpus.

[1] https://wwwmatthes.in.tum.de/pages/nqpi6qljq0x9/CreateData4AI-CD4AI

This extends the established rules to cover potentially infinite unseen content, bridging the gap between manual rule-setting and unsupervised categorization.

1.2. Thesis Focus: Keyword Extraction

This thesis forms a part of the CD4AI project and aims to advance its overarching objectives by focusing on the initial step in this pipeline - keyword extraction.

A keyword is a concise textual representation that captures the fundamental essence information of a longer document. The term "keyword" is generally attributed to a single word, whereas a "keyphrase" denotes a sequence of multiple words. For the scope of this thesis, the terms "keywords" and "keyphrases" are used interchangeably, both signifying textual elements regardless of their word count.

This first step in the project is to support the domain expert in defining the classes with the help of keyword extraction techniques. In this context, the role of a domain expert involves conceptualizing the desired classes by assigning relevant tags or creating class descriptions. This domain-specific knowledge can then be injected into state-of-the-art keyword extraction methods, offering support for the domain expert to better identify related class-specific keywords and potentially refine the scope of the class.

While many conventional keyword extraction methodologies might take into account the context in which a word appears, they often do not consider domain-specific knowledge. Our approach is designed to fill this gap. Rather than just identifying words based on their prominence in a text, we aim to extract keywords that are both contextually relevant and aligned with the intricacies of a specific domain or class. This dual emphasis on class-specificity and context ensures that our extracted keywords truly capture the core essence of the domain.

The desired outcome of this research is to craft a comprehensive set of keywords for each pre-defined class. These keywords, enhanced in relevance and precision, serve as the basis for subsequent project phases, ensuring heightened efficiency and accuracy tailored to the specific requirements of the domain expert.

1.3. Research Questions

This thesis investigates the following research questions:

1. What approaches currently exist that can be utilized to extract keywords from large unstructured text corpora?

2. How can short textual class descriptions and class-specific seed keywords from the Klassifikation der Wirtschaftszweige, Ausgabe 2008 (WZ2008) classification,[2] validated by domain experts, be leveraged to adapt the identified keyword extraction approaches for class-specific keyword extraction?

[2]https://www.destatis.de/DE/Methoden/Klassifikationen/Gueter-Wirtschaftsklassifikationen/klassifikation-wz-2008.html

3. Without the use of external knowledge bases, how can the extracted class-specific keywords be used as a basis for the generation of further class-specific keywords?

4. In what way can the modified approach be evaluated by domain experts to validate the representativeness of the resulting keyword sets?

By addressing these questions, the thesis aims to contribute to the overarching goal of the CD4AI project: supporting domain experts in the definition of classes, particularly in the creation of keywords and keyphrases, for characterizing large text corpora.

2. Foundations

Before delving into the methodology for our keyword extraction and generation pipeline, it is crucial to first lay down the foundational principles and techniques that underpin these processes. This chapter provides an overview of varied extraction methodologies, text data representation and transformer models, similarity calculation, and morphological analysis. Additionally, we will explore the role of lexical tools in keyword generation, setting the stage for the discussions that follow.

2.1. Keyword Extraction

Keyword and keyphrase extraction in NLP is often referred to as the automatic identification of words and phrases that best encapsulate the core information of a text [24]. In essence, high-quality keywords elucidate the content, simplify comprehension, convey ideas, and evaluate the text's significance. Consequently, they are extensively employed in NLP tasks including but not limited to text summarization [8], topic modeling [68], and information retrieval.

This section will briefly introduce the different types of keyword extraction based on the availability of labeled training data as well as the common metrics used for extraction. For a thorough review of these keyword and keyphrase extraction methods, the reader is referred to [56].

2.1.1. Types of Keyword Extraction

Keyword extraction methodologies can be broadly categorized into three primary types based on the availability and utilization of labeled data: supervised, unsupervised, and semi-supervised extraction.

Supervised keyword extraction is heavily dependent on labeled data. In traditional supervised techniques, keyword and keyphrase extraction is viewed as a binary classification task. Here, annotated documents with labeled keyphrases train a classifier to determine whether a candidate phrase qualifies as a keyphrase or not [24]. Witten et al.'s KEA [78] is one of the pioneering systems to approach supervised keyphrase extraction. Yet, these classification methods often struggle to rank candidate keyphrases by their relative significance [29]. To address this, ranking-based supervised methods were introduced and demonstrated superior performance compared to the binary classification approach [31]. An example is the Ranking SVM [31] that employs a linear model to rank candidate phrases and then selects the top-ranked phrases as keyphrases.

Although supervised approaches tend to yield a higher accuracy due to their adaptability to the specific syntax, semantics and content of documents [43], they require a large annotated corpus for the training, a resource that is frequently unavailable. Moreover, these methods do not perform well when encountering domains not represented in the training corpus [7].

In contrast, **unsupervised keyword extraction** methods provide an alternative approach that obviates the need for labeled data. Instead of relying on annotations, these methods approach keyword extraction as a ranking problem, employing various heuristics to assign scores to candidate words or phrases. Widely adopted techniques encompass graph-based ranking methods like TextRank [50], clustering-based methods [40], and approaches grounded in language modeling [70]. Unsupervised methods are also language and content independent [43], permitting applicability across varied domains.

While unsupervised methods offer various advantages, they are not without limitations. Firstly, unlike their supervised counterparts, these techniques cannot be trained and fine-tuned using specific data sets, potentially compromising their accuracy [43]. Furthermore, many unsupervised methods require the input document to be embedded within a larger corpus, also provided as input, to facilitate effective keyword extraction [7]. Notably, innovations like EmbedRank [7] have addressed this constraint by leveraging sentence embeddings, allowing keyphrases to be extracted without requiring the document to be part of a larger corpus.

With the distinct advantages and disadvantages of both supervised and unsupervised techniques in view, researchers have shifted towards **semi-supervised keyword extraction**. By integrating both labeled and unlabeled data, this approach seeks to maximize the strengths in both paradigms. For instance, Karnalim's software keyphrase extractor [34] combines both supervised and unsupervised techniques, harnessing the strengths of both approaches. Another graph-based method, proprosed by Aggarwal et al., yields a higher accuracy than existing supervised algorithms and offers adaptability across domains [1]. Furthermore, Jonathan and Karnalim's research introduced a semi-supervised method centered on fact-based sentiment, proving its efficacy, especially for scientific articles [32].

2.1.2. Common Techniques and Metrics

This section aims to categorize and elucidate the primary techniques employed for keyword extraction, categorized according to the following principles: statistics-based, graph-based, and embedding-based keyword extraction methods.

Statistics-based approaches score terms in a document using various statistical measures, subsequently extracting the top n terms with the highest scores as the keywords. One widely-accepted baseline in this category is the TF-IDF, which is the product of TF (term frquency) and IDF (inverse document frequency). Here, the TF represents the raw frequency of a term within a document, while the IDF quantifies the term's importance across a collection of documents.

While TF-IDF is fundamental, many statistics-based keyword extractors combine it

with other metrics. For instance, KP-Miner [6] utilizes it with other factors like a cut-off constant. Additionally, recent research has seen TF-IDF integrated with semantic classification [76] and neural taggers [35] for enhanced keyword extraction.

Other statistical methods have begun to incorporate context information. YAKE [12], for instance, employs five distinct features to determine a term's contextual relevance and its distribution across various sentences.

Statistics-based methods are valued for their computational simplicity and universal applicability regardless of domain and language boundaries. However, a notable limitation is their inability to deeply comprehend keyword meanings and hence the meaning of the document. Moreover, these methods often struggle with polysemy, as they are not able to distinguish between different meanings of the same word. Even advanced models like YAKE, which account for context, remain constrained in textual understanding, since their basis for context incorporation remains statistical, lacking an in-depth semantic comprehension.

The essence of **graph-based** keyword extraction approaches lies in transforming a document into a graph. In this representation, each candidate word becomes a node, and any pair of candidate words co-occurring within a certain window length are connected by an edge. The nodes are subsequently ranked using algorithms such as Google's PageRank [10], and the top n nodes, based on the ranking scores, are selected as keywords for the document.

TextRank [50] was the first keyphrase extraction algorithm based on PageRank, which creates an undirected and unweighted graph from a document. SingleRank [73] enhanced TextRank by introducing weighted edges between words co-occurring within a window of size $w \geq 2$. Rose et al. introduced RAKE (Rapid Automatic Keyword Extraction) [64], which builds on word co-occurrences to create a graph, and subsequently scores words using word frequency and word degree. It has been shown to be more computationally efficient and precise than TextRank [64].

More recent algorithms have combined graph-based metrics with other measures to improve keyword extraction performance. For instance, SGRank [15] fuses statistical and graphical metrics, while PositionRank [18] incorporates all positions of a word's occurrences into a position-biased PageRank model, predicated on the assumption that words appearing earlier in a document carry more significance.

One of the strengths of graph-based methods is their robustness to function without requiring linguistic processing [5]. This adaptability extends further, as graph-based techniques are designed to be domain and language independent. Although extracted keywords by graphical methods can be more coherent than that of statistical methods, they are more computationally intensive due to the overhead of graph generation [71].

However, much like their statistics-based peers, traditional graph-based methods encounter challenges in capturing semantic depth. Built to identify structural and co-occurrence patterns, they tend to overlook the underlying meanings of words. Recognizing this limitation, researchers have turned to word embeddings, which have demonstrated the capacity to encode both syntactic and semantic features of words [51]. This motivated the integration of pre-trained, domain-unspecific word embeddings as an

external knowledge base into graph-based ranking and extraction models [74, 75].

Inspired by the potential of embeddings, **embedding-based keyword extraction approaches** such as Key2Vec [42] emerged. Key2Vec employs domain-specific word embeddings to rank extracted keyphrases from scientific articles. Progressing further in this direction, EmbedRank [7] utilized sentence embeddings, specifically Doc2Vec [36] and Sent2vec [55], to represent the document and the candidate phrases in the same high-dimensional vector space, subsequently ranking the candidate phrases. This methodology performs better than traditional graph-based approaches such as TextRank [50] and SingleRank [73], and even outperforms complex graph-based methods incorporating word embeddings such as those proposed by [75]. Moreover, the Reference Vector Algorithm (RVA) [57] makes use of local word embeddings—particularly GloVe vectors [58]—to represent candidate words and phrases. Local embeddings are those trained on the single document in question, differing from global embeddings that rely on pre-training with large external text corpora. Such local representations have been observed to extract better keyphrases than those obtained from global representations or other unsupervised keyphrase extraction methods [57].

In comparison to statistics-based and graph-based approaches, embedding-based models have a heightened ability for a deeper understanding of textual data, enriching keyword extraction results by leveraging both syntactic and semantic nuances [51]. While they offer numerous benefits in terms of semantic comprehension, the reliance on embeddings could introduce complexities. Additionally, the quality of the embedding models and their alignment with the specific domain in question becomes crucial for achieving optimal results [75].

2.2. Transformer Models

Traditional approaches for sequence modeling predominantly relied on recurrent neural networks (RNNs), including architectures like Long Short-Term Memory (LSTM) [27]. However, RNN-based models come with intrinsic limitations: their training is time-consuming due to their sequential text processing nature, inhibiting the parallel processing capabilities of modern GPUs [72]. This sequential nature also poses challenges in managing longer sequences and capturing long-range dependencies, primarily due to memory constraints. In contrast, transformers, as introduced by Vaswani et al. [72], represent a revolutionary shift in NLP architectures. Central to transformers is the self-attention mechanism, which enables each word in a sequence to attend to every other word, hence capturing contextual relationships irrespective of their relative positions. This design allows transformers to assimilate information from the entire text concurrently, leading to a richer comprehension of global dependencies. Building on this foundation, the following subsections will delve into some of the most prominent transformer-based models.

2.2.1. BERT

Bidirectional Encoder Representations from Transformers (BERT) is a well-known transformer model introduced by researchers at Google [16]. Unlike previous models that process text sequences in a uni-directional manner, either left-to-right or right-to-left [59, 60], BERT is a deep bidirectional model pre-trained on vast text corpora with two objectives: masked language modeling (MLM) and next sentence prediction (NSP). Prior to processing, BERT uses the WordPiece tokenizer [79] to split text into tokens, which could be either whole words or subwords. For MLM, 15% of these input tokens are randomly masked, and the model is tasked with predicting them. In NSP, given two sentences A and B, BERT determines if B follows A in the original context. After such pre-training, BERT can be efficiently fine-tuned for specific downstream tasks using labeled data [16].

Shortly after the initial BERT model was released, Google introduced a multilingual BERT model (M-BERT[1]), pre-trained on the largest Wikipedia corpora across 104 languages. However, when it comes to the German language, **GBERT** [13]—a language model trained by deepset—has demonstrated superior performance over M-BERT in various downstream NLP tasks, including text classification and Named Entity Recognition (NER) [13]. Remarkably, GBERT also outperforms early German language models like GermanBERT (`bert-base-german-cased`[2]), another deepset creation, and BERT models by DBMDZ (`dbmdz/bert-base-german-cased`[3] and `dbmdz/bert-base-german-uncased`[4]) in text classification, NER, and question answering [13].

2.2.2. RoBERTa

RoBERTa (Robustly Optimized BERT Approach) was developed by Facebook AI as an enhanced version of the original BERT with a focus on optimizing its pre-training procedure [39]. This model undergoes longer training periods, uses larger batch sizes, and processes extended sequences. Moreover, RoBERTa was trained on a much larger data set—160 GB in size, dwarfing the 16GB data set originally used to train BERT. RoBERTa also eliminates the NSP task found in BERT, and replaces BERT's static masking in MLM with a dynamic masking strategy. This updated masking approach continually generates a new masking pattern for every sequence fed into the model during training. Collectively, these modifications result in a significant performance improvement over the previous BERT models [39].

Building on the advancements of RoBERTa, Facebook AI introduced **XLM-RoBERTa** (often shortened to XLM-R) [14][5], a transformer-based multilingual model that significantly outperforms M-BERT [16]. XLM-RoBERTa was pre-trained on a data set of 2.5 TB and covering 100 languages. This extensive training data allows it to demonstrate im-

[1] https://github.com/google-research/bert/blob/f39e88/multilingual.md
[2] https://huggingface.co/bert-base-german-cased
[3] https://huggingface.co/dbmdz/bert-base-german-cased
[4] https://huggingface.co/dbmdz/bert-base-german-uncased
[5] https://github.com/facebookresearch/fairseq/tree/main/examples/xlmr

pressive performance on cross-lingual classification, question answering, and sequence labeling tasks. Notably, it is the first multilingual model that achieves performance benchmarks rivaling those of strong monolingual models without compromising on individual language performance [14].

2.2.3. BART

Encoder-only transformer models like BERT [16] are remarkably effective for classification tasks, primarily due to their design which produces a single predictive output from an input. However, this characteristic renders them less suitable for generative tasks such as text summarization and machine translation [61]. Recognizing the limitations of encoder-only models and the need for transformers adept at sequence generation, the research landscape evolved to develop models like BART.

BART, short for Bidirectional and Auto-Regressive Transformers, is another groundbreaking model developed by Facebook AI [38]. Unlike the encoder-only architecture of BERT and RoBERTa, BART utilizes both an encoder and a decoder. During its pre-training phase, BART goes beyond mere token masking; it corrupts input sentences in various ways such as token deletion, document rotation, and text replacement. It then learns to reconstruct the original text using its encoder-decoder structure. This denoising approach makes BART versatile in its applications. It not only rivals RoBERTa in traditional downstream tasks like question answering and classification, but also sets new benchmarks in sequence-to-sequence generative tasks like abstractive summarization and machine translation [38].

2.2.4. T5

T5, or Text-to-Text Transfer Transformer, was introduced by researchers at Google [61]. Breaking from conventional NLP approaches, T5 treats every problem—be it translation, summarization or question answering—as a text-to-text problem. Simply put, everything is framed as producing an output text from a given input text. The model was trained on the "Colossal Clean Crawled Corpus" (C4)[6] data set—a vast data set cleaned of duplicates, incomplete sentences, and sensitive content [61]. The researchers undertook extensive experiments on T5, exploring a spectrum of NLP transfer learning methodologies, spanning model architecture, unsupervised pre-training objectives, and training strategies. Among their key findings was the superior performance of encoder-decoder models over encoder-only counterparts. They also found that fully updating all parameters of a pre-trained model during fine-tuning yielded better results than updates covering fewer parameters [61].

[6] https://www.tensorflow.org/datasets/catalog/c4

2.3. Vector Representation for Text Data

Transforming textual data into machine-understandable formats is crucial in the realm of NLP. Presently, the prevalent method of text representation is based on the distributional hypothesis proposed by Harris in 1954 [23], which states that words appearing in similar contexts tend to have similar meanings.

One simple vector representation technique is one-hot encoding. Using this approach, each word in a vocabulary is represented by a unique vector where only the one specific element that corresponds to the word is set to the value 1, while all other elements are set to 0. This technique underpins the Bag-of-Words (BoW) model [23] that describes texts solely by the frequency of their words, disregarding grammar and punctuation.

The BoW approach, despite its simplicity and computational efficiency, produces large and sparse vectors. Such vectors often fail to capture the semantic information and similarity between texts [7]. This limitation motivated the development of dense vector representations, commonly referred to as *embeddings*. Throughout this thesis, the terms *word vectors* and *embeddings* will be used interchangeably to denote the same concept. These dense representations can be broadly categorized into two types based on their adaptability to context: static embeddings and contextual embeddings.

2.3.1. Static Embeddings

In static word embeddings, each word in a vocabulary is mapped to a single vector. This vector encapsulates the semantic essence of the word, leading to similar word vectors for semantically related words such as synonyms. These embeddings are static, meaning that the representation of a word remains unchanged irrespective of its context in different sentences or documents.

Google's **Word2Vec** [51, 52] posits that a word's representation can be learned from its neighboring words within a certain window size. This algorithm learns word associations using neural networks with one of two model architectures: Continuous Bag of Words (CBOW) or Skip-Gram. CBOW is trained to predict the center word from the surrounding context words, while Skip-Gram aims to predict context words from the center word [51, 52]. Sent2Vec [55] extended the scope of CBOW from individual words to entire sentences, facilitating the inference and training of sentence embeddings. Additionally, Google devised a simple enhancement to its Word2Vec algorithm, designed specifically for longer texts like sentences and documents, known as *Paragraph Vector* or Doc2Vec [37].

Global Vectors for Word Representation or **GloVe** [58], developed at Stanford University, makes use of statistics to derive word meanings from global word-word co-occurrence counts. In contrast to Word2Vec's dependency on local context, GloVe creates a global word-word co-occurrence matrix with statistics from the entire corpus, which is then factorized to produce dense word vectors. By leveraging both global statistical information and local semantic relationship, GloVe has delivered remarkable results in tasks like word analogy, word similarity and Named Entity Recognition (NER) [58].

Despite the wide adoption of static embeddings like Word2Vec or GloVe in various NLP tasks, they present numerous limitations. As these embeddings generate a single representation for each word regardless of its multiple meanings or senses, they are inherently incapable of handling polysemy. The model cannot dynamically adjust the representation of a polysemous term based on the current context, leading to a loss in semantic details. Furthermore, these embeddings cannot efficiently handle out-of-vocabulary words without resorting to techniques like subword embeddings [9]. Such shortcomings underline the need for more flexible and context-aware word representations, leading to the emergence of contextual embeddings.

2.3.2. Contextual Embeddings

The intuition behind contextual embeddings is their ability to incorporate context into word representations, making them more sophisticated than their static counterparts, which can only assign a single global representation for each word. These context-dependent representations are thus capable of capturing more syntactic and semantic information of words in different contexts.

ELMo (Embeddings from Language Model) [59] computes word vectors by analyzing the complete sentence in which the target word appears, leveraging a bidirectional language model (biLM). This biLM operates using a multi-layer bidirectional LSTM architecture encompassing both a forward and a backward pass. For any given word, the forward pass processes the part of the sentence preceding the target word, while the backward pass processes the part following it. The outputs from these LSTMs are subsequently concatenated, with the final embeddings being a combination of outputs across all layers of the LSTMs. Notably, the top layer of the biLM shows better performance in word sense disambiguation, whereas the first layer achieves a higher accuracy in POS (part-of-speech) tagging. By integrating results across all layers, the model ensures that word representations preserve both syntactic and semantic nuances [59]. Nevertheless, one limitation of ELMo is its lack of deep bidirectionality as the forward and backward LSTMs are trained independently [16], limiting a simultaneous context integration.

BERT [16] (see Subsection 2.2.1), on the other hand, takes a deeper bidirectional approach to model context by attending to both the left and the right context simultaneously. While BERT has set new benchmarks across various classification tasks, its original design poses challenges when generating quality sentence embeddings. Surprisingly, these embeddings sometimes underperform in quality compared to static embeddings like GloVe [62]. The inherent architecture of BERT does not compute independent sentence embeddings, making it less optimal for tasks like calculating sentence similarity, causing the process to be time consuming. **Sentence-BERT** or **SBERT** addresses this challenge by specifically fine-tuning pre-trained BERT and RoBERTa network on Natural Language Inference (NLI) tasks using a siamese and triplet network structure [62]. This fine-tuning process equips SBERT to deliver sentence embeddings that can be easily compared using cosine similarity. Moreover, SBERT generates consistent fixed-size embeddings for entire input sentences, capturing their semantic essence irrespective of

sentence length. The obtained embeddings are semantically meaningful, ensuring that semantically similar sentences are closely mapped within the vector space. As a result, SBERT emerges as a versatile tool for various tasks including semantic textual similarity (STS) and transfer learning [62].

Interestingly, although contextual embeddings yield impressive and nuanced results, [4] pointed out that when presented with simple language and substantial labeled data, simpler and less expensive non-contextual embeddings can unexpectedly rival the performance of their more sophisticated contextual counterparts. This observation offers valuable insights for the design and selection of embedding models.

2.4. Similarity Calculation

As keywords aim to represent the essence of a document, it is crucial to measure the similarity between keywords and the document, or between the keywords themselves, as this significantly influences the assessment of the keyword quality.

The *cosine similarity* is a favored technique in similarity calculation, largely owing to its computational efficiency and interpretability. This metric computes the cosine of the angle between two vectors, obtained by dividing the dot product of the vectors by their magnitudes. Notably, this result is only dependent on the angle between the vectors, not their magnitudes, making it particularly suitable for texts where the content matters more than the length or word count.

The cosine similarity between two vectors $\mathbf{A}$ and $\mathbf{B}$ is calculated as follows:

$$\text{cosine similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}|| \times ||\mathbf{B}||} \tag{2.1}$$

Where the dot product of the vectors is:

$$\mathbf{A} \cdot \mathbf{B} = \sum_{i=1}^{n} A_i \times B_i \tag{2.2}$$

And the magnitude of a single vector $\mathbf{A}$ is:

$$||\mathbf{A}|| = \sqrt{\sum_{i=1}^{n} A_i^2} \tag{2.3}$$

The cosine similarity always lies in the interval $[-1, 1]$. In the context of comparing textual items, a higher similarity value indicates a greater degree of relevance and similarity in content between the texts. Conversely, a value of 0 indicates no relatedness, and a negative value suggests dissimilarity or contrasting content between the two texts. In this thesis, the result of this computation is also referred to as the *similarity score*.

2.5. Morphological Analysis

Morphological analysis in NLP delves into the internal structure of words, dissecting them into *morphemes*—the smallest units of a word that carry distinct meaning. Morphemes play a pivotal role in elucidating the origins, grammatical functions, and semantics of words. Techniques commonly employed for this analysis encompass stemming, lemmatization, part of speech tagging, and word form generation. Prominent open-source toolkits that offer these functionalities include NLTK [41] and SpaCy [28].

2.5.1. Stemming and Lemmatization

Both stemming and lemmatization serve the purpose of reducing words to their base or root form. However, their methodologies and results vary.

Stemming is a rather crude process that truncates prefixes and suffixes from a word to obtain its base form or stem. As a result, the produced stem might not always be a valid word in the language.

Lemmatization, on the other hand, is a more sophisticated approach that aims to reduce a word to its base or dictionary form, often termed the *lemma*. This process usually involves the use of a comprehensive dictionary or lexicon.

To illustrate, stemming the word "happily" could yield "happi", which is not a proper word in English. In contrast, lemmatization would correctly derive the lemma "happy".

2.5.2. Part-of-Speech Tagging

Part-of-Speech (POS) tagging, often referred to as grammatical tagging, is the task of assigning grammatical labels to each word in a text corpus. These labels identify the word's part of speech like noun, verb, and adjective, but can also indicate other grammatical attributes such as tense, number, and case. Since many words could serve multiple grammatical roles depending on their context, POS tagging is not as simple as referencing a table of words and their parts of speech. The Hannover Tagger or Hanta [77], building upon the widely-used NLTK toolkit [41], addresses the lack of a dedicated POS tagger and lemmatizer for the German language.

2.5.3. Word Form Generation

Word form generation involves generating various inflected or derived forms of a word. One notable package for the English language that aids this task is `wordforms`,[7] which is capable of conjugating verbs and connecting parts of speech. For instance, given the

[7] https://github.com/gutfeeling/word_forms

word "operate", it outputs the following:

Noun: {operation, operations, operative, operatives, operator, operators}
Adjective: {operant, operative}
Verb: {operate, operated, operates, operating}
Adverb: {operatively}

Generating word forms is essential for the purpose of creating a comprehensive set of keywords, as it facilitates the ability to capture diverse expressions of a concept in textual data. To the best of our knowledge, no exact equivalent to `word-forms` is available for the German language. A less comprehensive alternative is the `german-nouns` [19][8] package, which contains around 100 thousand German nouns and specializes in inflecting them, specifically adjusting for grammatical properties like case and number.

2.6. Lexical Substitution

Lexical substitution, a task formally introduced at the SemEval-2007 [44], involves the generation of suitable replacements for a word within a given context. A suitable substitute is one that matches the meaning of the target word while also fits into the specific context.

Traditionally, most methodologies have tackled this task by incorporating some external lexical resources. Melamud et al. [46] proposed a simple approach based on a pre-trained skip-gram word embedding model [52], which ranks substitute candidates according to their similarity to both the target word and its surrounding context. Subsequent works utilizing the same principle of word/context embedding similarity have demonstrated improved performance over this approach [63, 45].

Zhou et al. [80] noted that these earlier approaches frequently failed to identify good candidates and neglected the impact of the substitution on the overall meaning of the sentence. Consequently, they proposed a BERT-based lexical substitution approach utilizing a dropout method. Specifically, the target word is partially masked for BERT to propose substitute candidates. These candidates are then evaluated according to whether their substitution changes the contextualized representation of the entire sentence. A more recent work, currently in development, proposed an enhancement to the dropout method by concatenating two instances of the sentence in question, one in its original form and another with the target word masked. This modified context is then fed into the model, which evaluates potential substitutes based on the impact of their inclusion on the entire context, rather than solely on the sentence in isolation, allowing for a more nuanced and context-sensitive evaluation of potential substitutes.

Integrating both categories of lexical substitution approaches, LexSubCon [49] leverages both BERT's contextualized embeddings and lexical knowledge bases like WordNet to identify potential synonyms. This integrated strategy has demonstrated superior performance over the dropout method proposed by Zhou et al. [80].

[8] https://github.com/gambolputty/german-nouns

2.7. Lexical Databases

Lexical databases, commonly referred to as *wordnets*, constitute lexical resources with information about words, their meanings and their relationships. In contrast to dictionaries which aim to define words, lexical databases often group words in *synsets*, which are sets of synonyms that represent the same concept [53]. These synsets are interlinked by various lexical and semantic relations including antonomy, hyponomy (subordination), hypernymy (superordination), meronymy (part-of relations), and more [53]. These relations result in a complex network of words, hence the name "wordnet". Recent advancements have seen the integration of wordnets with embedding models, including approaches to refine the learning of embeddings [65, 69] and enhance automatic wordnet construction through the application of embeddings [2].

The Princeton WordNet [53, 17] is arguably the most widely-known and utilized lexical database for the English language. Its structure, with nouns, verbs, adjectives, and adverbs organized into over 117,000 synsets, has played an important role in the development of similar databases for various languages.

Similar lexical-semantic nets exist for the German language. Inspired by the Princeton Wordnet, GermaNet [22, 26] has a similar structure to WordNet with several adjustments, including a hierarchical structure for adjectives. A more recent resource is OdeNet or Open German WordNet[9] [67], an open-source German wordnet first designed in 2017 at the Darmstadt University of Applied Science. It is built upon existing resources including the OpenThesaurus German synonym lexicon[10] and the Open Multilingual WordNet[11] which is linked to the Princeton WordNet [67].

[9] https://github.com/hdaSprachtechnologie/odenet
[10] https://www.openthesaurus.de/
[11] https://omwn.org/

3. Related Works

In this chapter, we focus on keyword extraction and generation techniques that integrate domain knowledge. We first introduce prior approaches that incorporate external knowledge into keyword extraction, highlighting their significance and challenges. We then present KeyBERT [21][1], an extraction tool capable of incorporating domain knowledge using seed keywords. Finally, we explore prior works on domain knowledge-guided and domain-specific keyphrase generation. This overview sets the foundation for the methodologies presented in the subsequent chapters of this thesis.

3.1. Incorporating Domain Knowledge in Keyword Extraction

To the best of our knowledge, there has been no prior work that incorporates the semantic meaning of domain knowledge into keyword extraction without relying on labeled data. Existing approaches predominantly focus on integrating statistical, syntactic, or linguistic features or leveraging external annotated databases as prior knowledge, as seen in the works of Gollapalli et al. [20] and He et al. [25].

Gollapalli et al. [20] approached the task of keyphrase extraction as a sequence labeling problem, incorporating expert knowledge by using feature labeling. Yet, the features they considered were predominantly statistical or linguistic in nature. These include part-of-speech tags, whether a phrase appears in the title, and even external features like previous identifications of a phrase as a keyphrase in labeled data sets. This indicates that, while the method does incorporate a certain degree of expert knowledge, such knowledge is primarily derived from the data's structure or previous annotations rather than a comprehensive understanding of a specific domain itself.

He et al. [25] integrated external knowledge by utilizing a controlled keyphrase vocabulary specific to a domain for keyphrase extraction. This vocabulary is constructed from a corpus of annotated documents with labeled keyphrases, with each keyphrase assigned a probability score reflecting its frequency of being selected as a keyphrase by authors of the documents in the corpus. Subsequently, keyphrases from the target document are ranked and extracted based on this prior probability, TF-IDF, and TextRank [50]. Although this approach has significantly outperformed TF-IDF and TextRank [25], it requires a substantial amount of data annotation, which is unavailable in our case. Furthermore, it solely considers simple features like TF-IDF and co-occurrences, thereby hindering a comprehensive understanding of the context surrounding the keyphrases. In addition, the prior knowledge is represented merely as a probability score, devoid of

[1] https://maartengr.github.io/KeyBERT/

any understanding of its actual meaning. This makes the prior knowledge ineffective if the keyphrases in the vocabulary do not appear in the target document.

More recent works have started to integrate external databases into the keyword extraction process. Altuncu et al. [3] introduced a post-processing method that enhances existing keyword extraction techniques by prioritizing candidate keywords found in domain-specific thesauri or identified as named entities in Wikipedia, provided they do not include overly general terms. However, this method is contingent upon the availability of a comprehensive and accurate thesaurus for the relevant domain, which may not always be readily accessible due to the extensive effort required to compile and maintain such a resource.

As a result, previous approaches are often suboptimal for domain-specific keyword extraction, particularly when extensively annotated domain-specific resources are unavailable. In the following, we will discuss an existing algorithm, KeyBERT, that incorporates lightweight domain knowledge without the need of extensive training, annotation, or external resources, thereby offering a more flexible and adaptable solution for various domains and applications.

3.1.1. KeyBERT

KeyBERT [21] is a minimalistic and straightforward keyword extraction algorithm that leverages BERT embeddings. It is designed to identify the sub-phrases in a document that are most similar to the overall document, serving as its representative keywords. KeyBERT utilizes BERT by first transforming the document and potential candidate keywords into embeddings. It then compares the similarity between the document embedding and candidate keyword embeddings. Specifically, it employs cosine similarity to score each candidate keyword. Additionally, KeyBERT offers flexibility in selecting an embedding model, allowing users to choose from a range of pre-trained models tailored to specific needs.

For the extraction process, KeyBERT provides the `extract_keywords` method. This method requires the document(s) as its primary input, from which it extracts keywords and keyphrases. Additionally, it accepts several other optional parameters that control the extraction process. For example, `keyphrase_ngram_range` controls the word length of the extracted keywords or keyphrases, `top_n` specifies the number of top keywords or keyphrases to return for each document, and `stop_words` specifies which words should be removed from the document.

One notable feature of KeyBERT is its ability to incorporate domain knowledge in the form of seed keywords. These seeds can be provided using the input parameter `seed_keywords` to steer the extraction. When a document is passed into KeyBERT, it is embedded and its embedding is stored in a `doc_embeddings` variable. If seed keywords are passed in, KeyBERT first computes the mean embedding of the seeds and stores it in a `seed_embeddings` variable. The `doc_embeddings` is then updated to a weighted average between the original document embedding and the mean seed embedding. Specifically, it is computed as:

$$\texttt{doc\_embeddings} = \frac{\texttt{doc\_embeddings} \times 3 + \texttt{seed\_embeddings}}{4} \tag{3.1}$$

The score of each candidate keyword is then calculated by comparing its cosine similarity to the `doc_embeddings`.

An additional advantage of KeyBERT is that it is open-source, meaning that it can be modified to suit specific needs. For example, the weight of the `seed_embeddings` in the doc_embeddings can be adjusted, or a different similarity measure can be used instead of cosine similarity. This flexibility makes KeyBERT a valuable tool for researchers and practitioners working on keyword extraction in specialized domains.

3.2. Domain Adaptable Keyword Generation

Keyword generation, crucial for various applications like information retrieval and document classification, is a complex task that can be enhanced by the incorporation of domain knowledge. In this section, we will discuss a few notable approaches in this direction, highlighting their strategies, strengths, and limitations.

One notable approach is the Deep Keyphrase Generation (DKG) model proposed by Meng et al. [48]. While this model does not explicitly inject domain knowledge, it attempts to mimic a human annotator in generating keywords for a task, which can be considered as an indirect way of incorporating domain knowledge. The model leverages a modified sequence-to-sequence (seq2seq) model with a copying mechanism in the decoder, which enables the generation of absent keyphrases by combining words from the document in novel ways. However, this model has a significant limitation in that it can only generate phrases composed of words that appear, at least separately, in the source text.

A more recent work by Meng et al. [47] presents a three-stage pipeline for a domain-adaptable keyphrase generation model, addressing the issue of domain gaps that hinder generalization in the keyphrase generation task. Initially, the model is pre-trained using open-domain knowledge from Wikipedia data. This knowledge is then distilled into new domain data through an iterative process, ultimately allowing the model to be fine-tuned using labeled data specific to the target domain. However, its limitations lie in the need for a substantial amount of labeled data in the target domain for fine-tuning. This dependency on labeled data, albeit reduced, raises concerns about the model's applicability and performance in domains where such data is scarce or unavailable.

An approach to domain-specific keyword generation is presented in [30], where the authors utilize *WordNet* for keyword generation in the 21 sections of the Polish business classification document. While this bears some similarity to our synonym generation (Subsection 4.6.2), it uses hyponyms, hypernyms, and cohyponyms, in addition to synonyms, to generate keywords. Subsequently, a simple filtering process is performed, ensuring that every keyword can only belong to a single section. Although their approach increases the extensiveness of the generated keyword sets, there are some significant limitations: 1) using all lexical relations including hypernyms could introduce overly

generalized keywords that are not domain specific and 2) a filtering process without word sense disambiguation could result in the inclusion of words that do not belong to the section. Both of these factors would diminish the accuracy and the relevance of the keyword sets for the target section. In contrast, our approach aims to achieve a higher accuracy and domain relevance by only considering synonyms and incorporating a more sophisticated filtering process of word senses.

4. Methodology

This chapter outlines the methodology employed for domain-specific keyword extraction and generation for this thesis. We start with clarifying key terminologies and describing the data set to set a clear foundation. Subsequently, we will delve into our overarching approach, spanning the phases of preprocessing, extraction and generation of keywords. Each phase is marked by specific techniques and parameters, further elaborated in the subsequent sections. By the end of this chapter, the design and methodology adopted for our evaluation efforts is detailed.

4.1. Terminology

Before exploring the methodology, it is important to clarify certain terms that are fundamental for a comprehensive understanding of this thesis.

- **Class**: A class is a predefined set or category of items that share common characteristics or themes. For instance, in a data set of commercial registers, classes can be defined by business sectors such as "real estate", "manufacturing", or "education". In this thesis, the terms "class", "predefined class", and "target class" are used as synonyms.
- **Class description**: A class description is a short piece of text that provides a concise summary of the attributes, content, or characteristics of a given class.
- **Class-specificity**: Class-specificity denotes the degree to which a term, phrase, or feature is uniquely associated with a particular class. A highly class-specific keyword would be one that is strongly relevant in one class as opposed to others.
- **Keyword**: Within the scope of this thesis, the term "keyword" is used to denote a "class-specific keyword"—a term or phrase that holds significant relevance to a particular class due to its capacity to embody the core themes of the class. The terms "keyword" and "keyphrase" are employed synonymously. Both can refer to a single word or a phrase comprising multiple words.
- **Seed keywords**: Seed keywords are the initial set of terms or phrases that are highly representative of a specific class. In the context of this thesis, seed keywords are predefined. They serve as a starting point for the subsequent extraction and generation of additional class-specific keywords.
- **Synset**: A synset, or synonym set, is a group of synonoymous words and phrases that express the same idea.

4.2. Data Set

For this thesis, we utilize the *German Business Registry (BR)* (*Handelsregister* in German) database as our data source for extracting keywords. The full BR contains over 2.3 million entries in the German language, with each entry representing a registered business in Germany. For the purposes of this study, we have retained only selected attributes, including a unique identifier (`fb_id`), an official business name, and a `purpose` attribute describing the business purpose. Our keyword extraction task exclusively targets the `purpose` attribute. To expedite processing, we work on a random subset of 10,000 entries from the database, often referred to as the "sampled data set", "data subset", or simply "subset" throughout this thesis. Figure 4.1 provides an excerpt from the data set.

fb_id	legal_name	purpose
bf5c6bceb6de50e0a6526fc691ef421c	Soso Handels GmbH	Groß- und Einzelhandel sowie Import und Export von Textilien, Getränken und Souvenirs aller Art
cde00cf49cee3316f3b18542e09b593d	Öl & Gasfeuerung Teschang GmbH	Zentralheizungsbau, Gas- und Wasserinstallation, Wartung und Reparatur von Öl- und Gasfeuerungsanlagen.
fec9d3090326e923203b84d51ba3f531	ATM Verwaltungs GmbH	Die Stellung einer persönlich haftenden Gesellschafterin bei Personengesellschaften, insbesondere bei der ATM Leitungsbau GmbH & Co. KG.
6f5b83eae7bb2daddd433ace2002c58e	Hartha West GmbH	Verwaltung eigenen Vermögens, insbesondere Erwerb, Halten und Verwaltung von Immobilien und Beteiligungen auf eigene Rechnung und im eigenen Namen
7e1ed40a6b38a7091508670e797e657b	Landtechnik Steprath GmbH	Ankauf, Verkauf und Service von Geräten aller Art für die Bau- und Agrarwirtschaft einschließlich aller damit in Zusammenhang stehenden Geschäfte und Tätigkeiten.
...	...	...

Figure 4.1.: Entry examples from the German BR dataset

As another foundational document for guiding our extraction methodology, the Klassifikation der Wirtschaftszweige (WZ2008), based on the Statistical Classification of Economic Activities in the European Community (NACE Rev. 2) (EUROSTAT European Commission, 2006),[1] systematically and uniformly categorizes the economic activities of German businesses within a predefined hierarchical framework. This hierarchy encompasses five levels: 21 sections (Abschnitte), 88 divisions (Abteilungen), 272 groups (Gruppen), 615 classes (Klassen), and 839 subclasses (Unterklassen). Figure 4.2 presents an example of the five hierarchical levels within a specific section.

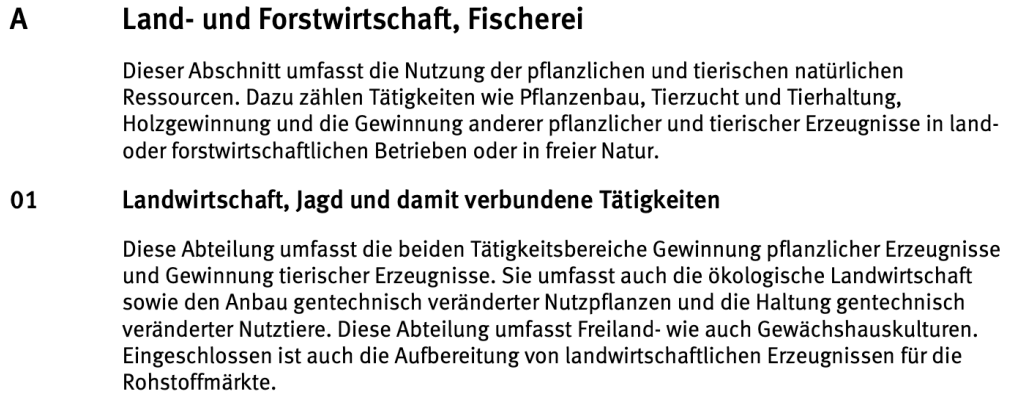
A	**LAND- UND FORSTWIRTSCHAFT, FISCHEREI**	**Abschnitt (Section)**	
01	**Landwirtschaft, Jagd und damit verbundene Tätigkeiten**	**Abteilung (Division)**	
01.1	**Anbau einjähriger Pflanzen**	**Gruppe (Group)**	
01.11	Anbau von Getreide (ohne Reis), Hülsenfrüchten und Ölsaaten	Klasse (Class)	0111
01.11.0	Anbau von Getreide (ohne Reis), Hülsenfrüchten und Ölsaaten	Unterklasse (Subclass)	

Figure 4.2.: Excerpt from the WZ2008 structure with marked hierarchy

Each section contains a textual description detailing its tasks, specifying what is included within and what falls outside its scope (Figure 4.3). These descriptions offer invaluable domain knowledge that can be injected in our pipeline, guiding the extraction of class-specific keywords. When referencing the WZ2008 classification in the context of this thesis, unless otherwise specified, the term "class" denotes the section (Abschnitt)—the most overarching tier in the hierarchical structure, and "class-specific" denotes the

[1] https://ec.europa.eu/eurostat/web/products-manuals-and-guidelines/-/ks-ra-07-015

specificity to a section rather than a class (Klasse) or subclass (Unterklasse) in its hierarchy. In essence, our approach extracts and generates keywords for each section of the WZ2008 using our sampled data set.



A **Land- und Forstwirtschaft, Fischerei**

Dieser Abschnitt umfasst die Nutzung der pflanzlichen und tierischen natürlichen Ressourcen. Dazu zählen Tätigkeiten wie Pflanzenbau, Tierzucht und Tierhaltung, Holzgewinnung und die Gewinnung anderer pflanzlicher und tierischer Erzeugnisse in land- oder forstwirtschaftlichen Betrieben oder in freier Natur.

01 **Landwirtschaft, Jagd und damit verbundene Tätigkeiten**

Diese Abteilung umfasst die beiden Tätigkeitsbereiche Gewinnung pflanzlicher Erzeugnisse und Gewinnung tierischer Erzeugnisse. Sie umfasst auch die ökologische Landwirtschaft sowie den Anbau gentechnisch veränderter Nutzpflanzen und die Haltung gentechnisch veränderter Nutztiere. Diese Abteilung umfasst Freiland- wie auch Gewächshauskulturen. Eingeschlossen ist auch die Aufbereitung von landwirtschaftlichen Erzeugnissen für die Rohstoffmärkte.

Figure 4.3.: Excerpt from section A of WZ2008

4.3. General Approach

The primary objective of our approach is to produce a complete set of keywords for each class. Initially, this set comprises only the predefined seed keywords associated with each class. Our methodology is structured around two stages: keyword extraction (Section 4.5) and keyword generation (Section 4.6), applied on each class independently.

During the **extraction** phase, class-specific knowledge, provided in the form of predefined seed keywords or class description, is injected into our selected keyword extraction algorithm. Since a substantial portion of the extracted keywords may not be class-specific or relevant, we deploy a damping function to retain the most pertinent ones and exclude the less relevant ones. The selected keywords are subsequently added into the keyword set. The extraction process can be summarized using Algorithm 1.

Algorithm 1 Keyword Extraction (simplified)

```
1: function EXTRACTION(dataset, seedKeywords)
2:     extractedKeywords ← Inject class-specific seedKeywords into extraction
3:     relevantKeywords ← Apply damping function to extractedKeywords
4:     return relevantKeywords
5: end function
```

The **generation** phase ensues, which can be further divided into three distinct steps: lexical substitution, synonym generation, and word form generation. The pseudocode depicting the generation process is detailed below in Algorithm 2.

We first employ lexical substitution on the class's keyword set, which now contains both the seed and selected extracted keywords. This substitution process utilizes class

Algorithm 2 Keyword Generation

function LexicalSubstitution(keywordSet, classDescription)
    substitutedKeywords ← Using classDescription as context, apply lexical substitution on keywordSet
    filteredSubstitutes ← Filter substitutedKeywords for relevance
    return filteredSubstitutes
end function
function SynonymGeneration(keywordSet)
    synonyms ← Extend keywordSet with relevant synonyms
    return synonyms
end function
function WordFormGeneration(keywordSet)
    wordForms ← Generate various word forms for each keyword in keywordSet
    return wordForms
end function

descriptions as contexts to enhance the relevance of the substitute words. A filtering process further refines the substitution results to ensure class-specificity. These refined results are then incorporated into the keyword set. Subsequently, the synonym generation step operates on this refined set, extending it with relevant synonyms. In cases where words possess multiple meanings or senses, only the synset most similar to the seed keywords is kept. Finally, the word form generation step is applied to the collective set derived from all previous steps. In the end, a comprehensive set of keywords is produced for the class. The entire pipeline, integrating both the extraction and generation phases, is depicted in Algorithm 3.

Algorithm 3 Complete Keyword Extraction and Generation Pipeline

Input: Predefined seed keywords, class description, extraction dataset
Output: Comprehensive set of keywords for each class
procedure KeywordPipeline
    for each class **do**
        keywordSet ← Predefined seed keywords
        keywordSet ← keywordSet ∪ Extraction(dataset, seedKeywords)
        keywordSet ← keywordSet ∪ LexicalSubstitution(keywordSet, classDescription)
        keywordSet ← keywordSet ∪ SynonymGeneration(keywordSet)
        keywordSet ← keywordSet ∪ WordFormGeneration(keywordSet)
    end for
    return Comprehensive set of keywords for each class
end procedure

An illustrated workflow of the complete extraction and generation process for each

class can be found in Figure 4.4.

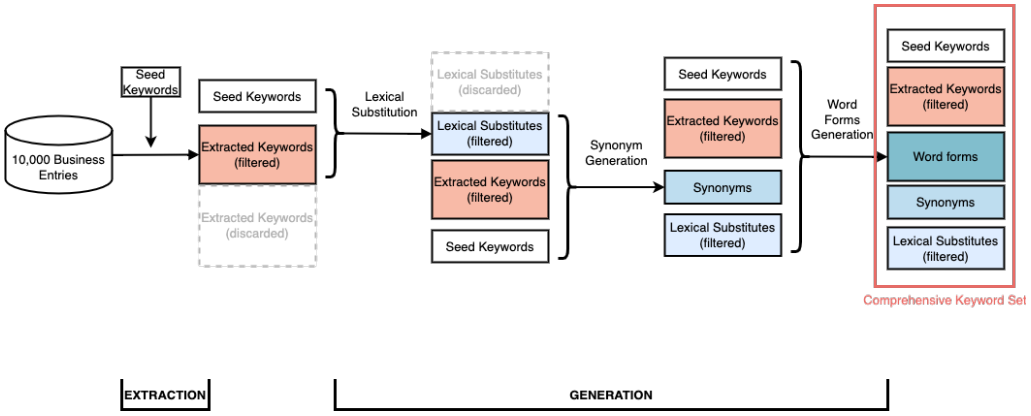


Figure 4.4.: Illustrated workflow of our proposed approach, performed on each class individually.

It is important to highlight that during the initial stages of our methodology's development, a preprocessing step was contemplated. However, it was eventually omitted as it did not enhance performance. The following sections detail the criteria, rationale and procedures for each stage of our approach. It should be noted that the unit of each processing step is one predefined class, meaning the entire pipeline runs separately for every individual class.

4.4. Preprocessing

Since we are extracting keywords for each class individually, it became apparent that only a small portion of the 10,000 entries in the subset data would be relevant for each class. Therefore, we initially designed a preprocessing step to retain only these relevant entries. An overview of the preprocessing steps is given below. The following subsections will provide a detailed account for each step.

1. Define a class description and a set of seed keywords for each class.
2. Identify all entries in the subset data that contain any of the seed keywords and compute their similarity to the class description. Use a statistical measurement of the similarity scores as a baseline, such as the average or specific percentiles.
3. Compute similarity scores of all entries to the class description. Only those scoring above the defined baseline will be considered for subsequent keyword extraction.

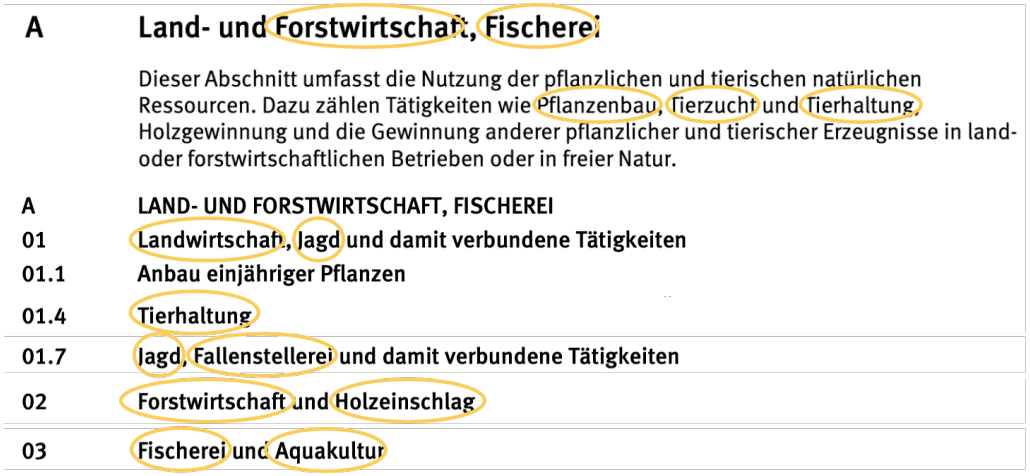
4.4.1. Defining Class Description and Seed Keywords for Preprocessing

Although each section in the WZ2008 classification comes with a textual description, directly employing it as our class description presents challenges. Specifically, the length

and content of these descriptions vary by section. Some descriptions not only detail what is included but also specify what is excluded. Using this description directly might thus introduce unnecessary noise when comparing its similarity to other texts.

To standardize these descriptions, we used a summarization tool to constrain text length between 20 to 45 words. We explored the `t5-base`[2] and `google/flan-t5-base`[3] models, which produce extractive summaries by simply extracting sentences from the text. In order to potentially obtain more dynamic content, we also utilized abstractive summarization and paraphrasing models. Due to limited availability of German models in these two tasks, some of our selected models, including `facebook/bart-large-cnn`,[4] `humarin/chatgpt-paraphraser-on-T5-base`,[5] and `prithivida/parrot-paraphraser-on-T5`,[6] were primarily trained on English data.

For preprocessing, seed keywords were required to perform exact string matching on the data set. We first defined some seed keywords for each section by manually extracting keywords from its title, description, and associated divisions and groups. An example of such definitions can be found in Figure 4.5, where the circled words and phrases are deemed as potential keywords. A detailed record of these manually defined seed keywords is available in Appendix A.



A	**Land- und Forstwirtschaft, Fischerei**
	Dieser Abschnitt umfasst die Nutzung der pflanzlichen und tierischen natürlichen Ressourcen. Dazu zählen Tätigkeiten wie Pflanzenbau, Tierzucht und Tierhaltung, Holzgewinnung und die Gewinnung anderer pflanzlicher und tierischer Erzeugnisse in land- oder forstwirtschaftlichen Betrieben oder in freier Natur.
A	LAND- UND FORSTWIRTSCHAFT, FISCHEREI
01	Landwirtschaft, Jagd und damit verbundene Tätigkeiten
01.1	Anbau einjähriger Pflanzen
01.4	Tierhaltung
01.7	Jagd, Fallenstellerei und damit verbundene Tätigkeiten
02	Forstwirtschaft und Holzeinschlag
03	Fischerei und Aquakultur

Figure 4.5.: Section A in WZ2008 as an example for the definition of class-specific seed keywords. Potential seed keywords from the section's name, description and outline are circled.

However, similar to the length variation of the class descriptions, the number of such manually defined keywords varies between classes. In order to ensure uniformity across seed keywords, we decided to extract keywords from the newly refined class descriptions using the KeyBERT extraction algorithm, while still utilizing our manually defined

[2]https://huggingface.co/t5-base
[3]https://huggingface.co/google/flan-t5-base
[4]https://huggingface.co/facebook/bart-large-cnn
[5]https://huggingface.co/humarin/chatgpt_paraphraser_on_T5_base
[6]https://huggingface.co/prithivida/parrot_paraphraser_on_T5

keywords as the `seed_keywords` input parameter. Given the observation that the majority of the manually defined keywords were nouns or noun phrases, we restricted our approach to extract five nouns or noun phrases from the class description. Since the class summaries were short and may not contain many nouns, we also randomly appended five manually defined keywords into a context to the end of the class summaries as a guide for the seed keywords extraction. Moreover, we prefixed each summary with the name of the corresponding WZ2008 section. Therefore, our final class description started with the class name, followed by the model-generated summary, and ended with a sentence containing five seed keywords of the class. For this extraction process, we employed KeyBERT's default model, `sentence-transformers/all-MiniLM-L6-v2`.[7]

At this point, we assumed that the seed keywords of each class should be unique. If a keyword appeared in multiple classes, it was deemed class-unspecific and consequently removed. Therefore, we added a postprocessing step to eliminate such generic keywords after the extraction.

4.4.2. String Matching and Similarity Baseline Computation

Since we only want to extract keywords from relevant entries specific to a class, a quantifiable measure for this relevance is essential. We utilized the seed keywords extracted from the class descriptions as initial relevance measures. First, using exact string matching, we identified all entries within our data subset that contained at least one of the seed keywords associated with a class. Subsequently, for every matching entry, we computed its similarity to the class description. Statistical measures such as the mean, median and specific percentiles of the similarity scores were then used to establish a baseline. For our experiments, the 25th, 50th, and 75th percentiles were tested as potential baselines. The results corresponding to these baselines are elaborated in Subsection 5.1.4. We employed several transformer models to embed texts and evaluate similarity scores, including:

- `sentence-transformers/all-MiniLM-L6-v2`
- `sentence-transformers/distiluse-base-multilingual-cased-v1`[8]
- `symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli`[9]
- `deutsche-telekom/gbert-large-paraphrase-cosine`[10]

4.4.3. Baseline-Guided Similarity Search

Upon establishing the baseline for each class, we proceeded to calculate the similarity between our subset data and the defined class descriptions. Only data set entries

[7] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[8] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1
[9] https://huggingface.co/symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli
[10] https://huggingface.co/deutsche-telekom/gbert-large-paraphrase-cosine

surpassing the respective class baseline were filtered and considered eligible for the following class-specific keyword extraction. For computing this similarity, the models used were consistent with the ones employed for similarity calculations for the matched entries in Subsection 4.4.2. Detailed results from this baseline-guided similarity search are in Subsection 5.1.4.

4.5. Keyword Extraction

Our keyword extraction methodology is rooted in leveraging class-specific knowledge and refining results for utmost relevance. This section delineates the structured methodology we adopted as well as the rationale behind each decision. A detailed pseudocode of the extraction process can be found in Algorithm 4.

Algorithm 4 Keyword Extraction (Detailed)

```
function EXTRACTION(seedKeywords, dataset, iterations, percentile)

    for i ← 1 to iterations do
        candidateKeywords ← Extract candidate unigrams from dataset
                                        ▷ Keyword Score Computation (Subsection 4.5.7)
        Initialize keywordScores as an empty dictionary
        meanSeedEmbedding ← Generate mean embedding from seedKeywords
        for each candidateKeyword in candidateKeywords do
            meanSeedScore ← Calculate similarity to meanSeedEmbedding
            maxSeedScores ← Calculate similarity to each seedKeyword individually
            maxSeedScore ← Find the highest score among maxSeedScores
            finalScore ← Average of meanSeedScore and maxSeedScore
            keywordScores[ candidateKeyword ] ← finalScore
        end for
                                                ▷ Keyword Damping (Subsection 4.5.8)
        dampingValue ← 5 * (ln(|candidateKeywords|) - ln(0.001))
        Sort keywordScores in descending order
        relevantKeywords ← keywordScores[: dampingValue]
        seedKeywords ← Add top keywords scoring above percentile from
    relevantKeywords to seedKeywords                ▷ Iteration (Subsection 4.5.6)
    end for
    return relevantKeywords
end function
```

4.5.1. Seed Keyword Formulation

As introduced in 4.4.1, we manually defined some potential seed keywords for each section of WZ2008 based on its textual description and outline. More concretely, these

keywords were largely derived from the titles of the WZ2008 sections and their respective divisions. Unlike our approach detailed in 4.4.1, where only five keywords were extracted from the refined class description, in this step we utilized all manually defined keywords as seed keywords to steer our extraction process. These seed keywords, along with the class description, serve as domain knowledge, guiding our pipeline to extract and generate additional domain-relevant and class-specific keywords. All defined seed keywords for the WZ2008 sections can be found in Appendix A.

4.5.2. Model Selection for Extraction

For scoring and extracting keywords in our modified KeyBERT algorithm, which we will refer to as *KeyBERTMod*, we employed two different models from the HuggingFace Transformers library: `sentence-transformers/distiluse-base-multilingual-cased-v1` and `deutsche-telekom/gbert-large-paraphrase-cosine`. Both are BERT-based SentenceTransformers models tailored for generating and comparing sentence embeddings. While the former supports multiple languages, the latter, based on GBERT, was exclusively trained on German. A comparative analysis regarding their performance is available in Subsection 5.2.1.

4.5.3. Keyword Scope Delimitation

The `extract_keywords` function in KeyBERT offers flexibility in defining the scope of our keyword extraction, allowing us to set both the part of speech and the length of the extracted units. In our approach, we considered two different scopes of extraction: focusing solely on nouns and targeting only unigrams (single words). Intuitively, nouns have a higher likelihood to be keywords, especially in technical domains [33]. Our observations align with this assertion, as the majority of our manually defined keywords from the WZ2008 classification were nouns and noun phrases (Appendix A). For noun extraction in KeyBERT, we utilized the dedicated KeyphraseCountVectorizer[66][11] and restricted the POS to only nouns. The following Python code snippet illustrates this process:

```
from keyphrase_vectorizers import KeyphraseCountVectorizer
from keybert import KeyBERT
# specify the POS pattern and language
k_vectorizer = KeyphraseCountVectorizer(spacy_pipeline="de_core_news_lg",
                                        pos_pattern="<N.*>+",
                                        stop_words="german")

keyword_extractor = KeyBERT()

# extraction using KeyBERT,
# docs are the document(s) to extract keyword from
keyword_extractor.extract_keywords(docs, vectorizer=k_vectorizer)
```

[11] https://github.com/TimSchopf/KeyphraseVectorizers

Our alternative method focused on extracting unigrams. The underlying rationale is that if a phrase is indeed a keyphrase, its constituent words are likely identified as keywords. When progressing to the next stage in the CD4AI pipeline on context window extraction, the complete keyphrase should be identifiable by examining the context windows of its individual constituent words, which were previously extracted as keywords. The *unigram-only* approach offers the advantages of faster extraction speed and obviates the need for external vectorizers. Since KeyBERT defaults to extract unigrams, there is no need to specify any additional parameters for this aspect.

4.5.4. Parameter Specifications for KeyBERTMod

The essential scoring mechanism in KeyBERT involves comparing the embedding of each candidate keyword with a `doc_embeddings` attribute, which captures both the embedding of the document containing these candidates as well as the seed embeddings of any provided seed keywords (Subsection 3.1.1).

Given our goal to derive class-specific keywords, the document itself—the business registry entry in our data set—does not contribute to the quality measurement of the extracted keyword. Therefore, we modified KeyBERT to emphasize a candidate's similarity to the seed keywords and disregard the influence of the document's embedding. This was done by adding `doc_weight` and `seed_weight` as input arguments and defaulting the former to 0. This design choice preserves flexibility; if we later decide to take the document into account, we can easily adjust the `doc_weight` value. In comparison to Equation 3.1, the modified `doc_embedding` is calculated as shown in Equation 4.1:

$$\texttt{doc\_embeddings} = \frac{\texttt{doc\_embeddings} \times \texttt{doc\_weight} + \texttt{seed\_embeddings} \times \texttt{seed\_weight}}{\texttt{doc\_weight} + \texttt{seed\_weight}} \tag{4.1}$$

With `doc_weight` defaulting to 0, our `doc_embeddings` essentially mirrors the `seed_embeddings`, translating to the mean seed embedding.

Given that our data set entries predominantly consist of short texts with an average length of 30 tokens, we confined our extraction to five keywords per entry, consistent with KeyBERT's default setting.

4.5.5. Extraction Strategies

We explored three different strategies for the extraction process. The first one, which we named *proprocessed extraction*, involves a shortened extraction after a preprocessing step. The second one is named the *assignment* strategy, which perceives the class-specific keyword extraction as an assignment problem. Lastly, the *guided KeyBERT* strategy is a guided extraction process utilizing seed keywords.

In employing the *preprocessed extraction* strategy, only entries surpassing a certain baseline for each class were considered for keyword extraction (Section 4.4). This resulted in handling mere hundreds or even fewer entries instead of the original 10,000 in the subset data, thus improving the extraction speed. Additionally, we incorporated the defined seed keywords (4.5.1) into the `seed_keywords` parameter of KeyBERT as a guide

for the extraction. We also ignored the content of the entry itself by setting `doc_weight` to be `0` (Subsection 4.5.4). We restricted the extraction scope to nouns or noun phrases and conducted the extraction individually for each class.

Both the *assignment* and the *guided KeyBERT* strategy operated on the entire data subset. The *assignment*, although similar to the first strategy in terms of only extracting nouns and noun phrases, did not utilize any seed keywords for guidance. Instead, after removing duplicates in the extracted keywords, each unique keyword's embedding was compared to the embeddings of each class's seed keywords. Keywords were then assigned to the class with which their similarity to the seed keywords was the highest. To account for extracted keywords that were not specific to any class, we created a separate class to capture generic keywords in order to maintain coherence and high class-specificity in the keyword sets of the original classes. This generic class was initialized with the 15 most frequent words in our subset data (excluding stopwords) and some manually selected generic words. We then removed any terms in this list that appeared in the seed keyword lists of the other classes. The resulting generic seed keywords were the following:

> *Art, insbesondere, Unternehmen, Gesellschaft, Verwaltung, Vertrieb, Gegenstand, Betrieb, Dienstleistungen, Erwerb, Durchführung, Übernahme, Geschäfte, Dienstleistungen, Aktivität, Aktivitäten.*

The third strategy, *guided KeyBERT*, was ultimately the one we employed for our subsequent extractions. Fundamentally, it expands upon the principles of the *preprocessed extraction* strategy by utilizing all $10,000$ entries in the subset data rather than a preselected few. Without any preprocessing, all data entries are input into KeyBERTMod along with the class-specific seed keywords, leading to the extraction of keywords that demonstrate the highest similarity to the given seeds. The detailed measures of this similarity are documented in Subsection 4.5.7.

4.5.6. Iterative Extraction

We employed an iterative approach for the extraction, where the top extracted keywords from each iteration are incorporated as seed keywords to guide the subsequent rounds. Our rationale stems from the concern that a single extraction pass based on a limited set of seed keywords might not capture the entirety of the class or domain. Through iterative refinement and expansion of the seed set, we aim to progressively encompass a broader spectrum of keywords, driving a more comprehensive and targeted extraction while minimizing the risk of overlooking significant keywords.

Among the three different extraction strategies introduced in Subsection 4.5.5, the first *preprocessed extraction* frequently yielded less than a hundred documents, an insufficient quantity for multiple iterations. The second *assignment* strategy does not require seed keywords during extraction, thus leaving *guided KeyBERT* as the only viable strategy for iterative extraction. From our subset of $10,000$ entries, we partitioned the data into five segments of $2,000$ entries each, leading to five distinct iterations. During each

iteration, we extended our seed set with keywords that achieved similarity scores above a certain configurable percentile to the seeds, which is provided as an input via the `percentile_newseed` parameter in our KeyBERTMod extraction algorithm. Given that existing seed keywords frequently attain high scores, we implemented a mechanism to ensure the inclusion of new seeds in each iteration: if all keywords above the selected percentile were already in the seed set, the next three highest-scoring keywords not in the seeds were added. This number of additional keywords (three in this case) could also be adjusted and is referred to as the `number_newseed` input parameter. Initially, we considered both the 99.5th and 99.9th percentiles for adding new seed. However, given that the 99.9th percentile rarely introduced new seeds, often leading us to default to the `number_newseed` setting, we opted for the more inclusive 99.5th percentile as our threshold. To avoid unnecessary reprocessing, we ensured that duplicated keywords in each class were removed after every iteration.

4.5.7. Keyword Score Computation

The *mean seed* method, the default in the original KeyBERT, generates a mean embedding from the seed keywords. Each candidate keyword is then scored based on its similarity to this mean seed embedding. An illustration of this computation is shown in Figure 4.6.

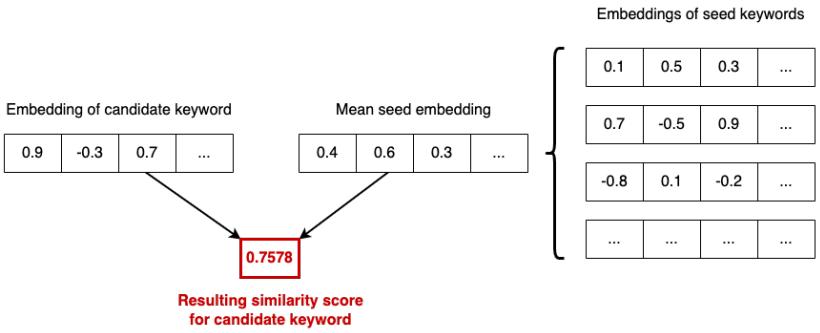


Figure 4.6.: Illustration of score computation using the *mean seed* method.

The *max seed* approach, on the other hand, calculates the similarity of a candidate keyword to each seed keyword individually. The highest score among these then becomes the final score for the candidate, as illustrated in Figure 4.7.

To combine the strengths of both scoring methods, we introduced the *average scoring* approach. This method averages the scores from both the *mean seed* and *max seed* approach for each candidate as its final score. If a candidate keyword emerges through only one scoring technique, either *mean seed* or *max seed*, we would not leave it unexamined by the other. Instead, we also calculate its score with the unused methodology and then average both scores. This dual evaluation allows for a robust and thorough assessment of each keyword, regardless of its origin in the extraction process.

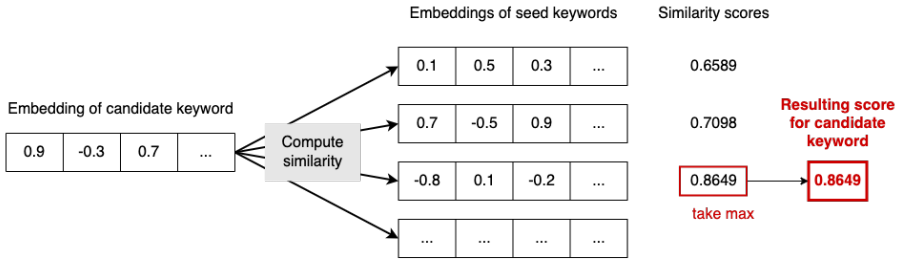


Figure 4.7.: Illustration of score computation using the *max seed* method.

4.5.8. Damping Function for Keyword Filtering

Utilizing the *guided KeyBERT* strategy, we extracted 5 keywords for each of the 10,000 entries. After eliminating duplicates, we had over 6,000 keywords extracted per class. It is evident that not every keyword in this expansive list would be genuinely class-specific, which motivated us to incorporate a filtering mechanism.

Intuitively, the larger the extracted keyword set was for a class, the more we should ideally select. To streamline our filtering process, we implemented a damping function to choose only the top-ranked keywords for each class, as shown in Equation 4.2:

$$x = k \cdot (\ln(n) - \ln(\alpha)) \tag{4.2}$$

Here, x is the number of top extracted keywords to be selected, and n is the total number of extracted keywords in the class. k represents a scaling factor that amplifies the effect of the function and α is a reference point for the damping function to take effect. To ensure we select only the most pertinent keywords, we set rather restrictive parameters with $k = 5$ and $\alpha = 0.001$. After the filtering, the selected top extracted keywords were added to the class's keyword set along with the seed keywords.

4.6. Keyword Generation

After filtering the extracted keywords, we combined them with the seed keywords to facilitate further keyword generation. Our generation approach composes of three consecutive stages: lexical substitution, synonym generation, and word form generation. Initially, we placed word form generation at the forefront. However, recognizing its inability to produce new base words, unlike the other two stages, we decided to shift it to the end. This shift also ensured the generation of word forms for the created lexical substitutes and synonyms. Further details of our generation methodology are discussed in the following subsections.

4.6.1. Lexical Substitution

The goal of lexical substitution is to find words that are both semantically similar to the substituted word and fitting in the specific context. For this purpose, a context needed

to be defined. In our approach, we designed the context in four ways by leveraging the combination of class descriptions and seed keywords differently. The class descriptions for this task were defined by manually refining the extractive summaries from the preprocessing step in Subsection 4.4.1, mainly by adding a few sentences from the WZ2008 section description to each class description to make it more comprehensive. Similar to the class summaries in Subsection 4.4.1, we maintained a similar length for the descriptions across different classes, averaging 57 words. A detailed listing of all of the manually refined class descriptions can be found in Appendix A. The seed keywords remained consistent with the ones manually defined in Subsection 4.5.1. The four contexts were designed as follows:

1. Combining both the **class description and all seed keywords**. The target word to be substituted will be placed in a text with the class description and all the seed keywords. Specifically, the context looks like this:

   [`class_description`]. Die Schlüsselwörter beinhalten: [`seed_keywords`], [`target_word`].

2. Only **class description and no seed keyword**. Instead of giving all the seed keywords, only the target word is mentioned after the class description:

   [`class_description`]. Ein Schlüsselwort ist [`target_word`].

3. With **no class description but all seed keywords**. Instead of a long class description, only the name of the class is used with all the seed keywords:

   [`class_name`]. Die Schlüsselwörter beinhalten: [`seed_keywords`],

   [`target_word`].

4. With **no class description and no seed keyword**. Only the name of the class and the target keyword is used:

   [`class_name`]. Ein Schlüsselwort ist [`target_word`].

For our concrete lexical substitution approach, we used a combination of the first three contexts, sidelining the last one due to its lack of contextuality. The targeted words for each class encompassed all existing words in the class's keyword set, which includes both the seed keywords and the top extracted keywords above the filtering point (Subsection 4.5.8). Each of the first three contexts underwent the substitution individually, and the results were subsequently congregated to create a set of substitute keywords.

The lexical substitution algorithm allows us to choose the specific model for the task as well as a SpaCy pipeline. We experimented with different models for the German language, including `xlm-roberta-base`, `bert-base-german-cased`, GBERT-based models from deepset (`deepset/gbert-base` and `deepset/gbert-large`), and DBMDZ (`dbmdz/bert-base-german-cased` and `dbmdz/bert-base-german-uncased`).

After the computation, we first eliminated all outputs that were already in a class's keyword set. Moreover, we needed a filtering step to ascertain class relevance. This

was done by retaining the substitute words with the highest similarity to the seed keywords of their respective class. Similar to the score computation in the extraction process (Subsection 4.5.7), we leveraged both the *mean seed* and *max seed* scoring approach, with their average forming the final similarity score for the substitute keyword. Our approach adopted the 75th percentile as the score threshold in order to ensure high class-specificity. Any substitute keywords scoring above this percentile in their respective class were integrated into the class keyword set for further processing. To provide more flexibility, we designed this filtering percentile as an input argument that can be adjusted according to the specific needs. For generating word embeddings and computing similarity, the model `deutsche-telekom/gbert-large-paraphrase-cosine` was used.

4.6.2. Synonym Generation

Although lexical databases lack context incorporation, their synset structure, which groups semantically similar terms together, can enrich our keyword set. In our pipeline, we employed OdeNet [67] to generate synonyms for the class's expanded keyword set, which now contains the seed keywords, the filtered extracted keywords, and the filtered lexical substitutes. We avoided including hypernyms due to their generic nature, which runs contrary to our required class-specificity. Hyponyms, on the other hand, are also unfitting since they describe a "part-of" relationship that is not always suitable for the context. For example, for the word "Bearbeitung" (processing) in the context of the manufacturing industry, the hyponyms in OdeNet include "Handel" (trade) and "Verleih" (hiring, renting), which are keywords both belonging to different classes according to our definition. Another example is the hyponyms "Preis" (price) and "Kosten" (cost) for the word "Handel" (trade) in the context of the trading sector, both of which are general terms unspecific to the class at hand.

Similar to many other lexical databases, OdeNet only stores the base form of a word. Therefore, a lemmatization step using HanTa [77] is executed prior to the synonym search.

Occasionally, a keyword may correspond to multiple senses and thus multiple synsets. In such instances, we select the synset that is most relevant to our class at hand. This relevance is calculated as a weighted average of the synset's cosine similarity to both the seed keywords and to the target word itself. More specifically, as shown in the equations below, with S representing the embeddings of the items in the synset, V representing the embeddings of the seed keyword, and W representing the embedding of the word from which we want to generate synonyms, the cosine similarity between the synset embeddings S and the seed embeddings V is given as:

$$\text{cos\_sim}(S, V)$$

and its mean is represented by:

$$\bar{c}_{SV} = \text{mean}(\text{cos\_sim}(S, V)).$$

Similarly, the cosine similarity between the S and the word embedding W is:

$$\text{cos\_sim}(S, W)$$

and its mean is represented by:

$$\bar{c}_{SW} = \text{mean}(\text{cos\_sim}(S, W)).$$

We set the weights of $\bar{c}_{SV} : \bar{c}_{SW}$ to be 3 : 1. The final score of a synset s can be represented as a weighted average using Equation 4.3:

$$Score_s = \frac{3 \times \bar{c}_{SV} + \bar{c}_{SW}}{4} = \frac{3 \times \text{mean}(\text{cos\_sim}(S, V)) + \text{mean}(\text{cos\_sim}(S, W))}{4}. \quad (4.3)$$

Only the synset with the highest score is deemed suitable for the keyword. Since we already selected the most relevant synset, and given the inherent high semantic similarity of synonyms to their target word, a further filter for the generated synonyms did not seem necessary. Therefore, the selected synonym set will be added into the class's keyword set.

In addition to the German OdeNet, our pipeline also supports English synonym generation using the Princeton Wordnet [53] from the NLTK corpus.[12]

4.6.3. Word Form Generation

all forms of a word, rather than solely the base form, is essential for our purpose to build a comprehensive keyword set for each class. Moreover, as the next phase in our project pipeline involves extracting meaningful context windows from the finalized keyword set (Chapter 1), having all word forms available increases the potential to discover more of these context windows, ultimately contributing to a more thorough and comprehensive analysis.

For an English corpus, generating word forms can be easily achieved with the `word-forms` package. For German, however, due to a lack of similar tools, we are essentially limited to word inflection and cannot generate word forms for other parts of speech. We divided the inflection process into two phases: one for nouns and another for adjectives.

We used the `german-nouns` package to inflect all nouns it recognizes within each class's keyword set. To streamline the adjective inflection process and avoid redundant computations, we created a derivative set from the class's keyword set by removing all recognized nouns. As German adjectives have five fixed endings—"-er", "-en", "-em", "-es" and "-e", we built a simple adjective declension algorithm for this purpose. This involves initially identifying and lemmatizing the adjectives before appending the endings to the lemmas. For the lemmatization step, we experimented with both SpaCy [28] and HanTa [77], ultimately settling on a combination of both, henceforth referred to as the *combined tagger*.

HanTa is distinguished by its capability to not only tag parts of speech but also to decompose words into their constituent morphemes [77]. For example, when presented with the word "agrarwirtschaftlichen", HanTa provides the following analysis:

[12] https://www.nltk.org/howto/wordnet.html

```
('agrarwirtschaftlich',
[('agrar', 'NN'), ('wirtschaftlich', 'ADJ'), ('en', 'SUF_ADJ')],
'ADJ(A)')
```

This decomposition identifies "agrarwirtschaftlich" as the lemma and breaks it down into its morphemes "agrar", "wirtschaftlich", and the suffix "en" with their respective POS tag. It also identifies the word as an adjective.

In German, nouns are always capitalized, a unique linguistic feature that distinguishes it from many other languages. This capitalization can assist in better identifying parts of speech, especially when distinguishing between nouns and adjectives. Given that our initial keyword extraction yielded lowercased outcomes, we created a supplementary keyword list where all keywords begin with an uppercase letter. Our combined tagger accepts a boolean parameter, `capital`, which determines whether the POS tagging and lemmatization results should stem from the capitalized version in case it differs from the uncapitalized one. By default, `capital` is set to `True`, indicating preference for capitalized words. The rationale behind this default setting, based on our experiments, will be elaborated upon in Subsection 5.3.3. To derive the lemmas of adjectives, which are essential for the upcoming declension process, we engaged the combined tagger. We first analyzed both the uncapitalized and the capitalized keyword list with HanTa, which can result in the following cases:

Case 1: If in both casings the word is identified as an adjective, we acknowledge it as a genuine adjective. We then use its identified lemma of the casing according to the parameter `capital` and return this lemma.

Case 2: If the analysis of the `capital` casing breaks down the word into more than two morphemes, those results are stored. If it does not offer a list of morphemes, but the other casing's analysis does, those outcomes are stored.

Case 3: If neither Case 1 nor Case 2 play out, store the POS and lemma according to the casing specified by `capital`.

If the word is identified as an adjective in the second or third scenario, the stored lemma is the final output. Otherwise, SpaCy's tagger is employed to offer additional insight. Specifically, we let SpaCy generate the lemma and the POS tag based on the lemma we previously obtained from HanTa, either in its capitalized or uncapitalized form depending on the `capital` argument. Using the `de_core_news_sm` pipeline from SpaCy, if it tags a word as an adjective or adverb and provides a non-capitalized lemma, that word is confirmed as an adjective and its generated lemma is returned as output. Illustrative results of this process can be found in Subsection 5.3.3. If, after all these steps, a word is not tagged as either an adjective or adverb, the algorithm outputs a null value.

After this lemmatization step, we obtain a list of proper adjective lemmas. These are then inflected by simply appending the aforementioned endings. In the end, we add all the resulting word forms from both noun and adjective inflections into the class's keyword set, marking the completion of our pipeline.

4.7. Evaluation

4.7.1. Expert Validation of Extracted Keyword Sets

In line with our last research question regarding the validation of our modified approach by domain experts in terms of the representativeness of the keyword sets, we crafted a detailed evaluation to assess the relevance and accuracy of our extracted keywords through expert judgment.

We selected five classes for our evaluation: A, B, M, P, and S. They were chosen based on specific criteria, a primary factor being the total number of keywords extracted from each class. These classes span a spectrum, from those with the largest keyword sets to the smallest, while also including classes from the mid-range. Such a varied selection strategy ensured a balanced perspective, preventing any bias towards classes with inherently richer or sparser keyword sets.

In addition to the criterion of keyword set length, classes M and S were intentionally selected due to their unique characteristics. The description of class M offers a vague outline, emphasizing high levels of education and expertise but lacks specific examples. This ambiguity underscored the need of seed keywords to guide accurate and class-specific extraction. Similarly, while Class S has the fewest extracted keywords, it acts as a residual category, encompassing activities not clearly categorized elsewhere. Its broad description necessitated seed keywords to ensure specificity. Given the challenges posed by these classes, they provided a robust testing ground for our approach.

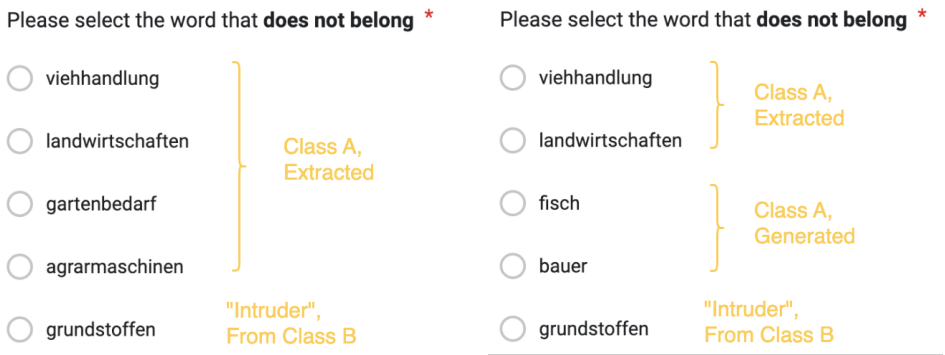
For the evaluation, domain experts were presented with a subset of the filtered extracted keywords for these classes. The filtered extracted keywords were obtained after the application of the damping function (Equation 4.2) as detailed in Subsection 4.5.8. From this filtered set, we created a subset by intentionally selecting every fifth keyword, ranked by their descending final similarity scores. This resulted in an average of 18 keywords per class, which were then presented to the domain experts. Our decision to select keywords in such a systematic manner, rather than using random sampling, was driven by the intent to ensure that these keywords captured a broad spectrum of quality and class-specificity, thus preventing any potential over-representation from a narrow score range. Alongside this subset, the experts were also given short class descriptions and seed keywords to offer context. Their task was to assess these presented keywords, identifying any that did not align with the class's core theme. The complete instructions and questions of this evaluation task are detailed in Appendix B.1.

4.7.2. Intruder Detection

We also conducted a second evaluation with an intruder detection task, in which the same five classes (A, B, M, P, and S) were chosen as in the previous evaluation approach. For each class, respondents faced five questions, where they had to identify the "intruder" keyword—the one they think does not belong or is least related to the others. Each of the five questions contained four random keywords from the selected class and one intruder. The intruder in the first four questions for each class was a random key-

word from one of the other four classes, while the last question contained one random keyword from a class not among the selected five classes.

This task was performed for two different keyword sets: one comprised solely of extracted keywords, and another with both extracted and generated keywords. For each question pertaining to the *extracted-only* set, participants were presented with four filtered extracted keywords (after applying the damping function in Subsection 4.5.8) from the designated class, and one keyword not originating from that class. For the *extracted and generated* set, we made a deliberate choice; we replaced two of the previously mentioned extracted keywords with two generated ones, keeping the other options consistent with the *extracted-only* set. This decision was taken to control for variables, ensuring the results between the two sets were more comparable and measurable. Figure 4.8 shows an example of the question design for the two different sets. The aim of the intruder detection task was twofold: first, to determine the coherence and class-specificity in the extracted keywords, and second, to assess whether the inclusion of generated keywords in the keyword sets influenced the perceived coherence. Since our data set was in German, this survey targeted individuals with a substantial knowledge of the German language, either native German speakers or those who have completed their education in German. To minimize any potential order effect, the sequence of questions and the order of options within each question were randomized for each participant. A detailed list of all the keywords used in the questions of this survey is available in Appendix B.2.



(a) Example question for *extracted-only* set

(b) Example question for *extracted and generated* set

Figure 4.8.: Example questions in the intruder detection task for different keyword sets (class A). Annotations within the figures specify keyword origins.

4.7.3. Automatic Evaluation

In addition to qualitative evaluations, we employed an automatic evaluation approach to assess the keyword relevance across each stage in relation to the original seed keywords. We first systematically applied our automatic evaluation on individual sets of keywords from each stage. Subsequently, this evaluation was extended to collective sets

of keywords in order to obtain a holistic view of the combined results. Specifically, the following sets of keywords were assessed for each class:

1. Extracted keywords
2. New keywords generated in lexical substitution
3. New keywords from synonym generation
4. New keywords from word form generation
5. All new keywords from the generation stage (combining stages 2-4)
6. The entire set of keywords (combining stages 1-4)

For each keyword within a specific set, we computed its score using both the *mean seed* and *max seed* scoring metrics (Subsection 4.5.7). The final score of the keyword was determined by averaging these two metrics. Subsequently, we calculated the class average of these resultant scores across each of the 21 classes, allowing us to arrive at an overarching average for all classes.

5. Results

In this chapter, we present the outcomes of our proposed pipeline. We begin by detailing the keyword extraction results, followed by insights from our generation process. Concluding the chapter, we provide a holistic statistical overview of our entire workflow and delve into the evaluation results.

5.1. Preprocessing Results

Following the preprocessing methodologies outlined in Section 4.4, this section provides a detailed record of the results obtained.

5.1.1. Defined Class Descriptions

Commencing our preprocessing, the initial focus was on defining class descriptions using the different models for summarization and paraphrasing.

The extractive `t5-base` model predominantly selected a few sentences from the beginning of the original section description in WZ2008. The results from the `google/flan-t5-base` model, on the other hand, raised more concerns. Despite its design catered towards the German language, the model frequently generated summaries from a German text that were either partially or entirely in English. Figure 5.1 shows some of these suboptimal summaries by comparing them with those generated by the `t5-base` model for the same classes.

Abschnitt	Name	Beschreibung	summary_flan-t5	summary_t5
A	Land- und Forstwirtschaft, Fischerei	Dieser Abschnitt umfasst die Nutzung der pflanzlichen und tierischen natürlichen Ressourcen. Dazu zählen Tätigkeiten wie Pflanzenbau, Tierzucht und Tierhaltung, Holzgewinnung und die Gewinnung anderer pflanzlicher und tierischer Erzeugnisse in land- oder forstwirtschaftlichen Betrieben oder in freier Natur.	This section covers the use of plant and teeming natural resources, including activities such as plant cultivation, horticulture and forestry, wood harvesting and the sale of other plant and tieric products in land- or forested farms or in free-natural nature.	Dieser Abschnitt umfasst die Nutzung der pflanzlichen und tierischen natürlichen Ressourcen. Dazu gehören Tätigkeiten wie Pflanzenbau, Tierzucht und Tierhaltung, Holzgewinnung und die Gewinnung anderer pflanzlicher und tierischer Erzeugnisse in land- oder forstwirtschaftlichen Betrieben oder in freier Natur.
D	Energieversorgung	Dieser Abschnitt umfasst die Elektrizitäts-, Gas-, Wärme- und Warmwasserversorgung u. Ä. durch ein fest installiertes Netz von Strom- bzw. Rohrleitungen. Der Umfang des Netzes ist nicht entscheidend. Eingeschlossen ist auch die Versorgung von Industrie- und Gewerbegebieten, sowie von Wohngebäuden. Unter diesen Abschnitt fällt daher der Betrieb von Anlagen, die Elektrizität oder Gas erzeugen und verteilen bzw. deren Erzeugung und Verteilung überwachen. Ebenfalls eingeschlossen ist die Wärme- und Kälteversorgung. Dieser Abschnitt umfasst nicht: – Betrieb von Wasser- und Abwasseranlagen (s.Abteilungen 36 und 37, Abschnitt E) – Transport von Gas in Rohrfernleitungen (s.49.50.0, Abschnitt H)	– Operation of water and wastewater facilities (s. Abteilungs 36 and 37, Section E) – Transport of gas in pipelines (s. 49.50.0, Section H)	Dieser Abschnitt umfasst die Elektrizitäts-, Gas-, Wärme- und Warmwasserversorgung u.. durch ein fest installiertes Netz von Strom- bzw. Rohrleitungen. Eingeschlossen ist auch die Versorgung von Industrie- und Gewerbegebieten, sowie von Wohngebäuden.
E	Wasserversorgung; Abwasser- und Abfallentsorgung und Beseitigung von Umweltverschmutzungen	Dieser Abschnitt umfasst Tätigkeiten im Zusammenhang mit der Entsorgung (Sammlung, Behandlung und Beseitigung) verschiedener Abfälle, wie z. B. fester oder nicht fester Abfälle aus Industrie, Gewerbe oder Haushalten, sowie die Sanierung von Altlasten. Die Endprodukte der Abfall- oder Abwasserbehandlung können entweder beseitigt oder neuen Produktionsprozessen zugeführt werden. Auch Tätigkeiten der Wasserversorgung fallen unter diesen Abschnitt, da sie häufig entweder in Verbindung mit der Abwasserbehandlung durchgeführt werden oder von Einheiten erbracht werden, die auch mit der Abwasserbehandlung befasst sind.	This section includes activities related to the disposal (samming, treatment and removal) of various wastes, such as industrial, commercial or household wastes, as well as the disposal of wastes. The end products of waste or wastewater treatment can either be removed or new production processes carried out.	Dieser Abschnitt umfasst Tätigkeiten im Zusammenhang mit der Entsorgung (Sammlung, Behandlung und Beseitigung) verschiedener Abfälle, wie z. B. fester oder nicht fester Abfälle aus Industrie, Gewerbe oder Haushalten.

Figure 5.1.: Comparison between the summaries produced by `google/flan-t5-base` and `t5-base` on three sample sections (classes) A, B, and D. `Beschreibung` is the original section description in the WZ2008 classification. Texts that are exactly the same as in the `Beschreibung` are highlighted.

Seeking a more refined approach to formulating section descriptions beyond merely extracting sentences, we explored various abstractive summarization models. The results, however, were unsatisfactory. `facebook/bart-large-cnn` often cut German texts prematurely, producing incomplete sentences before reaching the specified maximum token length. For example, when given the description of section E (Figure 5.1), the model generated the following incomplete summary:

```
Dieser Abschnitt umfasst Tätigkeiten im Zusammenhang mit der
Entsorgung (Sammlung, Behandlung und Beseitigung) verschiedener
Abfälle. Die Endprodukte der Abfall- oder Abwasserbehandlings
können entweder beseitIGt oder neuen Produktionspro
```

Models like `Shahm/bart-german`[1] and `Einmalumdiewelt/T5-Base_GNAD`[2] often failed to include information that we qualitatively deemed as the most crucial part of the text. For instance, the former produced the following summary for section E, in which the most essential sentence describing the section—the first sentence in the original description—was left out:

```
Die Endprodukte der Abfall- oder Abwasserbehandlung können
entweder beseitigt oder neuen Produktionsprozessen zugeführt werden.
```

The latter, when given the description of section D (Figure 5.1), generated only one sentence:

```
Die Elektrizitätsversorgung u.. ist ein fest installiertes Netz
von Stromleitungen bzw. Rohrleitungen.
```

Furthermore, the model `bart.large.xsm` from fairseq[3] has a tendency to concatenate unrelated phrases from different sentences, leading to incoherent summaries. For section D, it fused the last part `Strom und Kälte` from another sentence, creating an illogical output:

```
Dieser Abschnitt umfasst die Elektrizitätsversorgung, Gasversorgung,
Wärmeversorgung und Warmwasserversorgung u.a. durch ein fest
installiertes Netz von Stromleitungen bzw. Strom und Kälte.
```

Due to these unsatisfactory results from abstractive summarization models, we shifted our search to paraphrasing tools. Unfortunately, we did not find any paraphrasing model trained for German, compelling us to experiment with English models such as `prithivida/parrot_paraphraser_on_T5`, `humarin/chatgpt_paraphraser_on_T5_base`, and `Vamsi/T5_Paraphrase_Paws`.[4] Regrettably, none of them is completely suitable for German. Specifically, the first model from `prithivida` simply returned the input text,

[1] https://huggingface.co/Shahm/bart-german
[2] https://huggingface.co/Einmalumdiewelt/T5-Base_GNAD
[3] https://github.com/facebookresearch/fairseq/blob/main/examples/bart/README.md
[4] https://huggingface.co/Vamsi/T5_Paraphrase_Paws

while the latter two both mixed German with English in the output. For example, when given the section description for A (Figure 5.1), the outputs were as follows, with the English portions italicized:

humarin/chatgpt_paraphraser_on_T5_base:
Dieser Abschnitt behandelt die Nutzung der natürlichen Ressourcen *such as* Pflanzenbau, Tierhaltung, Holzgewinnung, *and other* pflanzlicher *and* tierischer Erzeugnisse in land- *or* forstwirtschaftlichen Betrieben oder in freier Natur.

Vamsi/T5_Paraphrase_Paws:
Dieser Abschnitt umfasst die Nutzung *of* pflanzlichen *and* tierischen *natural resources*. Dazu gehören *activities such as plant planting, animal tot - and* tierhaltung, *and the harvesting of other* pflanzlicher *and* tierischer erzeugnisse in land- *or* forstwirtschaftlichen Betrieben *or* in freier Natur.

Therefore, we omitted abstractive summarization and paraphrasing due to the lack of high-quality German models. Instead, we proceeded with `t5-base` to generate solely extractive summaries.

5.1.2. Extracted Seed Keywords from Class Description

When extracting five nouns as seed keywords from the class description, we prefixed the class names and appended five manually defined keywords to the summarized class description. This addition proved to be necessary, as the extracted seed keywords without this added context contained mostly generic keywords. For example, as seen in Figure 5.2, without the additional information, the extracted keywords for class E are mainly generic terms such as "industrie" (industry), "behandlung" (treatment), and "tätigkeiten" (activities).

Nevertheless, the problem of generic keywords still persisted even after this addition, as can be seen in class M with the term "erbringung" (provision) (Figure 5.2). To address this issue, our initial strategy involved adding some generic keywords as stopwords, ensuring their exclusion during the extraction. However, anticipating every potential generic keyword is challenging, making the list inherently incomplete. Moreover, this process requires massive manual identification of generic keywords, a step that contradicts our goal of creating an automated pipeline with minimal manual intervention.

Employing outlier detection here was not plausible either, as five keywords are too small for a sample set. Furthermore, highly generic keywords are less likely to be viewed as outliers. However, we observed that the extracted seed keywords for different classes could contain duplicates, and these duplicates were frequently generic terms like "tätigkeiten" (activities) and "erbringung" (provision). Therefore, we simply removed these duplicate terms as a postprocessing step to eliminate generic keywords.

Section	Summarized Text with Additional Information	Keywords *Without* Additional Information	Keywords *With* Additional Information
E	Wasserversorgung; Abwasser- und Abfallentsorgung und Beseitigung von Umweltverschmutzungen. Dieser Abschnitt umfasst Tätigkeiten im Zusammenhang mit der Entsorgung (Sammlung, Behandlung und Beseitigung) verschiedener Abfälle, wie z. B. fester oder nicht fester Abfälle aus Industrie, Gewerbe oder Haushalten. Dazu gehören: Wasserversorgung, Abwasserentsorgung, Abfallentsorgung, Umweltschmutzungen, Sammlung von Abfällen	(entsorgung', 0.5351), ('**industrie**', 0.4991), ('abfälle', 0.4065), ('**behandlung**', 0.395), ('**tätigkeiten**', 0.383)	('abwasserentsorgung', 0.7632), ('abfallentsorgung', 0.7622), ('wasserversorgung', 0.6496), ('entsorgung', 0.5254), ('umweltverschmutzungen', 0.4616)
M	Erbringung von freiberuflichen, wissenschaftlichen und technischen Dienstleistungen. Dieser Abschnitt umfasst bestimmte freiberufliche, wissenschaftliche und technische Tätigkeiten. diese Tätigkeiten erfordern ein hohes Maß an Ausbildung und stellen den Nutzern Fachkenntnisse und Erfahrungen zur Verfügung. Dazu gehören: Rechtsberatung, Steuerberatung, Wirtschaftsprüfung, Buchführung, Unternehmensberatung.	('nutzern fachkenntnisse', 0.5579), ('**tätigkeiten**', 0.4), ('**verfügung**', 0.3785), ('**erfahrungen**', 0.3631), ('freiberufliche', 0.3285)	('wirtschaftsprüfung', 0.4996), ('rechtsberatung', 0.4592), ('unternehmensberatung', 0.4495), ('freiberufliche', 0.4134), ('**erbringung**', 0.4063)
R	Kunst, Unterhaltung und Erholung. Dieser Abschnitt umfasst Tätigkeiten, die die verschiedenen kulturellen, Unterhaltungs- und Freizeitinteressen der breiten ffentlichkeit abdecken, einschließlich Durchführung von Liveauftritten, Betrieb von Museen, Spiel-, Wett- und Lotteriewesen, sportliche und Freizeitaktivitäten. Dazu gehören: Kunst, Unterhaltung, Erholung, darstellende Kunst, Freizeitaktivitäten.	('museen', 0.516), ('freizeitaktivitäten', 0.5042), ('freizeitinteressen', 0.4259), ('liveauftritten', 0.3804), ('**tätigkeiten**', 0.3633)	('kunst', 0.636), ('museen', 0.555), ('unterhaltung', 0.4497), ('erholung', 0.4286), ('freizeitaktivitäten', 0.4002)

Figure 5.2.: Comparison of extracted keywords from the summarized class description with and without the inclusion of class names and five manually defined seed keywords. The added information in each class description is highlighted, and the bold text represents generic keywords that are not specific to any class.

5.1.3. String Matching Outcomes

Initially, we employed fuzzy string matching for the seed keywords, implying that any entry containing the string used in the seed keywords would be considered a match. This method, however, led to unintended matches where the seed keywords constituted a substring within a larger, unrelated word. For example, a seed keyword "Erz" (ore) is specific to the class on mining. However, it also emerged as a substring in the word "*Erz*eugnis" (product), a generic term that could appear in virtually any class. We therefore transitioned to exact string matching in order to ensure that no seed acts as a mere substring within another word in a data set entry. While this method enhanced precision, it had its limitations. For example, an entry containing the word "landwirtschaftlich" would not be recognized if our seed was solely "Landwirtschaft". However, our primary goal at this stage was to obtain a precise set of matched entries to compute a baseline for the similarity scores. We thus deemed it more important to have fewer accurate matches than an excess number of incorrect ones that could skew the similarity scores due to their semantic unrelatedness.

5.1.4. Baseline Selection and Similarity Search Results

One recurring issue across all models employed for similarity comparison was the low similarity scores attributed to entries highly relevant to the respective class. Specifically, entries containing at least one seed keyword generally proved to be relevant matches for the intended class, regardless of their similarity scores to the class description. Taking class L on "Grundstücks- und Wohnungswesen" (real estate and housing) as an instance,

the deduplicated extracted keywords were:

gebäuden, wohnungen, grundstücken, wohnungswesen.

When applying exact string matching on the data subset using these keywords, and subsequently computing the similarity between the matched entries and the summarized class description, we obtained the following percentiles with the `symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli` model:

25th percentile: 0.4858
50th percentile: 0.6141
75th percentile: 0.7100

However, many entries below the 25th percentile were also highly relevant to the class and could provide useful information for future keyword extraction, as shown in Figure 5.3. Therefore, we initially set the 25th percentile as the baseline for the subsequent similarity search. Any entry in the data set with a similarity score above this was deemed suitable for that class's keyword extraction.

legal_name	purpose	similarity_score
Projektgesellschaft Pasteurstraße Haus 4 mbH	ankauf, bebauung und verwaltung von grundstücken, insbesondere an der pasteurstraße in münchen. geschäfte im sinne des § 34c gewerbeordnung werden nicht durchgeführt.	0.088771
IC Objekt6 Berlin GmbH	erwerb, die verwaltung, vermietung, veräußerung und sonstige verwertung von grundstücken und immobilien in deutschland. die gesellschaft betreibt keine geschäfte im sinne von § 34c abs. 1 satz 1 gewo.	0.127798
Viktoriapark Grundbesitz GmbH	der an- und verkauf von grundstücken und grundstücksgleichen rechten. eine tätigkeit nach § 34 c gewo wird nicht ausgeübt.	0.184854
TIPP Reinigungsdienste West GmbH	die erbringung von dienstleistungen im bereich von reinigung von gebäuden und gebäudebestandteilen einschließlich außenanlagen, insbesondere für kliniken und unternehmen der helios kliniken gruppe.	0.209362
Antonius Immobilien GmbH	das halten und die verwaltung eigener grundstücke, miteigentumsanteilen an grundstücken sowie anteilen an personen- und kapitalgesellschaften. es werden keine tätigkeiten nach § 34 c der gewerbeordnung ausgeübt. die gesellschaft darf nur vermögensverwaltend tätig sein.	0.220721
VS.PLAN GmbH Planung + Projektsteuerung im Bauwesen	architektur, baubetreuung, projektsteuerung im hochbau und vermarktung von grundstücken, tätigkeiten die der genehmigung nach § 34 c der gewerbeordnung, nach der handwerksordnung oder nach sonstigen gesetzen bedürfen, werden nicht ausgeübt;	0.226623
RK Objektausbau GmbH	gegenstand geändert; nun: objektausbau, umbau, sanierung und modernisierung von gebäuden sowie der trockenbau.	0.261856

Figure 5.3.: Examples from class L: business registry entries that contain at least one of the keywords but have similarity scores to the class description below the 25th percentile (0.4858) using the model `symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli`. The exact string matches of the keywords are highlighted.

When testing this baseline on the entire data subset, we observed a significant discrepancy in its effectiveness. While the 25th percentile baseline was apt for entries containing at least one seed keyword, it was overly inclusive for entries without any seed keywords, leading to a large number of false positives. This is shown using the example of class L in Figure 5.4.

To mitigate this issue, we increased the baseline to the 50th percentile. However, while a few classes yielded good results, a significant number of false positives persisted in the majority of the classes. For example, class I is on the hospitality industry, which

legal_name	purpose	similarity_score
SH-Biomasse GmbH	Die Erstellung und der Betrieb von Anlagen zur Erzeugung von Energie aus nachwachsenden Rohstoffen (Biomassen), die Vermietung derartiger Anlagen und der Verkauf von hieraus erzeugten Energien.	0.486632
The Home Habit Shop UG (haftungsbeschränkt)	Verkauf von Ordnungsprodukten, digitalen Produkten und Templates sowie Labels.	0.486569
Grundstücksverwaltungsgesellschaft Wittener Straße mbH	Gegenstand geändert; nun: Der Erwerb und die Verwaltung von bebauten und unbebauten Grundstücken, insbesondere des Grundstücks Wittener Straße 6-12 in Mannheim.	0.485992
Hilmar Frank Licht Concepte GmbH	1) Betrieb einer Handelsagentur für Leuchten und Elektroartikel. 2) Betrieb eines Planungsbüros zur Objektberatung und Projektierung von Hoteleinrichtungen, Ladenbauten und Messebauten. 3) Betrieb eines Groß- und Einzelhandels für Leuchten und Elektroartikel.	0.485976
Cappel & Cie.Nachfolger Peter Conrad Eisenwaren und Hausrat GmbH	Vertrieb im Groß- und Einzelhandel von Haushaltswaren aller Art sowie von Werkzeugen und Schreinereibedarf, von Eisenwaren und Flachglas, von Kleinmöbeln, Badeinrichtung und Elektroartikeln, Farben und Porzellanwaren sowie von Geschenkartikeln.	0.485932
Recycling Kall GmbH	Aufbereitung von Altmaterial (recycling), insbesondere für den Straßenbau. Die Gesellschaft ist befugt, andere Unternehmen zu erwerben oder sich an solchen zu beteiligen.	0.485896

Figure 5.4.: Examples from class L when using the 25th percentile (0.4858) as baseline: business registry entries with similarity scores above the baseline but do not belong to class L. Sentence similarities are computed using the model `symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli`.

includes short-term accommodation like hotels and gastronomy services. As shown in Figure 5.6, most of the entries between the 50th and 75th percentile were not specific to the class and should be excluded. Therefore, we ultimately increased the baseline further to the 75th percentile. This adjustment significantly improved the precision of the results, leading to a higher proportion of true positives. However, a notable trade-off was an increase in false negatives, meaning that some useful and relevant results below the baseline were excluded. This is evidenced in Figure 5.5 by class L, where many valid class-specific matches between the 50th and 75th percentile were excluded from further use.

The similarity scores in the provided examples were calculated using `symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli`. Since these observations were consistent across all the models employed, we have refrained from repetitively showcasing each instance.

It should be noted that the absence of a standardized ground truth for our data set necessitated a qualitative assessment to establish a reference standard for the performance of our model. In the search for data set entries with similarity scores above the baseline, this assessment involved manually reviewing a scored entry and categorizing it as true positive, true negative, false positive, or false negative based on their relevance to the class description. A false positive is an entry scored above the baseline but irrelevant to the class upon manual inspection, whereas a false negative is an entry which scored below the baseline but should be deemed class-specific and included in subsequent keyword extraction and generation stages.

5.1.5. Preprocessing Reevaluated: Rationale for Omission

We decided to omit the preprocessing step upon the realization that its limitations outweighed its benefits, mainly due to the following reasons:

legal_name	purpose	similarity_score
AMI Immobiliengesellschaft UG (haftungsbeschränkt)	An- und Verkauf von Grundstücken und Immobilien.	0.850374
Triebe & Bleil Immobilien GmbH	Erwerb von Immobilien aller Art, die Verwaltung, Vermietung und Verpachtung von Immobilien, der Erwerb und die Verpachtung von Mobilien sowie die Erschließung und Vermittlung von gewerblichen Standorten und die Geschäftsführung bei ähnlichen oder gleichartigen Gesellschaften.	0.848633
Weingarten Projekt GmbH	der An- und Verkauf von Immobilien und Grundstücken.	0.846071
hagema Immobilien GmbH	Kauf und Verwaltung von Grundstücken und Gebäuden im In- und Ausland sowie die Vermietung.	0.834686
...	...	...
BBW-124-130 Grundstücks GmbH	Die Entwicklung von Immobilienprojekten sowie der An- und Verkauf von Grundstücken und grundstücksgleichen Rechten sowie sämtliche damit im Zusammenhang stehende Tätigkeiten.	**0.71**
...	...	...
helifly Flugplatzhallen GmbH	Vermietung und Verpachtung von Grundstücken, insbesondere Flugplatzhallen.	0.619829
Pro Immobilia Limited	Vermittlung von Immobilien, Darlehens- und Versicherungsverträgen, gewerbliche Zwischenvermietung, Vermittlung und Veranstaltung von Firmenevents, Seminaren, privaten Feiern und Kochkursen.	0.619764
Silberbaum Industrial GmbH	Die Bebauung von eigenen Grundstücken und die Vermietung und Verwaltung eigener Immobilien unter Ausschluss von Tätigkeiten im Sinne des § 34c Gewerbe- Ordnung.	0.619228
I&B GmbH	Erwerb, Halten, Verwalten, Bebauen und Veräußern von Grundstücken sowie die Sanierung von Gebäuden. Tätigkeiten nach § 34 C Gewerbeordnung sind ausgenommen.	0.618753
C & B Immobilienverwaltung GmbH	Die Immobilienverwaltung, Ankauf und Verkauf von Immobilien, Vermittlung von Immobilien, Immobilienfinanzierung und Vermittlung von Immobilienfinanzierungen, Reinigungsservice und Hausmeisterservice.	0.61804

Figure 5.5.: Examples from class L (real estate and housing) with 50th percentile 0.6141 and 75th percentile 0.71. The entries above the 75th percentile are highly class-specific. However, many entries between the 50th and 75th percentile are also good matches for the class. Similarity scores computed using `symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli`.

legal_name	purpose	similarity_score
PP-Hotels UG (haftungsbeschränkt)	der Hotelbetrieb und das Gaststättengewerbe.	0.718236
SH Mehra Hotels GmbH	Betreiben von Hotels und Pensionen, die Beherbergung von Personen und alle damit zusammenhängenden Tätigkeiten und der Betrieb von Restaurants und Gaststätten.	0.690916
Makowski Hotel Service Kiel UG (haftungsbeschränkt)	Service und die Dienstleistungen für Hotel & Gaststätten und sonstigen Gewerbebetrieben	0.672952
...	...	...
Lode GmbH	sämtliche Leistungen des Rohrreinigungsgewerbes sowie die Zimmervermietung	0.464145
Hausverwaltung MF GmbH	Verwaltung von Haus- und Grundbesitz.	0.463527
Simple Travel GmbH	Vermittlung und Veranstaltung von Reisen	0.463389
" Haus Maar " Hotel Gesellschaft mit beschränkter Haftung	die Führung eines Hotelbetriebes;	0.463034
...	...	...
Meson el toro GmbH	Der Betrieb eines Restaurants und das Betreiben einer Weinhandlung	0.411332
Transporte Schwarzbauer UG (haftungsbeschränkt)	Die Erbringung von Dienstleistungen, insbesondere Transportdienstleistungen, medizinische Transporte, Kurierfahrten und vergleichbare Tätigkeiten.	0.411273
...	...	...
SAFRA GmbH	Der Erwerb, die Bebauung und Verwaltung von Immobilienvermögen.	0.342906
Raumspiel Vertriebsgesellschaft mbH	Handel mit Möbeln, Accessoires, Lampen, Stoffen, Farben und Spielzeug sowie Dienstleistungen im Bereich Einrichtung.	0.342894

Figure 5.6.: Examples from class I (hospitality industry) with 50th percentile 0.3429 and 75th percentile 0.4139. Entries between the 50th and 75th percentile are, in general, no good matches for the class. Similarity scores were computed using `symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli`.

1. Comparing the sentence similarity between the class description and the data set entries did not yield satisfactory results. As exemplified in Figure 5.5, although the precision was high above the 75th percentile baseline, there were many false negatives. This indicates that our current similarity measures might not fully reflect the nuanced relationship between the class descriptions and the individual data set entries.

2. Determining an appropriate baseline was a challenge. While the results above the 75th percentile were primarily true positives, the 50th percentile was also a fitting baseline for certain classes, indicating that setting the threshold was inherently class-specific. A high baseline would inadvertently exclude many relevant entries, while a lenient baseline might flood the results with false positives, undermining the purpose of preprocessing.

3. Limiting the entries to those that surpassed the baseline, as opposed to utilizing the 10,000 data subset, resulted in a significant reduction in the number of keywords extracted for each class, as seen with the examples in Table 5.1. Moreover, there was an evident variation in the number of entries and keywords across classes, as depicted in Figure 5.7. On average, only 1,931 of the 10,000-entry subset entries surpassed the baseline for each class, ranging from a minimum of 37 to a maximum of 4,866. As a result, only an average of 1,039 keywords were extracted per class, with a span from a minimum of 135 to a maximum of 3,283. This preprocessing step might have thus inadvertently omitted a substantial number of potential keywords.

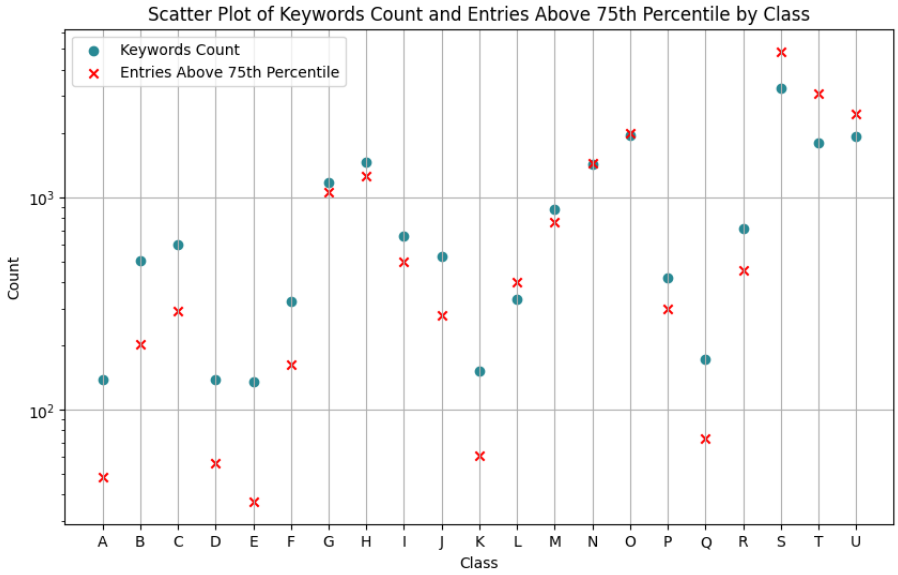


Figure 5.7.: Number of entries above the 75th percentile and their corresponding extracted keyword count after preprocessing, organized by class

Despite these challenges, it is worth mentioning that although we omitted the preprocessing step in our approach, it could yield better results under enhanced conditions, such as more consistent class descriptions and better similarity comparison metrics. This exploration, however, lies beyond the scope of the current study and is recommended for future research.

5.2. Extraction Results

In this section, we present the results of our keyword extraction process corresponding to the strategies and hyperparameters detailed in Section 4.5. It is important to note that the extraction algorithm processes all text in lowercase. As a result, even terms that are typically capitalized, like German nouns, will be presented in lowercase. This intentional representation is a direct result of our extraction method, not a typographical oversight.

5.2.1. Comparative Model Analysis

In our qualitative comparison between the models `sentence-transformers/distiluse-base-multilingual-cased-v1` (hereafter DBMC) and `deutsche-telekom/gbert-large-paraphrase-cosine` (hereafter Telekom), several key differences became apparent. Overall, the Telekom model demonstrated superior performance. Utilizing the *mean seed* scoring method without iterations, DBMC often yielded less accurate and class-specific keywords. To illustrate, Figure 5.8 contrasts their performance on class A (forestry, agriculture and fishing) when given the following seed keywords:

> *Anbau von Pflanzen, Aquakultur, Fallenstellerei, Fischerei, Fischzucht, Forstwirtschaft, Holzeinschlag, Holzgewinnung, Jagd, Landwirtschaft, Pflanzenbau, Tierhaltung, Tierzucht, Veredlung landwirtschaftlicher Erzeugnisse.*

Moreover, DBMC showed a propensity to favor keywords with morphological similarity to the seeds, especially evident when using the *max seed* scoring method. For instance, with DBMC, "Todes*fall*" (death case) received a high similarity score of 0.817 to the seed "*Fall*enstellerei" (trapping) due to the common morpheme "fall". This would result in less class-specific keywords, as morphological similarity does not always equate to semantic relevance, and semantic relevance does not always guarantee class-specificity in our case. A pertinent example could be found in class P on "Erziehung und Unterricht" (education and teaching). Using the Telekom model, the seed keyword "Flugschule" (flight school) led to the term "flugzeugen" (airplanes) achieving a high similarity score of 0.8309. Both terms share the morpheme "flug" and can be considered semantically related. However, "flugzeugen" is not specific to education and teaching. This example also showed that although the Telekom model sometimes overemphasized morphological similarity, it did so to a smaller extent than DBMC.

On the downside, the Telekom model demanded a significantly longer processing time. Compared to DBMC and the default `sentence-transformers/all-MiniLM-L6-v2`

model, which both completed the extraction from our 10,000-entry subset data in five to six minutes, the Telekom model required over 15 minutes, almost tripling the time. Despite this extended processing time, we ultimately proceeded with the Telekom model for the remainder of the extraction and generation stages, as the keyword quality was deemed more critical than the processing speed for our application.

```
('erwirtschaftete', 0.8713, 1)
('erbbaurechten', 0.8585, 7)
('bewirtschaftet', 0.8484, 7)
('frotteewaren', 0.846, 1)
('gezüchtet', 0.8346, 1)
('viehzucht', 0.83, 2)
('bewirtschaftung', 0.8292, 59)
('gebäudebewirtschaftung', 0.8282, 1)
('erbbaurecht', 0.8275, 1)
('erwerbsund', 0.8246, 1)
('erwerbs', 0.8221, 68)
('grabau', 0.8212, 1)
('tierhäuten', 0.8182, 2)
('baureparaturen', 0.8176, 2)
('lagerwirtschaft', 0.8169, 1)
```

(a) Top keywords from DBMC

```
('viehwirtschaft', 0.8335, 2)
('viehzucht', 0.818, 2)
('landwirtschaften', 0.8175, 1)
('landwirtschaftsgestaltung', 0.8046, 1)
('ackerbauund', 0.8016, 1)
('viehhandlung', 0.7789, 1)
('landwirtschaftlich', 0.7782, 3)
('landwirtschaftliche', 0.7697, 6)
('holzwirtschaft', 0.7648, 1)
('ackerbau', 0.7646, 1)
('landwirtschaftlicher', 0.7538, 19)
('agrar', 0.7535, 3)
('landwirtschaftlichen', 0.7527, 55)
('landwirtschaftlicheund', 0.7484, 1)
('pflanzenzucht', 0.7476, 2)
```

(b) Top keywords from Telekom

Figure 5.8.: Comparison of the top extracted keywords of class A (agriculture, forestry and fishing) using the DBMC and Telekom model. The keywords were extracted without any iteration and scored using the *mean seed* approach. The values in each tuple are the keyword, its similarity score, and the number of occurrences in the subset data.

5.2.2. Keyword Scope Analysis: Noun-Only and Unigram-Only

Our initial extraction was centered around nouns and noun phrases, facilitated by the `KeyphraseCountVectorizer`. However, a notable drawback was the extraction time. Utilizing the vectorizer, even on a GPU[5], extended the duration to more than two hours for one round on the data subset, a process likely prolonged due to issues with GPU optimization for the `KeyphraseCountVectorizer`.[6]

Interestingly, we observed that the majority of the extracted keywords through this *noun-only* approach were still unigrams. This aligns with the linguistic construct of the German language, which often combines potential noun phrases into singular compound words. An example from our extraction is "Landwirtschaftszubehör" (farming accessories), which is composed of the two nouns "Landwirtschaft" and "Zubehör".

Given this observation, we explored the alternative extraction scope without the vectorizer and solely focused on unigram extraction. This *unigram-only* approach was con-

[5]Throughout this thesis, we utilized a V100 Nvidia GPU for our experiments.
[6]https://github.com/MaartenGr/KeyBERT/issues/108

siderably faster, completing one extraction round within 15 minutes. Without a limit set on the part of speech, the extraction yielded predominantly nouns and adjectives. Upon qualitative assessments, we concluded that there was no significant difference in keyword quality between the two approaches. For visual clarity, Figure 5.9 shows a comparison between the top 35 keywords extracted using the two approaches for class A (agriculture, forestry, and fishing).

Opting for efficiency and the negligible disparity in keyword quality, our final choice was the *unigram-only* approach for the subsequent extractions. A notable aspect for both approaches was the recurrence of lemmatically identical words in various forms, such as "landwirtschaftlicher" and "landwirtschaftliche". This repetition was more common in the *unigram-only* approach as it included adjectives, which often appear in inflected forms. However, as word form generation is a subsequent stage in our pipeline, this repetition should not be a cause for concern.

5.2.3. Results from Different Extraction Strategies

Preprocessed extraction. As we are left with hundreds or sometimes only tens of matching entries after preprocessing, using the *preprocessed extraction* strategy yielded significantly fewer keywords. As shown in Table 5.1, the number of keywords extracted using the preprocessing step can be less than 10% compared to the count without preprocessing. While we cannot assert that all additional keywords extracted without preprocessing are class-specific, restricting ourselves to such a limited set undoubtedly leads to overlooking some potentially useful and relevant keywords. This was confirmed by our earlier findings in Subsection 5.1.4, where numerous entries with genuine class relevance received low similarity scores. As a result, these entries were excluded from the subsequent keyword extraction, despite their actual validity and relevance to the concerned class.

Class	#Entries Preprocessing	#Keywords Preprocessing	#Keywords, No Preprocessing
A	48	138	8,920
D	56	139	8,404
I	496	655	8,347
L	397	331	7,789

Table 5.1.: Comparison between the number of extracted keywords with and without preprocessing for classes A, D, I and L. *Entries preprocessing* denotes the number of unique entries above the baseline for the given class; *#keywords preprocessing* denotes the number of unique keywords extracted from those entries using the *preprocessed extraction* strategy, and *#keywords, no preprocessing* is the number of unique keywords extracted from the 10,000-entry data subset (using iteration and *average scoring* approach). Only unigrams were extracted.

```
('landwirtschaften', 0.8832, 1)
('landwirtschaftsgestaltung', 0.8821, 1)
('agrarflächen', 0.8433, 1)
('agrarbereich', 0.8388, 2)
('gartenbau', 0.8372, 7)
('landschaftspflege', 0.8342, 7)
('gartenbaubetrieb', 0.8307, 1)
('erwerbsgartenbau', 0.8305, 1)
('viehwirtschaft', 0.8288, 2)
('landschaftsbaus', 0.8253, 4)
('gartenarbeit', 0.8225, 1)
('landschaftsbaubetriebs', 0.8224, 1)
('agrarstruktur', 0.8134, 2)
('gartenpflege', 0.8126, 2)
('landwirtschaftszubehör', 0.8082, 1)
('agrarprodukten', 0.8081, 7)
('landschaftsbau', 0.8022, 45)
('landwirtschaftsflächen', 0.7985, 1)
('viehzucht', 0.7979, 2)
('straußenfarm', 0.7949, 1)
('landschaftsbaues', 0.792, 2)
('grünlandpflege', 0.7917, 1)
('gärtnerei', 0.791, 2)
('pflanzenzucht', 0.7906, 2)
('landwirtschaftlicheund', 0.7905, 1)
('kleingärtnerei', 0.7896, 2)
('zierpflanzenbau', 0.7876, 1)
('gärtnereibedarf', 0.7858, 1)
('landschaftsbauarbeiten', 0.7815, 4)
('landwirtschaftbetriebes', 0.7801, 1)
('landwirtschaftsgeräten', 0.7794, 1)
('gartenlandschaftsbau', 0.7753, 3)
('bewirtschaftungsleistungen', 0.7749, 1)
('nutzpflanzenrassen', 0.7749, 1)
('kultivierung', 0.7731, 2)
```

(a) Results using the *noun-only* approach

```
('landwirtschaften', 0.9415, 1)
('landwirtschaftsgestaltung', 0.9389, 1)
('landwirtschaftliche', 0.9309, 6)
('landwirtschaftlich', 0.9226, 3)
('landwirtschaftlichen', 0.9211, 55)
('landwirtschaftlicher', 0.9205, 19)
('landwirtschaft', 0.9143, 15)
('landwirtschaftlticher', 0.8969, 1)
('agrarbereich', 0.8918, 2)
('agrarflächen', 0.8911, 1)
('landwirtschftlicher', 0.8808, 1)
('agrar', 0.8792, 3)
('ackerbauund', 0.8765, 1)
('viehwirtschaft', 0.8737, 2)
('landwirtschaftlicheund', 0.865, 1)
('agrarprodukten', 0.8616, 7)
('landwirtschaftbetriebes', 0.8614, 1)
('agrarstruktur', 0.8596, 2)
('landwirtschaftszubehör', 0.8425, 1)
('agrarsektor', 0.8332, 1)
('landwirtschaftsflächen', 0.8274, 1)
('viehzucht', 0.8229, 2)
('bauernhofes', 0.8193, 1)
('ackerbau', 0.8177, 1)
('landwirtschaftsgeräten', 0.816, 1)
('agrarwirtschaftlichen', 0.8153, 1)
('agrarerzeugnissen', 0.8127, 7)
('forstwirtschaft', 0.8118, 2)
('erwerbsgartenbau', 0.7953, 1)
('landschaftsbaubetriebs', 0.7947, 1)
('farmerzeugnisse', 0.7914, 1)
('straußenfarm', 0.7903, 1)
('landschaftspflege', 0.775, 7)
('gartenbaubetrieb', 0.7747, 1)
('holzwirtschaft', 0.7729, 1)
```

(b) Results using the *unigram-only* approach

Figure 5.9.: Comparison of the top keywords extracted in class A (agriculture, forestry, and fishing) using both *noun-only* and *unigram-only* approach. All remaining parameters were identical: the Telekom model for extraction, 5 iterations with words scoring above the 99.5th percentile added to the seed keyword list. The scores were computed using the *mean seed* approach. The values in the brackets denote (keyword, score, occurrence count).

At this stage, it is preferable to adopt a more inclusive approach towards the acceptance of keywords. This implies that it is more beneficial to capture a larger set of keywords, even if it includes some irrelevant ones, rather than risk omitting potentially valuable keywords. This more liberal approach allows for a subsequent refinement process where class-unspecific keywords can be sifted out. Therefore, we opted to forgo this initial *preprocessed extraction* strategy, and this was a primary reason for excluding preprocessing in our workflow (see Subsection 5.1.5).

For the *assignment* strategy, which involved extracting nouns and noun phrases without any seed keywords or modifications to the original KeyBERT algorithm, we noticed minor variations among the models. For example, the default model `sentence-transformers/all-MiniLM-L6-v2` exhibited a stronger inclination towards extracting named entities compared to other models tested, including `T-Systems-onsite/cross-en-de-roberta-sentence-transformer` and the Telekom model. On the other hand, `symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli` showed a tendency to prioritize more generic keywords over class-specific ones in the same context. For instance, given the text "Betrieb eines Imbisses", where "Betrieb" is considered a generic word and "Imbiss" is specific to class I, the model assigned a higher score to the generic term. However, these differences between the models did not significantly influence the quality of their outputs.

For the task of assigning a unique extracted keyword to its most similar class, the naive approach involved comparing the keyword's embedding to the mean embedding of the class's seed keywords. While this approach is efficient, it has notable limitations. Simply averaging the seed embeddings can result in a centroid that does not accurately represent the broader range of concepts covered by the class, thereby affecting the quality of the assigned keyword set. If a word is similar to a particular seed, it is likely that both belong to the same class. In such cases, using the mean could be counterproductive, possibly lowering the similarity score. To address this issue, an enhanced method was employed in which each extracted keyword's embedding was compared to the embeddings of all seed keywords across all classes. The maximum similarity score is identified, and the extracted keyword is assigned to the class associated with this top-scoring seed keyword. This approach, although computationally more intensive, provided a more accurate keyword-to-class assignment.

Even with this refined approach, we identified some irregularities in the quality. Keywords already present in the seed set or those sharing the same morphemes with the seed keywords were generally correctly assigned, while the assignment of other keywords was not always ideal. For example, the word "yoga" was wrongly categorized under a class on mining. The previously created generic class also received keywords genuinely specific to an original class. For instance, the terms "salespromotionprojekten" (sales promotion projects) and "bereichen vermarktung" (marketing fileds) were allocated to the generic class, despite being more fitting for an original class that includes sales and marketing activities.

While some degree of misclassification is inevitable, the assignment strategy confines each keyword to a single class, essentially eliminating the possibility for other classes

to consider it as a potential keyword. Since we are comparing the extracted keywords with the seed keywords, improving the quality of the assignment results might entail using higher-quality seed keywords in larger quantities. This would, however, require significant manual effort, making it unsuitable for our pipeline that is intended to be predominantly automated. Moreover, since this thesis aims to obtain a comprehensive keyword set for each class without explicitly involving classification, incorporating an assignment problem at this early stage might be counter-productive, leading to our decision to abandon this assignment strategy.

The *guided KeyBERT* strategy (see Subsection 4.5.5) yielded a notably larger set of keywords per class. While many extracted keywords might not be class-specific or meaningful, and some have notably low similarity scores even below 0.2, our earlier discussion emphasized the importance of capturing a broader spectrum of keywords at this initial stage.

Although yielding significantly higher keyword counts, the *guided KeyBERT* strategy demands more computational power and time. Specifically, it requires running the extraction on the entire data subset for each of the 21 classes individually, with each session lasting around 15 minutes on a GPU. For comparison, the *preprocessed extraction* strategy averages 3.5 minutes per class on a GPU, with class variations due to the different numbers of class-specific entries after preprocessing. Meanwhile, the *assignment* strategy completes the entire extraction for all classes in a single round in approximately 15 minutes, and then takes roughly 10 seconds for assigning each extracted keyword to a class. This makes it the most time-efficient option, averaging less than a minute per class. This comparison underscores the trade-off between the quantity of extracted keywords and the computational resources required.

Upon qualitative analysis of the three strategies, we selected the *guided KeyBERT* approach. Despite its computational intensity and the inclusion of irrelevant keywords, this approach aligns best with our objective of developing a predominantly automated pipeline that minimizes manual intervention while maximizing the comprehensiveness and class-specificity of the keywords for each class. A comparison of the advantages, drawbacks, and other metrics of the three strategies is presented in Table 5.2.

5.2.4. Insights from Iterative Extraction

In our analysis of the iterative approach for keyword extraction, we observed a nuanced interplay between benefits and drawbacks. The most significant effect of such iteration is its ability to steer the extraction more closely towards the seed keywords and previously high-scoring keywords. This direction can be either beneficial or detrimental depending on the initial seed keywords. For example, consider the term "holzverarbeitung" (wood processing) which is specific to class C on the manufacturing industry. Without iteration, this term had a high score of 0.7409 in class A on agriculture and forestry, ranking 46th in the 8,664 extracted keywords of the class. Related terms like "holzbearbeitung" (woodworking) and "holzarbeiten" (woodwork) also had high scores in class A, which can be considered an inaccurate inclusion. With iteration, however, these terms were

	Preprocessed Extraction	*Assignment*	*Guided KeyBERT*
Advantages	• Filters out irrelevant entries for each class • Less time needed for extraction • Incorporates seed keywords	• Rapid processing time	• Significantly larger keyword lists • More accurate results due to inclusion of seed keywords
Drawbacks	• Class-specific entries could be incorrectly filtered out • Overlooks potential class-specific keywords • Fewer keywords	• Inaccuracies and misclassification • Assumes each keyword only belongs to one class • Fewer keywords	• The most time-consuming
Approximate Avg. Time (min)	3.5	0.72	15
Avg. Keyword Count	1,039	571	7,867

Table 5.2.: A comparison between the three extraction strategies.

no longer in the top-scoring range, highlighting the benefits of the iterative approach to reduce inaccuracies.

Class A also offers a counterexample. When employing the iterative approach, terms related to "gartenbau" (horticulture) consistently achieved higher scores and appeared frequently in the top-scoring keywords. Some were ranked among the top 20 extracted keywords, even though they were not truly associated with class A, but rather with class N. This implies that "gartenbau" or similar terms were added to the seed keywords during an iteration, demonstrating the influence of the iterative approach in steering the extraction direction. If this steering is misdirected, inaccuracies can propagate. As a result, the precision of the top-ranked extracted keyword in class A was significantly lower compared to classes where no erroneous keywords were added to the seed (Figure 5.11).

Despite the challenges associated with the iterative approach, we argue that its continued use is essential, provided that error mitigation mechanisms are integrated. Without iteration, the extraction remains confined to the initial seed keywords, implying that any subsequently added keywords are merely extensions of these seeds. This constraint

narrows the scope and diversity of the extraction, which can be particularly problematic if the initial seeds are incomplete or contain inaccuracies. The iterative approach is thus a necessary step towards capturing a broader spectrum of relevant keywords, as some level of diversification is more favorable than none at all. However, its effectiveness can be further enhanced with the integration of a robust error mitigation mechanism, which is a crucial next step discussed in more detail in Section 6.3.

5.2.5. Comparative Results from Keyword Scoring Approaches

After adopting the *guided KeyBERT* strategy for keyword extraction, the default *mean seed* scoring approach was initially applied. However, averaging the seed embeddings, instead of using each seed's embedding individually, could potentially dilute the class-specificity, inadvertently steering the extraction towards a more generic direction.

Conversely, the *max seed* approach presented its own set of shortcomings. All of the models we experimented with exhibited a tendency to favor candidates with morphological similarities to the seed keywords, potentially disregarding their semantic relevance. This issue was particularly pronounced in the `sentence-transformers/distiluse-base-multilingual-cased-v1` model compared to others (see Subsection 5.2.1). Similar observations were made for other models, including the high-performing Telekom model, but to a smaller extent.

Moreover, the *max seed* approach occasionally assigns excessively high scores to irrelevant terms, making it less reliable when used in isolation. Within the "Erziehung und Unterricht" (education and teaching) class (class P), for example, the extracted term "flugzeugen" (airplanes) is contextually related to the seed keyword "Flugschule" (flight school), but not directly relevant to the broader category of education and teaching. However, its similarity score reached 0.8309 with the *max seed* method, whereas it was not even extracted by the *mean seed* approach. This illustrates a pitfall of the *max seed* scoring method—potentially overrating terms with semantic similarity but no class relevance. In such scenarios, averaging scores from both methods can moderate the influence of irrelevant terms. The term "flugzeugen" received a final average of 0.5911 after factoring in its *mean seed* similarity score.

Consequently, we opted to average the scores from both the *mean seed* and *max seed* approaches for each candidate keyword as its final score, ensuring a more balanced scoring system that mitigates the individual limitations of each method. In this way, the *mean seed* score serves as a check against the possibility of overvaluing morphological similarities over genuine class relevance. Notably, the *average scoring* approach computes the score during each iteration, resulting in a final score that might differ from a direct average of the concluding *mean seed* and *max seed* scores.

To illustrate, within class P, "Sportunterricht" (physical education) acted as a seed keyword. The related term "sportkurse" (sports course) received the following scores across different scoring techniques after the iterations:

mean seed:	0.5806	ranked 301 out of 5,705
max seed:	0.9146	ranked 9 out of 6,151
average:	0.7446	ranked 157 out of 7,006

In this example, "sportkurse" is highly class-specific and should ideally receive a high score. However, the *mean seed* score ranked it fairly low, while the *max seed* score ranked it significantly higher. Taking their average resulted in a ranking of 157th out of 7,006, offering a more balanced and representative assessment of the word's relevance and mitigating the potential undervaluation by the *mean seed* approach.

5.2.6. Consolidated Extraction Outcomes of Finalized Parameters

We finalized the parameters for our keyword extraction as follows:

- Model: `deutsche-telekom/gbert-large-paraphrase-cosine`
- Keyword scope: unigrams only
- Extraction strategy: *guided KeyBERT*
- Iteration: five iterations of 2,000 entries each were processed for each class, with extracted keywords above the 99.5th percentile of the scores added to the seed keywords after each iteration
- Scoring approach: average of *mean seed* and *max seed* scoring

With these parameters, we extracted 165,206 keywords across the 21 classes, among which 18,828 were unique keywords. An average of 7,867 keywords were extracted for each class, ranging from a minimum of 6,581 to a maximum of 8,920 for individual classes. The detailed breakdown per class can be found in Figure 5.10.

Initially, we operated under the assumption that a class-specific keyword could only be unique to a single class. However, upon closer inspection, we observed that a word's class-specificity can vary depending on its surrounding context. For example, consider the term "Druckererzeugnisse" (printed materials). In the context of "Verlegen von Druckererzeugnissen" (publishing of printed materials), it falls under class J on information and communication. Yet, when associated with its production ("Herstellung von Druckererzeugnissen"), it becomes a keyphrase in class C that is related to the manufacturing industry. Similarly, when referring to its trade ("Handel von Druckerzeugnissen"), it belongs to class G on trade. Given that our approach exclusively extracted unigrams, the surrounding context window remains unknown at this stage. As a result, we opted not to implement an inter-class deduplication, thereby allowing a keyword to be associated with multiple classes. This decision was strategic to avoid inadvertently overlooking any potential keywords, thus preserving the possibility to assess their class-specificity in the next phase of context window identification in the CD4AI project pipeline.

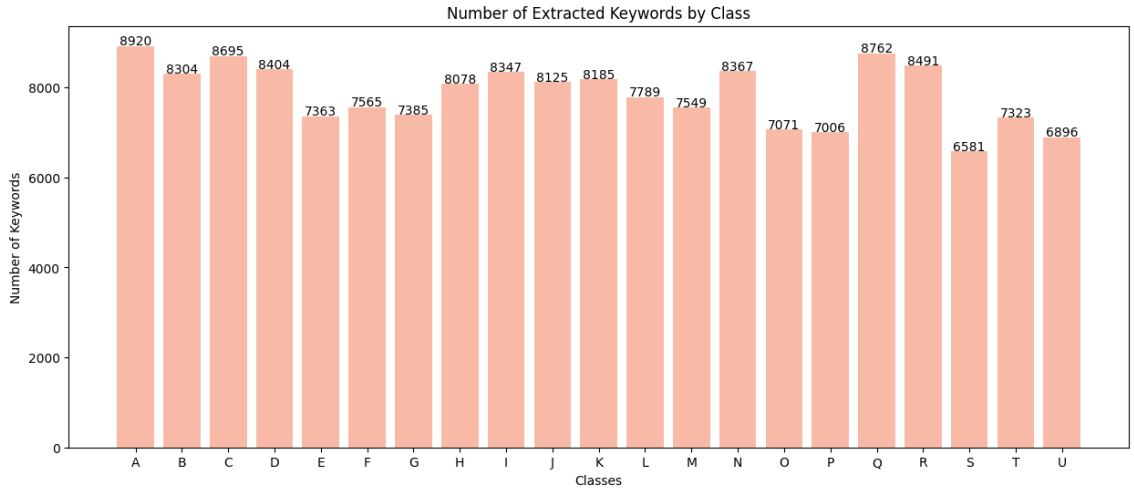


Figure 5.10.: Number of keywords extracted for each class using *guided KeyBERT* in five iterations. Scores in each iteration are computed using the average between *mean seed* and *max seed* approach.

Qualitative assessment revealed that the topmost 90–100 extracted keywords in each class were predominantly class-specific with minimal irrelevant terms. This led us to employ Equation 5.1 that would yield between 90 and 100 extracted keywords per class:

$$x = 5 \cdot (\ln(n) - \ln(0.001)) \tag{5.1}$$

We conducted a qualitative assessment of the precision of the filtered extracted keywords in some sampled classes. It is important to note that this assessment is inherently subjective and should be used as a reference rather than an absolute measure of precision. The computed precision values, illustrated in Figure 5.11, reflect the extent to which the filtered keywords were class-specific. In this assessment, typographical errors were overlooked if the word, when correctly spelled, was class-specific. Some classes exhibited lower precision, which could be attributed to two main factors. Firstly, the iterative process of keyword extraction might have led to the inclusion of incorrect keywords in the seeds, subsequently affecting the precision. An illustrative case is class A due to the erroneous inclusion of terms related to "gartenbau", as detailed in Subsection 5.2.4. Secondly, the inclusion of overly generic keywords, which do not pertain to any specific class, could have contributed to the reduced precision. For instance, the highest-ranked keywords in class N included several keywords associated with "dienstleistung" (service), a term too broad that should not be classified as specific to any particular class.

5.3. Generation Results

In this section, we present the results from our three-stage keyword generation: lexical substitution, synonym generation, and word form generation, as introduced in Sec-

tion 4.6.

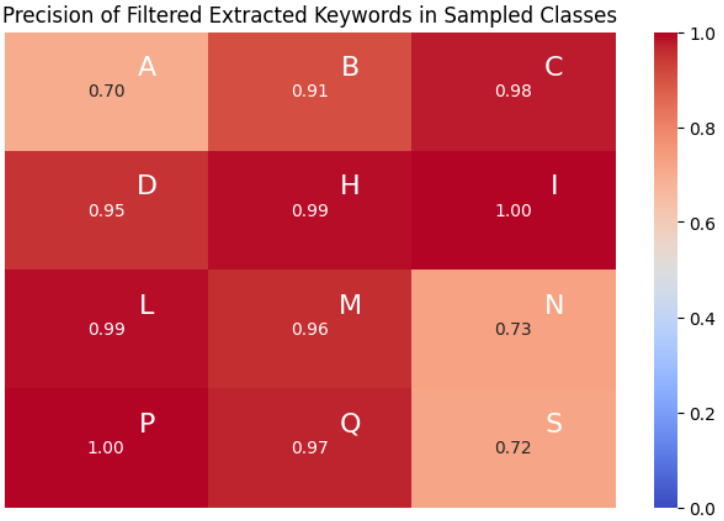


Figure 5.11.: Qualitatively computed precision scores for extracted keywords in sampled classes after filtering

5.3.1. Lexical Substitution Results

Among the models we experimented with (including `xlm-roberta-base`, `bert-base-german-cased`, `deepset/gbert-base`, and others, as documented in Subsection 4.6.1), we first discarded `xlm-roberta-base` due to its evident lack of consideration for the context-specific meaning of the targeted word. For example, the model was tasked to substitute the word "Landwirtschaft" from class A (agriculture, forestry, and fishing) given the following class description as context:

> Abschnitt A beinhaltet Landwirtschaft, Forstwirtschaft und Fischerei. Dieser Abschnitt umfasst die Nutzung der pflanzlichen und tierischen natürlichen Ressourcen. Dazu zählen Tätigkeiten wie Pflanzenbau, Tierzucht und Tierhaltung, Holzgewinnung und die Gewinnung anderer pflanzlicher und tierischer Erzeugnisse in land- oder forstwirtschaftlichen Betrieben oder in freier Natur.

For this input, the model output the following options:

```
Land, Wirtschaft, Handel, Landschaft, Produktion,
Lebensmittel, Ernährung, Handwerk, Natur.
```

From the results, it is evident that the model occasionally simply breaks down a word into its constituent morphemes, which are often neither semantically nor contextually appropriate. Moreover, terms like "Handel", "Produktion", "Lebensmittel", and "Handwerk" are not specific to class A, but rather fit as keywords in other classes.

As a comparison, the `bert-base-german-cased` model yielded the following:

`Fischerei, Forstwirtschaft, landwirtschaft, Naturschutz Jagd, Handel, Landwirte, Tier.`

While they are not perfect, these substitutes are more suitable and specific to our class on agriculture and forestry.

We conducted a qualitative assessment of the performance of the remaining models, mainly evaluating the precision of their outputs. A model is deemed performant if its results primarily consist of class-specific keywords. For example, the outputs from the `bert-base-german-cased` model, as shown above, display a higher precision than those of the `xlm-roberta-base` model, as only the terms "Handel" and "Naturschutz" are specific to another class. Based on this assessment, none of the remaining models significantly outperforms or underperforms the other ones across all classes and contexts (refer to Subsection 4.6.1 for a comprehensive context definition). Therefore, while setting the default to the `deepset/gbert-base` model, considering its recent release and lightweight nature, we also allow users to specify their preferred model.

Upon evaluating results from the four contexts defined in Subsection 4.6.1, we concluded that merely using the class name and target word as a context produced the poorest results. For example, when substituting the word "Landwirtschaft" for class A, which has the class name "Land- und Forstwirtschaft, Fischerei" and the class description as defined above, we obtained the following outputs using the `dbmdz/bert-base-german-uncased` model:

class name + no seed:	`landwirtschaft, verkehr, gesundheit, wirtschaft, handel`

Whereas the other contexts delivered much better results:

description + seed:	`landwirtschaft, forstwirtschaft, land, landwirtschafts, wirtschaft`
class name + seed:	`landwirtschaft, land, forstwirtschaft, bergbau, landwirtschafts`
description + no seed:	`landwirtschaft, land, forstwirtschaft, bergbau, jagd`

Similar observations were made for the other lexical substitution models. To avoid redundancy, we refrain from enumerating the findings for each model. Consequently, we discarded the last context, which consisted solely of the class name and the target word. However, the remaining three contexts displayed similar efficacy, with none significantly outperforming the others. Despite overlaps in their results, with an average of 43.76% duplicated substitute words using the `deepset/gbert_base` model, each context's substitute word set is distinct. None is a true subset of another, indicating that each context introduced unique words to the set. Therefore, we decided to incorporate all three contexts in our approach, prioritizing a broader word generation at the costs of higher computational intensity, rather than risking overlooking potential class-specific substitutes.

Following the deduplication, an average of 19.88% of the unique substitutes among the classes were removed since they already existed in the class's keyword set, either as a seed keyword or as an extracted keyword. Figure 5.12 shows a comparison of the results across three different stages for each class:

1. The initial set of substitutes generated.
2. After deduplication of the initial set.
3. After exclusion of existing keywords from the deduplicated set.

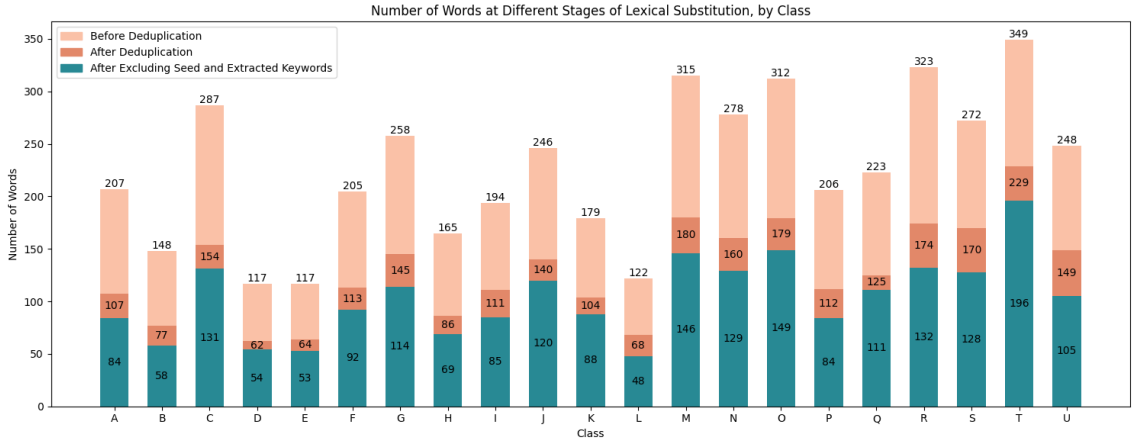


Figure 5.12.: Number of substitute words at different stages of lexical substitution using the `deepset/gbert_base` model and combining the three contexts

For the remaining lexical substitutes in each class after deduplication and exclusion, we computed their similarity to the class's seed keywords by averaging the scores of the *mean seed* and *max seed* scoring techniques. This was done to assess their class-specificity. An excerpt of the resulting scores from five classes is shown in Figure 5.13. As these lexical substitution results will be used for subsequent generation of synonyms and word forms, it is important to maintain a high precision, ensuring that only truly class-specific keywords are added to the class's keyword set. In order to minimize the propagation of irrelevant keywords in the subsequent generation stages, we set a relatively stringent threshold—the 75th percentile of similarity scores to the seeds—and filtered out words below this threshold. Illustrated by Figure 5.13, the choice of the 75th percentile becomes evident. While some classes might include relevant keywords between the 50th and 75th percentiles, a substantial number of these keywords lacked class-specificity. This lack is even more noticeable for terms ranking below the 50th percentile.

Utilizing the `deepset/gbert-base` model on a GPU, the entire lexical substitution took approximately 40 minutes, yielding a total of 4,771 scored lexical substitutes from an initial pool of 1,980 existing keywords. This translates to around 1.25 seconds to perform lexical substitution on one existing keyword using all three contexts. After

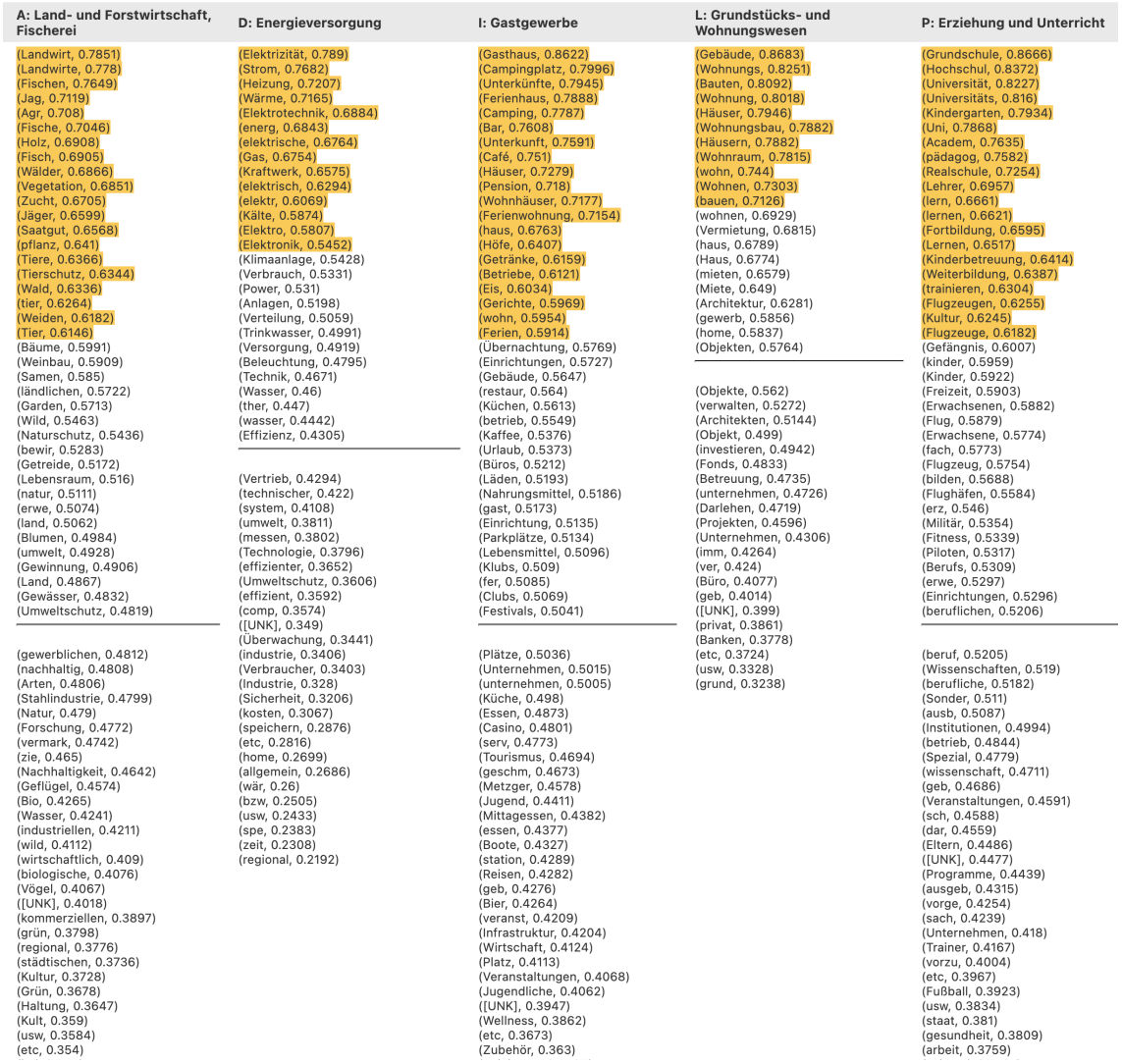


Figure 5.13.: Lexical substitutes on some sampled classes after computing their similarity to the seed, ranked in descending order of the similarity score. The horizontal line denotes the 50th percentile, and substitutes scoring above the 75th percentile are highlighted. The lowest ranked results of class A, I and P are not fully shown.

excluding existing keywords, eliminating duplicates, and filtering out the bottom 75 percent of these substitutes, 550 new keywords were added across classes, averaging to around 26 per class. The exact number of keywords added for each class is available in Figure 5.14.

5.3.2. Synonym Generation Results

While the majority of the existing keywords in the class keyword sets have unambiguous meanings, the filtering of the different senses and their corresponding synsets was still necessary. This is due to the lack of context in the synsets that may cause them to contain highly irrelevant terms. For instance, the Oxford German Dictionary [54] records the following senses for the word "Abfall":

1. rubbish, garbage, waste;
2. drop;
3. apostasy.

In the specific class context of "Abfallentsorgung", this is clearly referring to the first sense. OdeNet provides the following synsets:

1. {`Senkung, Gefälle, Hang, Abhang`}
2. {`Untergang, Niedergang, Sinken, Fall, Sturz, Fallen`}
3. {`Einbue, Verringerung, Nachlassen, Schwund, Rückgang, Regression, Degression, Dekreszenz, Abnahme, Dämpfung`}
4. {`Schrott, Spreu, Unrat, Kehricht, Ausschuss, Hausabfall, Müll, Hausmüll, Schmutz, Dreck, Siff`}.

The last set, with a weighted average score of 0.5406 (Equation 4.3), was evidently the most relevant.

Yet, sometimes, multiple synsets from OdeNet might be suitable to provide more class-specific keywords. For example, in the context of class I on gastronomy, one extracted keyword was "Café". When using OdeNet, two synsets were given:

1. {`Kaffeehaus`}, score = 0.6541;
2. {`Bistro, (kleines) Lokal`}, score = 0.5346.

While our algorithm only selected the first synset and added "Kaffeehaus" to the class's keyword set, both terms in the second synset were also relevant for the class. This highlights the need for further refinement to incorporate a smarter filtering mechanism that includes all relevant synsets.

On the other hand, elements within a selected synset might not always be relevant or specific to a given class. Even when a word is only associated with one synset, it is not always accurate to assume that the meanings within the synset match the intended meaning of the target word. For example, consider the word "Mineral" in a mining context. The only synset in OdeNet for this word corresponds to the meaning of "mineral water" rather than "mineral" with a geological interpretation, leading to the following associated synonyms:

```
{Soda, Mineralwasser, stilles Wasser, nervöses Wasser, Selters,
Arbeitersekt, Tafelwasser, Sprudel, Selterswasser, Wasser,
Sodawasser, Eskimo-Flip, Kribbelwasser, Wasser mit Zisch,
Sprudelwasser, saurer Sprudel}.
```

Although such cases are rare, they introduce unnecessary noise that causes the class keyword set to be less coherent. This underscores the necessity for a better filtering mechanism to ensure class-specificity of the generated synonyms.

In the end, a total number of 2,246 synonyms were generated, among which 2,044 were not already in their corresponding class's keyword set and were thus added. This resulted in an average of around 98 new keywords added per class. The process on all classes took 75 minutes to run on a GPU, averaging to roughly 2 seconds per generated synonym. The number of unique synonyms added to each class can be found in Figure 5.14.

5.3.3. Word Form Generation Results

Before utilizing the POS taggers to identify all adjectives in our keyword set, we first removed all the nouns and their word forms identified by `german-nouns`. This step was necessary as the POS taggers have a notable tendency to mislabel non-adjectives as adjectives. The combination of both HanTa and SpaCy in the POS tagging process was essential since neither showed optimal performance alone.

HanTa's case sensitivity is such that it often tags an unknown word as a noun if its initial letter is capitalized. For our purposes, we deemed a word "unknown" to HanTa if it could not decompose the word into more than two morphemes. For instance, the word "energiebozogener" is correctly labeled as an adjective by HanTa with its lemma recognized as "energiebezogen". Moreover, it deconstructs the word into its core morphemes:

```
('energiebezogen',
[('energie', 'NN'), ('bezog', 'VVnp_VAR_PP'), ('en', 'SUF_PP'),
    ('er', 'SUF_ADJ')],
'ADJ(A)')
```

However, upon capitalizing the first letter to "Energiebezogener", HanTa misconstrues it as a noun due to the capitalization:

```
('energiebezogener', [('energiebezogener', 'NN')], 'NN')
```

Conversely, a lowercase initial letter can lead HanTa to misidentify a noun as an adjective, as demonstrated by the following example:

```
natursteinmauern:  ('natursteinmauern',
                    [('natursteinmauern', 'ADJ')],
                    'ADJ(D)')
```

Whereas when capitalized, the POS is correctly identified:

```
Natursteinmauern:   ('natursteinmauer',
                    [('natur', 'NN'), ('stein', 'NN'),
                        ('mauer', 'NN'), ('n', 'SUF_NN')],
                    'NN')
```

Due to this difference resulting from capitalization, we let Hanta analyze both the uppercased and lowercased version of every word. Our previous examples revealed that when HanTa segments a word into more than two morphemes, the labeled POS is typically accurate. Therefore, we determined the word's POS based on the casing that results in a multi-morpheme breakdown, irrespective of the `capital` flag. However, if neither casing delivered a segmented morpheme analysis, we would default to the POS and lemma specified by `capital`. The integration of SpaCy's tagger would then occur only if the POS is an adjective.

A qualitative assessment showed that SpaCy's tagger from the pipeline `de_core_news_sm` was less accurate than HanTa concerning our extracted keywords. The larger model `de_core_news_lg` did not exhibit substantial improvements either. Hence, we prioritized the results from HanTa, resorting to SpaCy only if HanTa failed to break down a word yet still tagged it as an adjective. Since SpaCy performs more accurately when given the lemma rather than inflected form, we directly provided it with the output lemma from HanTa. For our tagging task with SpaCy, we only accepted words as adjectives or adverbs if SpaCy tagged them as `ADJ` or `ADV` and if the generated lemma did not start with a capital letter. This decision stemmed from our observation that SpaCy, although sometimes misclassifying a noun's POS, could still retain the noun's initial capitalization in its lemma. For example, it assigns the `VERB` tag to the extracted keyword "gärtnereibedarf", but recognizes its lemma as "Gärtnereibedarf" with the correct capitalization for nouns.

Our default value for the argument `capital` is `True`, favoring tagging unknown words as nouns over adjectives. This reduces the risk of mislabeling non-adjectives as adjectives and prevents the propagation of errors, particularly given the presence of invalid extracted keywords resulting from typographical errors in the data set. For instance, the string "steuerund" is most likely an error resulting from the two words "steuer" (tax) and "und" (and). HanTa tags the uncapitalized version as an adjective and the capitalized version as an named entity:

```
steuerund: ('steuerund', [('steuerund', 'ADJ')], 'ADJ(D)')
Steuerund: ('steuerund', [('steuerund', 'NE')], 'NE')
```

When using Spacy, we get the following POS tags:

```
steuerund: ADV
Steuerund: NOUN
```

Defaulting `capital` to `True` might lead to the oversight of some rare adjectives. However, any potential omission at this stage is no longer detrimental, since we have at least one form of the overlooked adjective in our keyword set. On the other hand, a greater risk is associated with misclassifying non-adjectives as adjectives, especially given the subsequent step of appending endings to the adjective lemmas, which could introduce unnecessary noise and invalid words into the adjective list.

The derivation of word forms proved efficient, with a full cycle across all classes completed within 21 seconds on a GPU, averaging to one second per class. The process generated 8,228 word forms in total, of which 4,166 were unique additions not already present in the respective class's keyword set, resulting in an average addition of approximately 199 new keywords per class (see Figure 5.14).

5.4. A Comprehensive Statistical Overview of the Pipeline

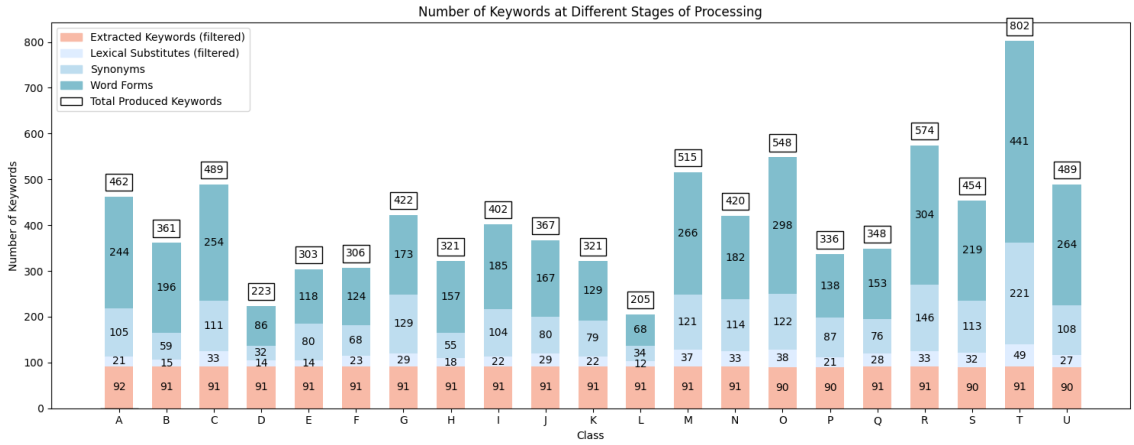


Figure 5.14.: Number of new and unique keywords in each pipeline stage by class

Figure 5.14 offers an overview of the distribution and accumulation of keywords across different stages for each class. On average, the final keyword sets across all classes contained 8,668 keywords, averaging to 413 keywords per class. These retained keywords were curated from substantially larger initial keyword sets to guarantee relevance to the content of their respective classes. Notably, for most classes, the word form generation stage yielded the highest number of keywords, a count that reflects the number of existing keywords present within that class.

Figure 5.15 illustrates the accumulation of keywords and the time taken for each stage in the pipeline, averaged for each class. The average duration for processing a single class through the entire pipeline is approximately 20.5 minutes. The extraction stage was the most time-consuming, mainly due to the larger size of the data set (10,000 entries) and the slower processing speed of the Telekom model (Subsection 5.2.1). The figure also provides insights into the filtering process at each stage. The dots marked on the green line represent the cumulative count of unique keywords per class retained

after each stage. The adjacent annotations specify the average number of keywords filtered and retained out of the total keywords produced for that stage. It is evident that more stringent filtering mechanisms were applied during the first two stages—extraction and lexical substitution—with only about 1.16% and 11.84% of keywords retained, respectively. In contrast, the latter stages retained a much higher proportion of keywords: 91.59% from synonym generation and 51.02% from word form generation. The stricter filtering in the early stages was essential to ensure class-specificity, given their higher propensity to produce keywords irrelevant to the specific class context.

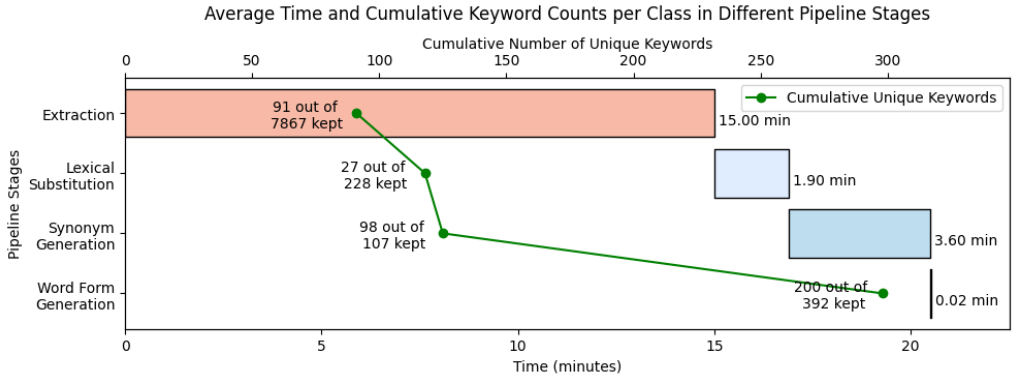


Figure 5.15.: Visualization of the average time duration and cumulative keyword counts for each pipeline stage, averaged per class. Horizontal bars represent stage durations (in minutes), while the green line tracks the cumulative keyword counts retained after each stage. Annotations (".. out of ..") specify the number of keywords filtered and retained *out of* the total extracted or generated in each respective stage.

5.5. Evaluation Results

In the following subsections, insights gained from domain experts, general participants, and computational evaluations are presented to provide a critical analysis of the efficacy of our proposed approach and potential areas for further refinement.

5.5.1. Domain Expert Evaluation Results

During the course of this thesis, four domain experts have participated in the evaluation process. On average, the experts identified 80.53% of the presented keywords as accurately representing their respective classes.[7] While this percentage indicates a decent degree of relevance in the extracted keywords, it also suggests areas of improvement in

[7]Data updated as of September 14, 2023, 13:30. This indicates the last update before the thesis submission, acknowledging that further evaluations might have been conducted after this point.

Class	% of Keywords Validated
A	92.11%
B	76.32%
M	59.21%
P	90.79%
S	84.21%
Overall Average	80.53%

Table 5.3.: Percentages of extracted keywords validated by domain experts as class-specific in the sampled classes (as of September 14, 2023)

refining our extraction methodology to enhance accuracy and class-specificity. Table 5.3 provides a detailed breakdown of the expert validation results for each class.

As observed in Table 5.3, Class M recorded the lowest validation rate among the evaluated classes. Beyond typographical errors such as truncated ("unternehmenver") or concatenated ("managementund") keywords, experts also rejected generic terms like "firmen" (firms) and "betriebliches" (operational). Such generic terms likely originated from seed keywords with morphemes related to the notion of a company, such as "unternehmen" (company) or "firmen", leading to elevated similarity scores of these general keywords. This outcome indicates the need for an enhanced filtering mechanism to eliminate generic terms from the extracted keyword set.

While some domain experts highlighted the presence of typographical errors and English words in their feedback, it is important to note that these issues are inherent to the data set itself and are not direct consequences of our extraction methodology. Additionally, some experts found it challenging to differentiate between keywords due to their pronounced similarity. On the one hand, this underscores the coherence and class-specificity of the extracted keywords, aligning well with our primary objective. On the other hand, it suggests that a broader keyword spectrum could provide a more comprehensive class representation. This notion aligned with our motivation for developing the keyword generation approach. Although the current generation results (Section 5.3) present their distinct challenges and have not been validated by domain experts, there is significant potential for refining this approach to introduce more diversity and enhance its efficacy.

5.5.2. Intruder Detection Survey Results

The intruder detection survey, designed to assess the coherence and potentially the class-specificity of the keyword sets, was completed by a total of 18 participants, all of whom possessed substantial knowledge in German. The results are illustrated below.

For the *extracted-only* set, participants were able to correctly identify the intruder keyword with a high accuracy of 90.89%, indicating high coherence in the extracted keywords for each class. In comparison, when considering the *extracted and generated* set,

% Correctly Identified / Class	*Extracted-Only*	*Extracted + Generated*
A	91.11%	75.29%
B	96.67%	85.88%
M	85.56%	50.59%
P	93.33%	75.29%
S	87.78%	48.24%
Overall Average	90.89%	67.06%

Table 5.4.: Percentages of correct intruder identification in the *extracted-only* and *extracted and generated* keyword sets for the intruder detection evaluation

the accuracy dropped by an absolute value of 23.83 percentage points, resulting in an overall success rate of 67.06% for intruder identification. This suggests that the inclusion of generated keywords introduced elements that compromised the clarity and coherence of the keyword sets, leading to potential ambiguities for the participants. Specifically, the most significant reductions were observed for classes M and S, with drops of 34.97 and 39.54 percentage points, respectively. A detailed breakdown of the results across all classes is available in Table 5.4.

It is essential to note that the participants, while having substantial knowledge in German, lacked specific domain expertise. This could have influenced their decision to select the intruder keyword, as they might have different categorizations when interpreting the provided options without knowing the specific categories of WZ2008. For instance, given a question for class A with the options "viehhandlung" (livestock dealing), "landwirtschaften" (farming), "grundstoffen" (raw materials), "fisch" (fish), and "bauer" (farmer), with "grundstoffen" being the intruder from class B, participants might diverge in their interpretations. Some might categorize based on a "biological versus non-biological resources" perspective, while others could focus on a "land-based versus aquatic" theme, leading them to select "fisch" as the intruder. Such variations in interpretation underscore the limitations of this evaluation approach and highlight the value of validations based on expert domain knowledge.

5.5.3. Automatic Evaluation Results

Figure 5.16 presents the overall similarity between the keywords sets and the seed keywords across different stages. For clarity, the individual keyword sets corresponding to each stage are the same as those depicted in Figure 5.14.

The extracted keywords yielded the highest similarity score of 0.7476. This outcome aligns with expectations, since the extraction process was directly guided by the seed keywords to extract keywords that were most similar to the seeds.

Lexical substitution yielded an average score of 0.6794 across all classes, showing a

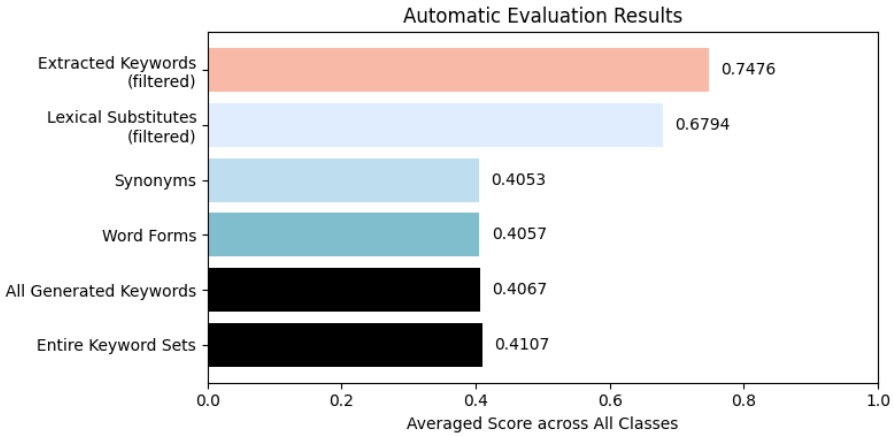


Figure 5.16.: Average similarity scores of keyword sets across different stages in relation to the original seed keywords

decent level of relevance to the seeds. This result is consistent with our methodology to leverage the seed keywords as a contextual guide as well as the high filtering threshold for retaining candidate substitutes.

The synonym generation process, while anticipated to produce a reduced score, surprisingly dropped to 0.4053. This sharp decline does not necessarily denote class irrelevance of the synonyms but rather their diminished similarity to the seeds, which could be attributed to the absence of seed-based filtering other than the simple word sense disambiguation. This outcome, along with our prior qualitative assessment in Subsection 5.3.2, emphasizes the need for enhanced filtering mechanisms, as will be elaborated in Subsection 6.3.1.

The word form generation, which does not introduce lemmatically new keywords, yielded a score marginally higher at 0.4057, suggesting that the newly introduced word forms were predominantly influenced by the synonym generation. This is consistent with the observation that the keyword sets after extraction and lexical substitution might already encompass several word forms for a given base word, and the word form generation stage only added forms not already in the keyword sets.

Considering the smaller counts of lexical substitutes (see Figure 5.14), the combined score for all generative processes (0.4067) was predominantly influenced by synonym and word form generations. Furthermore, the overall score for the entire keyword set (0.4107) reflects the dominance of the generative results (0.4067) over the extraction results (0.7476), primarily due to the larger volume of generated keywords as compared to the extracted ones (Figure 5.14).

While the automatic evaluation offers an objective means of assessing keyword relevance with minimal manual intervention, it is heavily dependent on the initial seed keywords, which might not always encapsulate the broader context of a class. Therefore, it is essential to complement these results with qualitative assessments for a more comprehensive evaluation.

6. Discussion

This chapter highlights the key contributions of our proposed approach, discusses the challenges and limitations encountered, and outlines potential avenues for future research.

6.1. Contributions

Throughout the course of this thesis, we have developed and refined methods that offer a pragmatic approach to real-world applications for incorporating domain knowledge in keyword extraction and generation. The main contributions can be summarized as follows:

- **Systematic Keyword Pipeline**: We have introduced a systematic pipeline tailored for domain-specific keyword extraction and generation. This approach integrates domain knowledge using seed keywords to guide the extraction process. Paired with a three-step generation strategy, our methodology is designed to produce keyword sets that are both comprehensive and class-specific. To the best of our knowledge, this is the first methodical approach that integrates domain knowledge in both keyword extraction and generation in this manner.
- **Efficiency and Efficacy**: Our approach significantly accelerates the keyword extraction process compared to manual extraction. Upon application to our data subset of $10,000$ entries with a total of $290,266$ words, the algorithm completes extraction for each class in approximately 15 minutes on a GPU, translating to an average processing rate of 323 words per second. For comparison, the average adult reading speed is roughly 4 words per second for English non-fiction [11]. When factoring in the actual extraction process, the time taken by a human would be considerably longer. This means our algorithm processes text at a rate approximately 80 times faster than just manual reading, and the disparity grows even larger when considering the entire manual extraction process. Furthermore, the incorporation of seed keywords enhances class-specificity, providing a notable advantage over conventional extraction algorithms, such as the standard KeyBERT model without using seed keywords.
- **Generalizability**: While our primary experiments were conducted with data from a particular domain, the fundamental principles and elements of our pipeline are highly adaptable to a wide range of domains, offering a high degree of flexibility.

- **Configurability**: Acknowledging the diverse needs of different projects and domains, our model has been designed to be highly configurable. It allows for adjustments in percentiles, model choices, and other parameters to fit specific requirements.
- **Intentional Design**: Every phase of our research was marked by deliberate choices and thoughtful decision-making. From model selection to score computation and threshold determination, each step was backed by valid reasoning and supplemented with experiments when necessary. This thorough process ensured a final pipeline with increased reliability and robustness.

6.2. Challenges and Limitations

In the following subsections, we detail the challenges encountered during the development of our approach and highlight areas that require further improvement.

Data Reliance and Subjectivity

Our approach is heavily dependent on the quality of the initial domain knowledge in the forms of class descriptions and seed keywords. Ideally, domain experts should be involved in the creation of this initial class-specific knowledge. However, it is important to acknowledge that even expert contributions cannot entirely eliminate subjectivity and bias, influenced by the experts' own preferences and past experiences.

Furthermore, in the absence of an objective ground truth, all aspects of our research required qualitative analysis. Consequently, decisions regarding optimal percentiles for seed additions, thresholds for keyword extraction and lexical substitution, among others, were determined experimentally. This inherently introduced degree of subjectivity into our approach.

Lack of Contextualization

The seed keywords were embedded as a list of words without being given any specific context. This less contextualized word representation can affect the accuracy of the candidate keywords' scores during the extraction process.

Similarly, the process of keyword generation, which encompasses both lexical substitution and synonym generation, faced a related challenge. When selecting lexical substitutes that scored above the 75th percentile of similarity to the seed keywords, the calculations were made without considering the context in which the lexical substitutes or the seeds could appear. Likewise, during synonym generation, determining the most appropriate synset for a polysemous word potentially became less accurate as neither the different synsets nor the seeds were incorporated into a specific context.

The absence of context in the embedding process presents a clear avenue for future exploration, which is discussed in Subsection 6.3.2.

Error Propagation

Due to the sequential nature of our pipeline, one notable limitation is the propagation of errors throughout the process. Mistakes made in the early stages can lead to subsequent errors in the later stages. Two primary sources of these early errors are the iterative extraction process and the filtering thresholds or their absence in various stages.

In the extraction process, the goal of iteration is to diversify the extracted set by adding top-scoring extracted keywords to the seed list. However, this can introduce a risk of adding irrelevant keywords. As evidenced in Subsection 5.2.4, an iteration that introduces an off-topic keyword into the seed keyword set, such as "gartenbau" in class A, can skew subsequent rounds towards similar irrelevant keywords. Such inclusions not only elevate these terms in the extracted set but can also influence downstream processes like lexical substitution and synonym generation, reducing the overall precision of the keyword sets. This is exemplified in Figure 5.11, where class A's precision declined significantly compared to classes B or I, which maintained keyword relevance.

The next area of potential error lies in the filtering thresholds applied during extraction and generation. Although we have established a seemingly high threshold for keyword extraction, it became evident that the optimal threshold varies depending on the specific class. As such, the inadvertent inclusion of unrelated keywords remains a risk.

When such irrelevant keywords make their way through the initial filtering, they can introduce errors in the succeeding stages. For example, lexical substitution could yield erroneous replacements if it operates on an incorrect keyword. Moreover, if the filtering threshold is not stringent enough in this phase, the problem will be further amplified. Subsequently, synonym generation can significantly expand the keyword set, particularly as there is no intra-synset filtering mechanism for the synonyms. During word form generation, all recognized nouns and adjectives undergo inflection, potentially further expanding the error scope. This can culminate in generating multiple additional forms for any misclassified noun or adjective.

The aforementioned limitations highlight the need for a more sophisticated error-protection mechanism that would reduce the propagation of errors throughout the different stages of the pipeline. Possibilities for the design of such mechanisms are discussed in more detail in Subsection 6.3.1.

Trade-Off between Precision and Exhaustiveness

We set a stringent filtering threshold—roughly the 99th percentile for extraction and over the 75th percentile for lexical substitution. This decision was driven by the intention to maintain the precision of keyword sets in terms of class-specificity and coherence.

On the negative side, this approach also means sidelining a substantial amount of potential keywords, especially in the extraction phase. While we limited our selection to the top 90 keywords for each class, we observed that a considerable number of valid class-specific keywords were found among the top 400 in some classes. Including all

400, however, would compromise precision by admitting more irrelevant terms to the keyword set, amplifying the risk of error propagation.

Recognizing this trade-off between precision and exhaustiveness of the keyword sets, it is essential to devise a mechanism that can effectively balance these two crucial aspects. Ideally, this would mean optimizing the keyword sets to contain as many class-specific keywords as possible while minimizing the inclusion of irrelevant ones. Solutions might involve developing more sophisticated filtering mechanisms or incorporating advanced NLP techniques to better differentiate between class-specific and unspecific keywords. A more detailed discussion on potential strategies to address this challenge can be found in Subsection 6.3.1.

Challenges in Keyword Generation Efficacy

The outcomes from both the intruder detection and automatic evaluation tasks indicate that the generation process often introduces potentially disruptive keywords rather than enhancing the overall quality (Section 5.5). This results in keyword sets that are less coherent and diverge from the seed keywords, implying a reduced class-specificity. Notably, a substantial part of these disruptive keywords is attributed to the synonym generation stage.

Given these findings, the inclusion of the generation process in its present form might not be worth the potential complications it introduces. More specifically, one might consider the feasibility of omitting the synonym generation step, especially since it appears to have a more detrimental effect on the keyword quality compared to the other generation steps—lexical substitution and word form generation. By excluding the synonym generation, we might be able to strike a balance between generating additional keywords and maintaining the coherence and class-specificity of the keyword sets.

However, entirely omitting the generation phase would limit our ability to derive class-specific keywords outside the extraction data set, thereby restricting our potential to create a comprehensive keyword set for each class. Consequently, while the current generation process is not optimal, we believe that it can be enhanced with targeted adjustments and the integration of more stringent filtering mechanisms, outlining a potential direction for future research.

Linguistic Challenges

The BR data set inherently contains spelling mistakes, presenting a linguistic challenge in our extraction process. The absence of a robust spell check algorithm for the German language hindered our ability to effectively identify and rectify typos and other linguistic anomalies. Although these inaccuracies are not a consequence of our extraction methodology, they did influence the quality and accuracy of the extracted keyword sets, as shown in Subsection 5.5.1. As a result, we tuned the word form generator to favor tagging unknown words as nouns rather than adjectives, in an effort to prevent incorrectly adding endings to these unknown words and possibly creating invalid words.

Nonetheless, this issue underscores the need for further investigation and development of more advanced linguistic processing tools for the German language.

The results of POS tagging and lemmatization pose another challenge. Despite employing proficient POS taggers and lemmatizers for the German language, their accuracy and robustness can be further improved. It is essential to recognize that this is not a limitation of our approach per se, as it is contingent on the availability and performance of external linguistic tools. However, it does highlight the need for continued research and development in the area of POS tagging and lemmatization for the German language.

Additionally, our approach to word form generation by leveraging POS tagging and lemmatization tools was a workaround necessitated by the absence of a dedicated word form generation tool for the German language. Therefore, a German equivalent to the `word-forms` package, capable of generating all word forms and establishing connections between parts of speech, would be incredibly beneficial. While these needs are not directly associated with our methodology, addressing them would undoubtedly enhance the accuracy and comprehensiveness of the extraction and generation processes. However, it is also worth noting that these linguistic challenges are not unique to our study but are prevalent in the field of NLP, especially when working with non-English languages.

Limitations of Qualitative Evaluation Approach

Our qualitative evaluation strategies, though carefully devised, faced inherent challenges due to the intricacies of comprehensively evaluating our process. These challenges are detailed below:

- **Coverage and Comprehensiveness**: Due to the lack of a definitive "ground truth" for our keyword sets, our primary focus was on precision, accuracy, and relevance. This constraint meant we could not effectively measure the "recall"—an indicator of the coverage or comprehensiveness of our keyword sets for a given class or context.
- **Expert Availability**: Engaging domain experts apt for the evaluation proved to be a challenge due to the specialized nature of the data. This scarcity resulted in limited feedback, restricting the scope of insights that could have been derived from expert validation.
- **Partial Presentation**: To maintain practicality in the evaluation process, we only presented a subset of our results to domain experts. This means that certain aspects or characteristics of the resulting keyword sets could have been overlooked.
- **Subjectivity in Intruder Detection Task**: The intruder detection evaluation is not immune to personal biases. Given that the participants were not necessarily domain experts, their interpretations could vary widely, as shown in an example in Subsection 5.5.2. Depending on the presented keywords, there is a tangible risk that class-specific keywords might be inaccurately identified as intruders.

6.3. Future Work

The limitations identified in this study suggest several directions for future work, each aimed at addressing a specific challenge encountered during our research.

6.3.1. Better Filtering Mechanism

To address the limitation of error propagation due to the inclusion of irrelevant keywords, potential avenues for future research could involve the following improvements:

Exclude Keyword List

Since we concluded that the classes' keyword sets are not mutually exclusive, having a *exclude keyword* list—a list of words explicitly marked as "not in this class"—could be beneficial in filtering out irrelevant keywords and optimizing the keyword sets for class specificity and coherence. For example, if 'gartenbau' is in the seed keywords for class N, it can be included in the exclude list for class A. When calculating the score for a candidate keyword, the exclude list should also be taken into account in addition to the seed keyword list. That is, a candidate word has to be more similar to the seeds than words in the exclude list to be considered a potential keyword for the class. Similar to the continuous expansion of the seed keyword list during each iteration round, the exclude list can also be expanded iteratively using, for instance, words that are most similar to those already in the exclude list.

This strategy can also address the challenge regarding the trade-off between precision and exhaustiveness. Taking our previous observations as an example, if an initial top 400 keyword selection contains only 200 relevant terms, and these are not exclusively found within the top 200, a conservative approach that prioritizes precision might involve selecting only the top 100 keywords to minimize the inclusion of irrelevant terms. However, with the implementation of an exclude list, the irrelevant terms within the top 400 can be pre-emptively filtered out, thereby enabling a more comprehensive selection of the top 200 keywords without sacrificing precision.

Error Pruning in Iteration

As we continue to refine our approach through iterations, it is critical to not only build upon the knowledge accumulated but also to actively identify and rectify errors that could propagate and adversely affect the results. Unlike the forward-building process of keyword extraction, error pruning operates in a retroactive manner. It reviews the keywords identified in each iteration, assessing their relevance and fit to a specific class. This process is analogous to pruning a tree, where unnecessary branches that have deviated too far from the main structure are removed. By incorporating error pruning into the iterative process, we can maintain the integrity of the keyword extraction process, prevent the accumulation of irrelevant keywords, and ensure a more accurate and robust

outcome. This pruning mechanism also helps balance the trade-off between comprehensiveness and precision. By actively identifying and removing these erroneous branches through error pruning, we can ensure that the keyword selection remains comprehensive while minimizing the inclusion of irrelevant terms, thus enabling the selection of a larger, more comprehensive set of keywords without sacrificing precision.

Filtering of Generated Synonyms

Based on the findings of Subsection 5.3.2, a filtering mechanism for the synonym generation process can be introduced, including but not limited to the following improvements:

1. Instead of selecting the "best" synonym set in case of a polysemous word, consider the synonyms from multiple synsets and potentially incorporate them into the class's keyword set.

2. Consider an intra-synset filtering mechanism. Even in cases where only one synset is available for a given word, it could be beneficial to assess the similarity of its components to the class context first before adding them to the keywords.

6.3.2. Context Integration in Embedding Process

The limitations associated with the lack of contextualization during the keyword extraction and generation processes can be addressed by incorporating contextual information. Future work should explore strategies for meaningfully incorporating context into the embedding process. This includes investigating which types of context should be included and determining the methods for their integration in order to improve the accuracy and robustness of the processes.

For instance, instead of embedding the seed keywords as a list of isolated words, they can be incorporated into a context such as the predefined class descriptions. A similar approach could be applied during the synonym generation step for evaluating the synsets. The synonyms' similarity to the seed keywords and the target word could be computed by embedding them in the same context rather than as standalone words. These approaches should allow the model to consider the relationship between words in a particular class context, potentially leading to more accurate similarity scores and higher class-specificity.

Additionally, it would be essential to evaluate the impact of these changes on the overall performance of the pipeline. This could involve a comparison of the performance with and without contextual embeddings, assessed through various methods such as qualitative analysis by domain experts using evaluation metrics like coherence and precision.

7. Conclusion

The essence of this thesis lies in the exploration and development of a systematic approach to yield domain-specific keywords from unstructured data, ensuring that the derived keyword sets are both meaningful and relevant.

Our extraction approach leverages short textual descriptions and class-specific seed keywords from the WZ2008 classification, validated by domain experts, to extract keywords from the German Business Registry data set. We enrich the diversity of the keywords by iteratively adding new seeds to the extraction. Furthermore, our approach preserves class-specificity of the extracted keywords by incorporating a filter that prioritizes precision over exhaustiveness.

The subsequent generation process, devoid of any external knowledge bases, builds upon the extracted class-specific keywords. The objective is to produce a more comprehensive set of keywords encompassing those that might have been missed during extraction or were absent from the data set. This generation process is divided into three stages: lexical substitution, fortified with a filtering mechanism, ensures class-relevance, synonym generation further broadens the keyword spectrum, and the final word form generation allows for the inclusion of various grammatical constructs.

Every choice in our methodology is the result of extensive experimentation and careful parameter selection in pursuit of the highest quality of results. However, we also recognize the diverse needs of potential users. Thus, while carefully crafted, our pipeline is also designed with flexibility in mind, leaving many options configurable to cater to diverse requirements.

We employed both qualitative and quantitative evaluation approaches to assess the efficacy of our methodology. While the overall extraction process has demonstrated high class-specificity, certain classes posed challenges, pinpointing areas for further refinement. The generation phase, despite being designed to enrich the keyword set, introduced inconsistencies into the existing keywords, especially during the synonym generation stage. This highlights the need for future improvements in generation approaches, potentially through integrating a more rigorous filtering mechanism.

The principle of domain knowledge guidance in our research offers insightful implications. Our systematic keyword pipeline is, to the best of our knowledge, the first methodical approach that integrates domain knowledge in both keyword extraction and generation in such a comprehensive manner. While many previous approaches lean towards either full automation or manual extraction, our pipeline strikes a balance. It offers the efficiency of automation, completing extraction at rates substantially superior to manual methods, yet retains the precision and relevance that domain knowledge brings. Furthermore, the generalizability of our pipeline means it can be applied across

a wide range of domains, offering a high degree of adaptability. The fusion of domain knowledge with automation introduces a fresh perspective that has not been extensively investigated in conventional keyword extraction methodologies, presenting potential for exploration in future research.

In summary, this thesis underscores the benefits of combining domain knowledge and automated processes for extracting and generating keywords from unstructured data. The challenges encountered, insights gained, and solutions devised throughout this research pave the way for future explorations and advancements in this domain.

A. WZ2008 Sections with Manually Constructed Class Descriptions and Seed Keywords

Table A.1 presents a detailed overview of the WZ2008 sections. We refined the class descriptions based on the original section descriptions and their summaries (see Subsection 5.1.1), and manually defined the seed keywords using each section's description and outline (see Subsection 4.5.1).

Table A.1.: WZ2008 Sections with our defined class descriptions and seed keywords

Section	Name	Refined Class Description	Defined Seed Keywords
A	Land- und Forstwirtschaft, Fischerei	Dieser Abschnitt umfasst die Nutzung der pflanzlichen und tierischen natürlichen Ressourcen. Dazu zählen Tätigkeiten wie Pflanzenbau, Tierzucht und Tierhaltung, Holzgewinnung und die Gewinnung anderer pflanzlicher und tierischer Erzeugnisse in landwirtschaftlichen oder forstwirtschaftlichen Betrieben oder in freier Natur.	Landwirtschaft, Forstwirtschaft, Pflanzenbau, Tierzucht, Tierhaltung, Jagd, Holzgewinnung, Holzeinschlag, Veredlung landwirtschaftlicher Erzeugnisse, Fischerei, Aquakultur, Anbau von Pflanzen, Fallenstellerei, Fischzucht
B	Bergbau und Gewinnung von Steinen und Erden	Dieser Abschnitt umfasst die Gewinnung natürlich vorkommender fester (Kohle und Erze), flüssiger (Erdöl) und gasförmiger (Erdgas) mineralischer Rohstoffe. Er umfasst auch zusätzliche Tätigkeiten zur Aufbereitung von Rohstoffen für den Absatz, z. B. Zerkleinern, Mahlen, Waschen, Sortieren, Konzentration von Erzen, Verflüssigung von Erdgas und Agglomeration von festen Brennstoffen.	Bergbau, Gewinnung von Steinen und Erden, Kohlenbergbau, Erzbergbau, Erdgas, mineralische Rohstoffe, Aufbereitung von Rohstoffen, Aufbereitung von Erzen, Minerale, Mineralen, Gewinnung von Erdöl, Gewinnung von Erdgas, Gewinnung von Steinen, Gewinnung von Erden
C	Verarbeitendes Gewerbe	Dieser Abschnitt umfasst die mechanische, physikalische oder chemische Umwandlung von Stoffen oder Teilen in Waren. Die wesentliche Änderung oder Neugestaltung von Waren wird generell als Herstellung von Waren angesehen. Das Ergebnis des Herstellungsverfahrens sind entweder Fertigwaren für den Gebrauch oder Verbrauch und Halbwaren zur weiteren Be- oder Verarbeitung. Das Zusammenbauen der Teile von Waren gilt ebenfalls als Herstellung von Waren. Zu diesem Abschnitt gehört auch die spezialisierte Wartung, Reparatur und Installation von Maschinen und Ausrüstungen.	Be- und Verarbeitung, Herstellung, Verarbeitung, Verarbeitung der gewonnenen Rohstoffe, Flaschenabfüllung von natürlichem Quell- und Mineralwasser, Fischverarbeitung, Milchverarbeitung, Obstverarbeitung, Gemüseverarbeitung, Pasteurisieren, Abfüllen von Milch, Lederveredlung, Holzimprägnierung, Runderneuerung von Reifen, Elektroplattieren, Plattieren, Wärmebehandlung von Metallen, Umbau von Maschinen, Grundüberholung von Maschinen, Herstellung von Nahrungsmitteln, Herstellung von Futtermitteln, Getränkeherstellung, Tabakverarbeitung, Herstellung von Textilien, Spinnerei, Weberei, Herstellung von Bekleidung, Schneiderarbeiten, Herstellung von Schuhen, Herstellung von Leder, Herstellung von Holzwaren, Papierherstellung, Herstellung von Druckerzeugnissen, Kokerei, Mineralölverarbeitung, Herstellung von chemischen Erzeugnissen, Herstellung von pharmazeutischen Erzeugnissen, Herstellung von Kunststoffwaren, Herstellung von Glas und Glaswaren, Herstellung von Keramik, Verarbeitung von Steinen und Erden, Metallerzeugung, Metallbearbeitung, GieSSereien, Herstellung von Metallerzeugnissen, Maschinenbau, Herstellung von Kraftwagen, Fahrzeugbau, Herstellung von Möbeln, Reparatur und Installation von Maschinen und Ausrüstungen
D	Energieversorgung	Dieser Abschnitt umfasst die Elektrizitätsversorgung, Gasversorgung, Wärmeversorgung und Warmwasserversorgung u. Ä. durch ein fest installiertes Netz von Stromleitungen bzw. Rohrleitungen. Eingeschlossen ist auch die Versorgung von Industriegebieten und Gewerbegebieten, sowie von Wohngebäuden. Unter diesen Abschnitt fällt der Betrieb von Anlagen, die Elektrizität oder Gas erzeugen und verteilen bzw. deren Erzeugung und Verteilung überwachen. Ebenfalls eingeschlossen ist die Wärmeversorgung und Kälteversorgung.	Energieversorgung, Elektrizitätsversorgung, Gasversorgung, Warmwasserversorgung, Wärmeversorgung, Kälteversorgung, Energieerzeugung, Energieverteilung

E	Wasser-versorgung; Abwasser- und Abfallentsorgung und Beseitigung von Umweltver-schmutzungen	Dieser Abschnitt umfasst Tatigkeiten im Zusammenhang mit der Entsorgung (Sammlung, Behandlung und Beseitigung) verschiedener Abfalle, wie z. B. fester oder nicht fester Abfalle aus Industrie, Gewerbe oder Haushalten, sowie die Sanierung von Altlasten. Auch Tatigkeiten der Wasserversorgung fallen unter diesen Abschnitt, da sie haufig entweder in Verbindung mit der Abwasserbehandlung durchgefuhrt werden oder von Einheiten erbracht werden, die auch mit der Abwasserbehandlung befasst sind.	Wasserversorgung, Abwasserentsorgung, Abfallentsorgung, Umweltschmutzungen, Sammlung von Abfällen, Abfallbehandlung, Abfallbeseitigung, Rückgewinnung
F	Baugewerbe	Dieser Abschnitt umfasst allgemeine und spezialisierte Hoch- und Tiefbautätigkeiten. Dazu zählen Neubau, Instandsetzung, An- und Umbau, die Errichtung von vorgefertigten Gebäuden oder Bauwerken auf dem Baugelände sowie provisorischer Bauten. Es handelt sich um die Errichtung von Gebäuden einerseits sowie von Autobahnen, StraSSen, Brücken, Tunneln, Häfen, Bewässerungsanlagen, Kanalisationen, Industrieanlagen, Rohrleitungen usw. andererseits. Ebenfalls eingeschlossen sind die Renovierung von Gebäuden und Ingenieurbauten. Dieser Abschnitt umfasst den vollständigen Bau von Gebäuden und von Tiefbauten sowie spezialisierte Bautätigkeiten.	Baugewerbe, Hochbau, Bau von Gebäuden, Tiefbau, Wasserbau, Neubau, Instandsetzung, Anbau, Umbau, Errichtung von Bauten, Ausführung von Herstellungstätigkeiten auf der Baustelle, Errichtung, Renovierung, Tiefbauten, Bautätigkeiten, Bauinstallation, Ausbau, vorbereitende Baustellenarbeiten, Gebäudeinstallation, Bauausführung, Gebäudefertigstellung
G	Handel; Instand-haltung und Reparatur von Kraftfahrzeugen	Dieser Abschnitt umfasst den GroSShandel und Einzelhandel (d. h. Verkauf ohne Weiterverarbeitung) mit jeder Art von Waren und die Erbringung von Dienstleistungen beim Verkauf von Waren. Der Abschnitt umfasst auSSerdem die Instandhaltung und Reparatur von Kraftfahrzeugen.	Handel, GroSShandel, Einzelhandel, Instandhaltung von Kraftfahrzeugen, Reparatur von Kraftfahrzeugen, Kraftfahrzeugreparatur, Handel mit Kraftfahrzeugen, Kraftfahrzeughandel, Handelsvermittlung, Verpacken, Umverpacken, Abfüllen von Erzeugnissen wie Spirituosen oder Chemikalien, Abfallsortieren, Sortieren von Waren, Zusammenstellen von Waren, Mischen von Waren, Mischen von Farben, Schneiden von Metallen, Import, Export, Verkauf, Wiederverkauf
H	Verkehr und Lagerei	Dieser Abschnitt umfasst die Personenbeförderung und Güterbeförderung im Linien- oder Gelegenheitsverkehr auf Schienen, in Rohrfernleitungen, auf der StraSSe, zu Wasser und in der Luft sowie damit verbundene Tätigkeiten wie Betrieb von Bahnhöfen, Häfen und Flughäfen, Parkplätzen und Parkhäusern sowie Frachtumschlag, Lagerei usw. Eingeschlossen sind auch die Vermietung von Fahrzeugen mit Fahrer oder Bedienungspersonal sowie Post-, Kurier- und Expressdienste.	Verkehr, Lagerei, Landverkehr, Transport in Rohrfernleitungen, Eisenbahnverkehr, StraSSenverkehr, Taxis, Schiffahrt, Luftfahrt, Postdienste, Kurierdienste, Expressdienste
I	Gastgewerbe	Dieser Abschnitt umfasst die kurzzeitige Gewährung von Unterkunft sowie die Bereitstellung von kompletten Mahlzeiten und von Getränken zum in der Regel sofortigen Verzehr. Beispiele für Unterkunft beinhaltet Beherbergung wie Hotels, Gasthöfe und Pensionen sowie andere Ferienunterkünfte. Unter Bereistellung von kompletten Mahlzeiten versteht man Gastronomie wie Restaurants, Gaststätten, Cafés usw.	Beherbergung, Hotels, Gastronomie, Restaurants, Gasthöfe, Pensionen, Ferienunterkünfte, Ferienhäuser, Jugendherberge, Campingplätze, Zubereitung von Nahrungsmitteln zum sofortigen Verzehr an Ort und Stelle, Gaststätten, Imbisstuben, Cafés, Eissalons, Caterer, Bars, Ausschank von Getränken
J	Information und Kommunikation	Dieser Abschnitt umfasst die Herstellung und den Vertrieb von Informations- und kulturellen Angeboten, die Bereitstellung der Mittel zur Übertragung und Verteilung dieser Produkte, einschlieSSlich der Datenübertragung und zur Kommunikation, Tätigkeiten im Bereich der Informationstechnologie, die Verarbeitung von Daten und andere Informationsdienstleistungen. Unter diesen Abschnitt fallen: das Verlagswesen, einschlieSSlich des Verlegens von Software; die Herstellung von Filmen und von Tonaufnahmen sowie das Verlegen von Musik; die Herstellung und Ausstrahlung von Fernseh- und Hörfunkprogrammen; die Telekommunikation; Dienstleistungen der Informationstechnologie und sonstige Informationsdienstleistungen.	Datenverarbeitung, Informationstechnologie, Kommunikation, Verlagswesen, Verlegen von Büchern, Verlegen von Zeitungen, Verlegen von Zeitschriften, Verlegen von Software, Filmen, Fernsehprogrammen, Kinos, Tonstudios, Verlegen von Musik, Rundfunkveranstalter, Hörfunkveranstalter, Fernsehveranstalter, Telekommunikation, Programmierung, Softwareentwicklung, Datenübertragung, Informationsdienstleistungen, Datenverarbeitung, Webportale, Korrespondenzbüros, Nachrichtenbüros
K	Erbringung von Finanz- und Versicherungsdien-stleistungen	Dieser Abschnitt umfasst die Erbringung von Finanzdienstleistungen einschlieSSlich Versicherungs- und Rückversicherungsdienstleistungen, die Tätigkeit von Pensionskassen und Pensionsfonds sowie mit Finanzdienstleistungen verbundene Tätigkeiten. Dieser Abschnitt umfasst auch das Halten von Vermögenswerten, z. B. die Tätigkeit von Holding- oder Treuhandgesellschaften, Fonds und ähnlichen Finanzinstitutionen.	Finanzdienstleistungen, Versicherungsdienstleistungen, Finazierungsinstitutionen, Zentralbanken, Kreditinstitute, Beteiligungsgesellschaften, Versicherung, Rückversicherung, Pensionkasse, Pensionfonds, Effektenbörsen, Warenbörsen, Risikobewertung, Schadensbewertung, Fondsmanagement, Treuhandgesellschaften
L	Grundstücks- und Wohnungswesen	Dieser Abschnitt umfasst den Kauf und Verkauf von Grundstücken, Gebäuden und Wohnungen, die Vermietung von Grundstücken, Gebäuden und Wohnungen, die Erbringung sonstiger Dienstleistungen im Zusammenhang mit Grundstücken, Gebäuden und Wohnungen, z. B. Schätzung von Grundstücken, Gebäuden und Wohnungen oder die Tätigkeit als Treuhänder von Grundstücken, Gebäuden und Wohnungen. Dieser Abschnitt umfasst auch die Errichtung von Bauwerken, wenn der Errichter Eigentümer der Gebäude bleibt und sie vermietet. Zu diesem Abschnitt gehört auch die Tätigkeit von Hausverwaltungen.	Grundstücken, Gebäuden, Wohnungen, Immobilien, Vermietung von Grundstücken, Vermietung von Gebäuden, Errichtung von Bauwerken, Hausverwaltung, Kauf und Verkauf von Grundstücken, Vermittlung von Grunstücken
M	Erbringung von freiberuflichen, wissenschaftlichen und technischen Dienstleistungen	Dieser Abschnitt umfasst bestimmte freiberufliche, wissenschaftliche und technische Tatigkeiten. Diese Tatigkeiten erfordern ein hohes MaSS an Ausbildung und stellen den Nutzern Fachkenntnisse und Erfahrungen zur Verfugung. Beispiele sind Rechtsberatung, Steuerberatung, Wirtschaftsprüfung und Unternehmensberatung sowie Verwaltung und Führung von Unternehmen und Betrieben.	Rechtsberatung, Steuerberatung, Wirtschaftsprüfung, Buchführung, Unternehmensberatung, Unternehmensverwaltung, Verwaltung von Unternehmen, Führung von Unternehmen, Public-Relations-Beratung, Management, Architekturbüros, Ingenieurbüros, technische Untersuchung, physikalische Untersuchung, chemische Untersuchung, Vermessungsbüros, Forschung und Entwichlung, Werbung, Marktforschung, Meinungsforschung, Ateliers, Design, Fotografie, Fotolabor, Übersetzen, Dolmetschen, Veterinärwesen, Tierarzt

N	Erbringung von sonstigen wirtschaftlichen Dienstleistungen	Dieser Abschnitt umfasst eine Vielzahl von Tätigkeiten zur Unterstützung der allgemeinen Geschäftstätigkeit, deren Hauptzweck nicht im Transfer von Fachwissen besteht. Beispiele sind Vermietung von beweglichen Sachen wie Kraftfarzeugen; Vermittlung und Überlassung von Arbeitskräfte; Reisebüros, Wach- und Sicherheitsdienste sowie Detekteien; Gebäudebetreuung; Garten- und Landschaftsbau; Copy-Shops, Call Center; Messe-, Ausstellungs- und Kongressveranstlater sowie Versteigerungsgewerbe.	Vermietung von Kraftwagen, Vermietung von Sportsgeräten, Vermietung von Gebrauchsgütern, Videotheken, Vermietung von Machinen und Geräten, Vermittlung von Arbeitskräften, Überlassung von Arbeitskräften, Reisebüros, Reiseveranstalter, Reservierungsdienstleistungen, Wachdienste, Sicherheitsdienste, Detekteien, Gebäudebetreuung, Hausmeisterdienste, Gebäudereinigung, Gartenbau, Landschaftsbau, Sekretariatsdienste, Schreibdienste, Copy-Shops, Call Center, Messeveranstalter, Ausstellungsveranstalter, Kongressveranstalter, Inkassobüros, Auskunfteien, Versteigerungsgewerbe
O	Öffentliche Verwaltung, Verteidigung; Sozialversicherung	Dieser Abschnitt umfasst die Tätigkeiten hoheitlicher Natur, die normalerweise von der öffentlichen Verwaltung ausgeführt werden. Dazu gehören das Erlassen und die juristische Auslegung von Gesetzen und daraus resultierenden Vorschriften sowie die Verwaltung von Programmen, die auf ihnen beruhen, Gesetzgebungstätigkeiten, Steuerverwaltung, Verteidigung, öffentliche Sicherheit und Ordnung, Einwanderungsdienste, auswärtige Angelegenheiten und die Verwaltung von Regierungsprogrammen. Dieser Abschnitt umfasst ferner die gesetzliche Sozialversicherung.	Öffentliche Verwaltung, Verteidigung, Sozialversicherung, Wirtschaftsförderung, Wirtschaftsordnung, Feuerwehren, Rechtspflege, Gesetzgebungstätigkeiten, Steuerverwaltung, öffentliche Sicherheit und Ordnung, Einwanderungsdienste
P	Erziehung und Unterricht	Dieser Abschnitt umfasst Erziehung und Unterricht auf allen Stufen und für alle Berufe. Der Unterricht kann mündlich oder schriftlich, über Hörfunk, Fernsehen, Internet oder als Fernkurs erteilt werden. Der Abschnitt umfasst sowohl den Unterricht in den verschiedenen Lehranstalten des regulären Schulsystems auf den verschiedenen Stufen (erster Bildungsweg) als auch Erwachsenenbildung, Alphabetisierungsprogramme usw. Eingeschlossen sind auch die verschiedenen Stufen von Militärschulen und -akademien, Gefängnisschulen usw. Die Klassen umfassen auf jeder Stufe des ersten Bildungsweges auch den Sonderunterricht für körperlich oder geistig behinderte Schüler. Dieser Abschnitt umfasst ferner die Erteilung von Unterricht überwiegend in sportlichen und Freizeitaktivitäten wie Tennis- oder Golfkurse und die Erbringung von Dienstleistungen für den Unterricht.	Erziehung, Unterricht, Schulen, Kindergärten, Vorschulen, Grundschulen, Fachhochschulen, Berufsakademien, Universitäten, Fachakademien, Sportunterricht, Freizeitunterricht, Kulturunterricht, Fahrschulen, Flugschulen, Erwachsenenbildung, Alphabetisierungsprogramme, Militärschulen, Gefängnisschulen, Sonderunterricht
Q	Gesundheits- und Sozialwesen	Dieser Abschnitt umfasst die Erbringung von Dienstleistungen des Gesundheits- und Sozialwesens. Die Tätigkeiten reichen von der medizinischen Versorgung durch medizinische Fachkräfte in Krankenhäusern und anderen Einrichtungen über stationäre Pflegeleistungen mit einem gewissen Anteil an medizinischer Versorgung bis hin zu Tätigkeiten des Sozialwesens ohne Beteiligung medizinischer Fachkräfte.	Gesundheitswesen, Sozialwesen, medizinische Versorgung, Krankenhäuser, Kliniken, Hochschulkliniken, Vorsorgekliniken, Rehabilitationskliniken, Arztpraxen, Zahnarztpraxen, Facharztpraxen, Heime, Pflegeheime, Altenheime, Behindertenwohnheime, Tagesbetreuung, soziale Betreuung
R	Kunst, Unterhaltung und Erholung	Dieser Abschnitt umfasst Tätigkeiten, die die verschiedenen kulturellen, Unterhaltungs- und Freizeitinteressen der breiten Öffentlichkeit abdecken, einschlieSSlich Durchführung von Liveauftritten, Betrieb von Museen, Spiel-, Wett- und Lotteriewesen, sportliche und Freizeitaktivitäten.	Kunst, Unterhaltung, Erholung, darstellende Kunst, Freizeitaktivitäten, Theater, Ballett, Orchester, Kapellen, Chören, Zirkus, Künstler, Theaterveranstalter, Konzertveranstalter, Opernhäuser, Schauspielhäuser, Konzerthallen, Bibliotheken, Archive, Museen, botanische Gärten, zoologische Gärten, Naturparks, Spielwesen, Wettwesen, Lotteriewesen, Betrieb von Sportanlagen, Sportvereine, Fitnesszentren, Vergnügungsparks, Themenparks
S	Erbringung von sonstigen Dienstleistungen	Dieser Abschnitt umfasst die Tatigkeiten von Interessenvertretungen, die Reparatur von Datenverarbeitungsgeraten und Gebrauchsgutern und eine Vielzahl von in dieser Klassifikation anderweitig nicht erfassten personlichen und anderen Dienstleistungen.	Interessenvertretungen, Wirtschaftsverbände, Arbeitgeberverbände, Berufsorganisationen, Arbeitnehmervereinigungen, kirchliche Vereinigungen, religiose Vereinigungen, politische Parteien, Vereinigungen, Reparatur von Datenverarbeitungsgeräten, Reparatur von Gebrauchsgütern, Wäscherei, chemische Reinigung, Frisörsalons, Kosmetiksalons, Sauna, Bäder, Bestattungswesen
T	Private Haushalte mit Hauspersonal; Herstellung von Waren und Erbringung von Dienstleistungen durch private Haushalte fur den Eigenbedarf ohne ausgepragten Schwerpunkt	Abteilung "Private Haushalte mit Hauspersonal" umfasst die Tätigkeit von Haushalten in ihrer Eigenschaft als Arbeitgeber von Hauspersonal wie Dienstmädchen, Kellner, Diener, Köchinnen, Köche, Wäscherinnen, Wäscher, Gärtnerinnen, Gärtner, Pförtnerinnen, Pförtner, Stallgehilfen, Fahrer, Hausmeisterinnen, Hausmeister, Erzieherinnen, Erzieher, Babysitter, Hauslehrerinnen, Sekretäre usw. beschäftigen. Abteilung "Herstellung von Waren und Erbringung von Dienstleistungen durch private Haushalte für den Eigenbedarf ohne ausgeprägten Schwerpunkt" umfasst die Herstellung von Waren und die Erbringung von Dienstleistungen durch private Haushalte für den Eigenbedarf ohne ausgeprägten Schwerpunkt.	Private Haushalte, Hauspersonal, privater Bedarf, Eigenbedarf, Dienstmädchen, private Kellner, Diener, private Köchinnen, private Köche, Wäscherinnen, Wäscher, Gärtnerinnen, Gärtner, Pförtnerinnen, Pförtner, Stallgehilfen, Fahrerinnen, Fahrer, Hausmeisterinnen, Hausmeister, Erzieherinnen, Erzieher, Babysitter, Hauslehrerinnen, Hauslehrer, Sekretärinnen, Sekretäre
U	Exterritoriale Organisationen und Korperschaften	Diese Abschnitt umfasst: • Tätigkeiten internationaler Organisationen wie der Vereinten Nationen und ihrer Sonder- oder Regionalorganisationen usw., des Internationalen Währungsfonds, der Weltbank, der Weltzollorganisation, der Organisation für Wirtschaftliche Zusammenarbeit und Entwicklung, der Organisation Erdöl exportierender Länder, der Europäischen Gemeinschaften, der Europäischen Freihandelsassoziation usw. • Tätigkeiten von diplomatischen und konsularischen Vertretungen fremder Staaten.	internationale Organisationen, Vereinten Nationen, Internationaler Währungsfond, Weltbank, Weltzollorganisation, Organisation für Wirtschaftliche Zusammenarbeit und Entwicklung, Europäische Gemeinschaften, Europäische Freihandelsassoziation, diplomatische Vertretung, konsularische Vertretungen

.

B. Evaluation Surveys

B.1. Questionnaire for Expert Validation of Extracted Keywords

Below are the instructions and questions presented to the domain experts for the validation of the extracted keywords (Subsection 4.7.1).

B.1.1. Instructions

The following text was displayed to the domain experts at the beginning of the survey. It serves as a short description of the evaluation approach and provides instructions for completing the survey.

> Diese Studie dient der Evaluierung unseres modifizierten Schlüsselwort- Extraktionsansatzes für den Handelsregister-Datensatz. Wir haben 5 der 21 Abschnitte aus dem WZ2008 Dokument für die Evaluierung ausgewählt. Für jeden dieser Abschnitte sind die Top-Keywords, die durch unseren modifizierten Ansatz extrahiert wurden, unten aufgeführt.
>
> Die sogenannten "Beispiel-Schlüsselwörter" dienten als Ausgangspunkt für die Extraktion, wobei unser Ansatz in der Regel Schlüsselwörter extrahiert, die eine hohe semantische Ähnlichkeit zu diesen haben. Beachten Sie jedoch, dass die extrahierten Schlüsselwörter nicht unbedingt direkt einem der Beispiel-Schlüsselwörter entsprechen müssen.
>
> Bitte beachten Sie: Rechtschreibfehler oder ungültige Wörter werden trotzdem als themenrelevant angesehen, solange die tatsächliche Bedeutung der korrigierten Version korrekt ist.
>
> **Bitte markieren Sie alle Wörter, die Ihrer Meinung nach NICHT zu dem jeweiligen Abschnitt gehören**, basierend auf den Abschnittsbeschreibungen und den Beispiel-Schlüsselwörtern

B.1.2. Questions

Prior to presenting the questions, a short textual description and some seed keywords were provided as context for each class under evaluation. Participation in each question was optional. Domain experts were instructed to select all options they deemed unfitting to the class. Conversely, they could also abstain from making any selections if they believed all options were specific to the class. In the following, we present the exact format and content for these questions.

Abschnitt A: Land- und Forstwirtschaft, Fischerei

Abschnittsbeschreibung: Dieser Abschnitt umfasst die Nutzung der pflanzlichen und tierischen natürlichen Ressourcen. Dazu zählen Tätigkeiten wie Pflanzenbau, Tierzucht und Tierhaltung, Holzgewinnung und die Gewinnung anderer pflanzlicher und tierischer Erzeugnisse in land- oder forstwirtschaftlichen Betrieben oder in freier Natur.

Ausgangsschlüsselwörter, um die Extraktion zu steuern:

> Landwirtschaft, Anbau von Pflanzen, Forstwirtschaft und Holzeinschlag, Holzgewinnung, Jagd und Fallenstellerei, Fischerei und Aquakultur, Fischzucht, Tierhaltung, Tierzucht, Veredlung landwirtschaftlicher Erzeugnisse

Bitte wählen Sie aus der folgenden Liste alle Wörter aus, die Ihrer Meinung nach **NICHT** zu Abschnitt A (Land- und Forstwirtschaft, Fischerei) gehören.

- ☐ agrarerzeugnissen
- ☐ agrarrohstoffen
- ☐ agrarstrukturverbesserung
- ☐ agrarwirtschaft
- ☐ agronomie
- ☐ forst
- ☐ forstservice
- ☐ forstwirtschaftlich
- ☐ friedhofsgärtnereien
- ☐ gartenanlagen
- ☐ gartenarbeit
- ☐ grünanlagenpflege
- ☐ gärten
- ☐ gärtnerischen
- ☐ landwirtschaftlich
- ☐ landwirtschaftsgestaltung
- ☐ pflanze
- ☐ viehhandlung
- ☐ viehzucht

Abschnitt B: Bergbau und Gewinnung von Steinen und Erden

Abschnittsbeschreibung: Dieser Abschnitt umfasst die Gewinnung natürlich vorkommender fester (Kohle und Erze), flüssiger (Erdöl) und gasförmiger (Erdgas) mineralischer Rohstoffe. Er umfasst auch zusätzliche Tätigkeiten zur Aufbereitung von Rohstoffen für den Absatz, z. B. Zerkleinern, Mahlen, Waschen, Sortieren, Konzentration von Erzen, Verflüssigung von Erdgas und Agglomeration von festen Brennstoffen.

Ausgangsschlüsselwörter, um die Extraktion zu steuern:

> Aufbereitung von Erzen und Rohstoffen, Bergbau, Erzbergbau, Erdgas, Gewinnung von Erdgas und Erdöl, Gewinnung von Steinen und Erden, Kohlenbergbau, Minerale, mineralische Rohstoffe

Bitte wählen Sie aus der folgenden Liste alle Wörter aus, die Ihrer Meinung nach **NICHT** zu Abschnitt B (Bergbau und Gewinnung von Steinen und Erden) gehören.

- ☐ agrarrohstoffen
- ☐ baugütern
- ☐ bergbaubedarfsartikeln
- ☐ bergbaus
- ☐ bergwerken
- ☐ erdbauleistungen
- ☐ erdstoffen
- ☐ erdöl
- ☐ gärsubstraten
- ☐ industriebau
- ☐ industrieflächen
- ☐ industriemineralien
- ☐ minerals
- ☐ mineralölerzeugnissen
- ☐ rohstoffe
- ☐ rohstoffmärkte
- ☐ rohstoffwirtschaft
- ☐ stahlerzeugnissen
- ☐ stahlwerke

Abschnitt M: Erbringung von freiberuflichen, wissenschaftlichen und technischen Dienstleistungen

Abschnittsbeschreibung: Dieser Abschnitt umfasst bestimmte freiberufliche, wissenschaftliche und technische Tätigkeiten. Diese Tätigkeiten erfordern ein hohes Maß an Ausbildung und stellen den Nutzern Fachkenntnisse und Erfahrungen zur Verfügung.

Ausgangsschlüsselwörter, um die Extraktion zu steuern:

> Architektur- und Ingenieurbüros, Ateliers, Dolmetschen und Übersetzen, Forschung und Entwicklung, Fotografie und Fotolabor, Public-Relations-Beratung, Rechts- und Steuerberatung, technische, chemische und physikalische Untersuchung, Tierarzt und Veterinärwesen, Verwaltung und Führung von Unternehmen, Unternehmensberatung, Werbung, Markt- und Meinungsforschung, Wirtschaftsprüfung und Steuerberatung, Buchführung

Bitte wählen Sie aus der folgenden Liste alle Wörter aus, die Ihrer Meinung nach **NICHT** zu Abschnitt M (Erbringung von freiberuflichen, wissenschaftlichen und technischen Dienstleistungen) gehören.

- ☐ beratungsleistungen
- ☐ betriebliches
- ☐ betriebsberatung
- ☐ consultingleistungen
- ☐ firmen
- ☐ geschäftsführungs
- ☐ geschäftsleitung
- ☐ ingenieurbüro
- ☐ managementund
- ☐ untemehmensberatung
- ☐ unternehmensabläufen
- ☐ unternehmensberatungsleistung
- ☐ unternehmensentwicklung
- ☐ unternehmenstand
- ☐ unternehmenver
- ☐ verwaltungsgesellschaft
- ☐ wirtschaftsberatenden
- ☐ wirtschaftsberatung

Abschnitt P: Erziehung und Unterricht

Abschnittsbeschreibung: Dieser Abschnitt umfasst Erziehung und Unterricht auf allen Stufen und für alle Berufe. Der Abschnitt umfasst sowohl den Unterricht in den verschiedenen Lehranstalten des regulären Schulsystems auf den verschiedenen Stufen (erster Bildungsweg) als auch Erwachsenenbildung, Alphabetisierungsprogramme usw. Eingeschlossen sind auch die verschiedenen Stufen von Militärschulen und -akademien, Gefängnisschulen usw. Die Klassen umfassen auf jeder Stufe des ersten Bildungsweges auch den Sonderunterricht für körperlich oder geistig behinderte Schüler. Dieser Abschnitt umfasst ferner die Erteilung von Unterricht überwiegend in sportlichen und Freizeitaktivitäten wie Tennis- oder Golfkurse und die Erbringung von Dienstleistungen für den Unterricht.

Ausgangsschlüsselwörter, um die Extraktion zu steuern:

> Erziehung und Unterricht, Alphabetisierungsprogramme, Kindergärten und Vorschulen, Grundschulen, Schulen, Universitäten, Fachakademien, Fachhochschulen, Berufsakademien Erwachsenenbildung, Fahr- und Flugschulen, Gefängnisschulen, Militärschulen, Sonderunterricht, Kulturunterricht, Sport- und Freizeitunterricht

Bitte wählen Sie aus der folgenden Liste alle Wörter aus, die Ihrer Meinung nach **NICHT** zu Abschnitt P (Erziehung und Unterricht) gehören.

- ☐ akademie
- ☐ ausbildungen
- ☐ ausbildungsbetriebe
- ☐ ausbildungsdienstleistungen
- ☐ berufsausbildung
- ☐ bildungsangebot
- ☐ bildungseinrichtung
- ☐ bildungsreisen
- ☐ erziehungsund
- ☐ fahrschule
- ☐ fortbildungen
- ☐ schule
- ☐ schulischen
- ☐ schulung
- ☐ schulungstätigkeit
- ☐ terbildungsmaßnahmen
- ☐ unterrichtung

Abschnitt S: Erbringung von sonstigen Dienstleistungen

Abschnittsbeschreibung: Dieser Abschnitt umfasst die Tätigkeiten von Interessenvertretungen, die Reparatur von Datenverarbeitungsgeräten und Gebrauchsgütern und eine Vielzahl von in dieser Klassifikation anderweitig nicht erfassten persönlichen und anderen Dienstleistungen.

Ausgangsschlüsselwörter, um die Extraktion zu steuern:

> Wirtschafts- und Arbeitgeberverbände, Berufsorganisationen, Arbeitnehmervereinigungen, kirchliche und religiöse Vereinigungen, politische Parteien, Interessenvertretungen, Vereinigungen, Reparatur von Datenverarbeitungsgeräten und Gebrauchsgütern, Bestattungswesen, Sauna, Solarien, Bäder, Wäscherei und chemische Reinigung, Frisör- und Kosmetiksalons

Bitte wählen Sie aus der folgenden Liste alle Wörter aus, die Ihrer Meinung nach **NICHT** zu Abschnitt S (Erbringung von sonstigen Dienstleistungen) gehören.

- ☐ arbeitsgemeinschaften
- ☐ associations
- ☐ gesellschaftsveniiögens
- ☐ gesellschaftern
- ☐ gesellschaftlich
- ☐ gesellschaftsbeteiligungen
- ☐ gesellschaftseigentums
- ☐ gesellschaftszwecken
- ☐ interessengruppen
- ☐ personengesellschaften
- ☐ untergesellschaften
- ☐ unternehmensverbund
- ☐ verbundgesellschaften
- ☐ verbänden
- ☐ vereine
- ☐ zusammenführung
- ☐ zusammenschluß

B.2. Intruder Detection Survey

The following provides a comprehensive list of the five options for each question in the intruder detection task (Subsection 4.7.2). To facilitate a clearer comparison, corresponding questions from the two distinct keyword sets—those solely from extracted keywords and those encompassing both extracted and generated keywords—are presented side by side. The last word in each listed item is the correct "intruder" for that question. During the actual survey, the sequence of the questions and the order of options within each question were randomized for each participant.

Section A

Extracted-Only

1. viehhandlung, landwirtschaften, gartenbedarf, agrarmaschinen, grundstoffen
2. pflanzenzucht, landwirtschaftlichen, gartengeräten, nutzflanzenrassen, unternehmensgegenstands
3. pflanze, agrarmaschinen, gartenbedarf, landwirtschaften, ausbildungsbetriebe
4. ackerbau, kulturpflanzen, agrarrohstoffen, tierhaltung, gemeinschaftskunde
5. bewirtschaftungsleistungen, kultivierung, landwirtschaftsmaschinen, landwirtschaftsgestaltung, telekommunikationsarbeiten

Extracted and Generated

1. viehhandlung, landwirtschaften, fisch, bauer, grundstoffen
2. pflanzenzucht, landwirtschaftlichen, landtechnische, ländliche, unternehmensgegenstands
3. pflanze, agrarmaschinen, waldbaues, spurten, ausbildungsbetriebe
4. ackerbau, kulturpflanzen, viehes, naturschutzes, gemeinschaftskunde
5. bewirtschaftungsleistungen, kultivierung, fischfänge, ackerlands, telekommunikationsarbeiten

Section B

Extracted-Only

6. steinzeugrohren, mineralien, stahlwerke, hartmetallen, agrarmaschinen
7. granitblöcken, steinbruch, baugütern, stahlerzeugnissen, geschäftsführung
8. erdgasnetz, baggerbetrieb, betonsteinwerksarbeiten, kunststeinen, bildungsveranstaltungen
9. steine, rohstoffmärkte, erdstoffen, stahlerzeugnissen, gesellschaftszwecken
10. baugütern, mineralöl, metalle, industriemineralien, ferienanlage

Extracted and Generated

6. steinzeugrohren, mineralien, mineralwassern, sprudelwassers, agrarmaschinen
7. granitblöcken, steinbruch, stahlindustrien, baumaterial, geschäftsführung
8. erdgasnetz, baggerbetrieb, erdreichs, stahlwerks, bildungsveranstaltungen
9. steine, rohstoffmärkte, rohmaterialien, natursteine, gesellschaftszwecken
10. baugütern, mineralöl, öls, untergrund, ferienanlage

Section M

Extracted-Only

11. betriebsverwaltung, unternehmen, beratungsdienstleistung, managementgesellschaft, bepflanzungen
12. betriebsgesellschaft, consulting, firmen, managementgesellschaft, stone
13. managementaufgaben, ingenieurdienstleistung, unternehmensorganisation, managementberatung, trainingsversanstaltungen
14. unternehmensorganisation, betriebsberatungs, unternehmensführung, betriebliche, gesellschaftsbeteiligungen
15. beratungsleistung, unternehmensführung, untrenehmensberatung, betriebsgesellschaft, inland

Extracted and Generated

11. betriebsverwaltung, unternehmen, reklame, promotern, bepflanzungen
12. betriebsgesellschaft, consulting, forschung, verwaltungsapparaten, stone
13. managementaufgaben, ingenieurdienstleistung, ingenieuren, konzept, trainingsversanstaltungen
14. unternehmensorganisation, betriebsberatungs, konstrukteure, stäben, gesellschaftsbeteiligungen
15. beratungsleistung, unternehmensführung, bildern, beraters, inland

Section P

Extracted-Only

16. bildungs, lehrveranstaltungen, schule, schulungen, gartenarbeiten
17. lehreinrichtungen, bildungsbereich, erziehung, schulungsleistungen, betonsteinwerksarbeiten
18. vorschule, bildungsveranstaltungen, bildungsgänge, weiterbildungsmaßnahmen, unternehmen
19. ausbildung, bildung, bildungsstandorten, bildungsbereich, berufsgruppen
20. unterrichtsangebot, lehrgängen, schule, berufsbildung, wirtschaftsangelegenheiten

Extracted and Generated

16. bildungs, lehrveranstaltungen, kultur, kinderhort, gartenarbeiten
17. lehreinrichtungen, bildungsbereich, unterrichtungen, büffels, betonsteinwerksarbeiten
18. vorschule, bildungsveranstaltungen, grundschule, studieren, unternehmen
19. ausbildung, bildung, lernen, kinderbetreuung, berufsgruppen
20. unterrichtsangebot, lehrgängen, kurs, ausbildendes, wirtschaftsangelegenheiten

Section S

Extracted-Only

21. gesellschafter, zusammenschluß, konsortiums, personengesellschaften, gärtnerisch
22. unternehmens, gemeinschaftlicher, gesesellschaft, corporation, natursteinmauern
23. gesellschaftsgegenstand, gemeinschaftskunde, organisations, interessengruppen, wirtschaftsprüfer
24. gesellschaftszwecke, gesellschafter, gemeinschaftanlagen, gesellschaftszwecken, schulungsveranstaltungen
25. gemeinschaftlicher, zusammenführung, bündnis, gesellschafterbeteiligungen, wärmesystem

Extracted and Generated

21. gesellschafter, zusammenschluß, teilnehmers, brötchengebers, gärtnerisch,
22. unternehmens, gemeinschaftlicher, netzwerkes, gemeinschaften, natursteinmauern
23. gesellschaftsgegenstand, gemeinschaftskunde,berufsfeldes, unternehmer, wirtschaftsprüfer
24. gesellschaftszwecke, gesellschafter, paktes, konsortien, schulungsveranstaltungen
25. gemeinschaftlicher, zusammenführung, bade, etwas deutlich machen, wärmesystem

C. Proposed Algorithm

In the following, we present the proposed algorithm for our keyword extraction and generation.

C.1. Code Structure

The structure of the proposed algorithm is organized into three main packages, each serving a distinct purpose:

C.1.1. `keybertmod`

This package is a modification of the KeyBERT algorithm, tailored for keyword extraction from a single document. Within this package, the `KeyBERTMod` class houses the primary method, `extract_keywords`. Compared to the method in the original KeyBERT under the same name, this method has undergone several modifications. We outline the most important ones in the following:

- Introduction of parameters `seed_weight` and `doc_weight` for determining the influence of the seed keywords and the target document during extraction (see Subsection 4.5.4).
- Integration of a score computation mechanism based on the *max seed* approach (see Subsection 4.5.7).
- Restriction of the method's scope to process only a singular document, in contrast to the original which permitted the processing of multiple texts in a list.

C.1.2. `generator`

This package encompasses tools required for the three-step keyword generation (Section 4.6). Each class offers methods that operate on individual instances. The package comprises:

- `_lexsub.py`: Performs lexical substitution on a individual text segment using the `LexSub` class.
- `_syn_gen.py`: Manages synonym generation for a single term with the `SynonymGenerator` class.
- `_wordformgen.py`: Takes care of word form generation for a given word using the `WordFormGen` class.

C.1.3. keyexgen

This is the overarching package that integrates the functionalities of both keybertmod and generator. It contains, among others, two important files: _toolkit.py with the KeyToolkit class, and meta.py with the class KeyExGen.

The KeyToolkit class within the _toolkit.py file acts as an intermediary layer, expanding upon the functionalities provided by both keybertmod and generator. Instead of focusing on singular instances, such as keyword extraction from a single document or lexical substitution for a single text segment, the KeyToolkit class broadens the input scope. Specifically, it allows for:

- Iterative keyword extraction on an entire data set
- Lexical substitution using a combination of different contexts
- Synonym generation and word form generation on sets of words, rather than on individual terms

The operations within the KeyToolkit class correspond to the algorithm previously detailed in Algorithm 4 and Algorithm 2.

The meta.py file introduces a meta-level class named KeyExGen, which operates at the highest abstraction level within the algorithm structure. Its primary role is to sequence the broadened functionalities provided by the toolkit.py into a streamlined pipeline, ensuring a smooth and efficient keyword extraction and generation process. This class has only one method, named keyword_pipeline, which orchestrates the pipeline and aligns with Algorithm 3.

By encapsulating the entire process within a single method, the KeyExGen class ensures that users can execute the complete keyword extraction and generation pipeline with minimal manual intervention, promoting ease of use and efficiency.

C.2. Repository Link

The complete repository can be accessed using the following link:
https://gitlab.lrz.de/CreateData4AI/bachelorthesis-weixin-yan

Acronyms

CD4AI CreateData4AI

NLP Natural Language Processing

WZ2008 Klassifikation der Wirtschaftszweige, Ausgabe 2008

BR Business Registry

NER Named Entity Recognition

POS Part-of-Speech

LSTM Long Short-Term Memory

BERT Bidirectional Encoder Representations from Transformers

List of Figures

List of Tables

Bibliography

[1] A. Aggarwal, C. Sharma, M. Jain, and A. Jain. "Semi Supervised Graph Based Keyword Extraction Using Lexical Chains and Centrality Measures." In: *Computación y Sistemas* 22 (Dec. 2018). DOI: 10.13053/cys-22-4-3077.

[2] F. Al Tarouti and J. Kalita. "Enhancing Automatic Wordnet Construction Using Word Embeddings." In: *Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 30–34. DOI: 10.18653/v1/W16-1204. URL: https://aclanthology.org/W16-1204.

[3] E. Altuncu, J. R. C. Nurse, Y. Xu, J. Guo, and S. Li. *Improving Performance of Automatic Keyword Extraction (AKE) Methods Using PoS-Tagging and Enhanced Semantic-Awareness*. 2022. arXiv: 2211.05031 [cs.CL].

[4] S. Arora, A. May, J. Zhang, and C. Ré. *Contextual Embeddings: When Are They Worth It?* 2020. arXiv: 2005.09117 [cs.CL].

[5] S. Beliga, A. Metrovi, and S. Martincic-Ipsic. "An Overview of Graph-Based Keyword Extraction Methods and Approaches." In: *Journal of Information and Organizational Sciences* 39 (July 2015), pp. 1–20.

[6] S. R. El-Beltagy and A. Rafea. "KP-Miner: A Keyphrase Extraction System for English and Arabic Documents." In: *Information Systems* 34.1 (Mar. 2009), pp. 132–144. ISSN: 0306-4379. DOI: 10.1016/j.is.2008.05.002.

[7] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, and M. Jaggi. *Simple Unsupervised Keyphrase Extraction using Sentence Embeddings*. 2018. arXiv: 1801.04470 [cs.CL].

[8] S. K. Bharti and K. S. Babu. *Automatic Keyword Extraction for Text Summarization: A Survey*. 2017. arXiv: 1704.03242 [cs.CL].

[9] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. "Enriching Word Vectors with Subword Information." In: *CoRR* abs/1607.04606 (2016). arXiv: 1607.04606. URL: http://arxiv.org/abs/1607.04606.

[10] S. Brin and L. Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine." In: *Computer Networks* 30 (1998), pp. 107–117. URL: http://www-db.stanford.edu/~backrub/google.html.

[11] M. Brysbaert. "How many words do we read per minute? A review and meta-analysis of reading rate." In: *Journal of Memory and Language* 109 (2019), p. 104047. ISSN: 0749-596X. DOI: https://doi.org/10.1016/j.jml.2019.104047. URL: https://www.sciencedirect.com/science/article/pii/S0749596X19300786.

[12] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt. "YAKE! Collection-Independent Automatic Keyword Extractor." In: Feb. 2018. ISBN: 978-3-319-76940-0. DOI: 10.1007/978-3-319-76941-7_80.

[13] B. Chan, S. Schweter, and T. Möller. "German's Next Language Model." In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 6788–6796. DOI: 10.18653/v1/2020.coling-main.598. URL: https://aclanthology.org/2020.coling-main.598.

[14] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. *Unsupervised Cross-lingual Representation Learning at Scale*. 2020. arXiv: 1911.02116 [cs.CL].

[15] S. Danesh, T. Sumner, and J. H. Martin. "SGRank: Combining Statistical and Graphical Methods to Improve the State of the Art in Unsupervised Keyphrase Extraction." In: *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 117–126. DOI: 10.18653/v1/S15-1013.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[17] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998. URL: https://mitpress.mit.edu/9780262561167/.

[18] C. Florescu and C. Caragea. "PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1105–1115. DOI: 10.18653/v1/P17-1102.

[19] Gambolputty. *German nouns*. 2023. URL: https://github.com/gambolputty/german-nouns.

[20] S. D. Gollapalli, X.-L. Li, and P. Yang. "Incorporating Expert Knowledge into Keyphrase Extraction." In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, 2017, pp. 3180–3187.

[21] M. Grootendorst. *KeyBERT: Minimal keyword extraction with BERT.* Version v0.3.0. 2020. DOI: 10.5281/zenodo.4461265.

[22] B. Hamp and H. Feldweg. "GermaNet - a Lexical-Semantic Net for German." In: *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*. 1997. URL: https://aclanthology.org/W97-0802.

[23] Z. Harris. "Distributional structure." In: *Word* 10.23 (1954), pp. 146–162.

[24] K. S. Hasan and V. Ng. "Automatic Keyphrase Extraction: A Survey of the State of the Art." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 1262–1273. DOI: 10.3115/v1/P14-1119.

[25] G. He, J. Fang, H. Cui, C. Wu, and W. Lu. "Keyphrase Extraction Based on Prior Knowledge." In: *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*. JCDL '18. Fort Worth, Texas, USA: Association for Computing Machinery, 2018, pp. 341–342. ISBN: 9781450351782. DOI: 10.1145/3197026.3203869. URL: https://doi.org/10.1145/3197026.3203869.

[26] V. Henrich and E. Hinrichs. "GernEdiT - The GermaNet Editing Tool." In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA), May 2010. URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/264_Paper.pdf.

[27] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory." In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[28] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. "spaCy: Industrial-strength Natural Language Processing in Python." In: (2020). DOI: 10.5281/zenodo.1212303.

[29] A. Hulth. "Enhancing Linguistically Oriented Automatic Keyword Extraction." In: *Proceedings of HLT-NAACL 2004: Short Papers*. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2004, pp. 17–20. DOI: 10.3115/1613984.1613989.

[30] T. Jastrzb and G. Kwiatkowski. "Enriching Keywords Database UsingWordnets – a Case Study." In: *Proceedings of the 10th Global Wordnet Conference*. Wroclaw, Poland: Global Wordnet Association, July 2019, pp. 329–335. URL: https://aclanthology.org/2019.gwc-1.42.

[31] X. Jiang, Y. Hu, and H. Li. "A Ranking Approach to Keyphrase Extraction." In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. Boston, MA, USA: Association for Computing Machinery, 2009, pp. 756–757. ISBN: 9781605584836. DOI: 10.1145/1571941.1572113.

[32] F. Jonathan and O. Karnalim. "Semi-Supervised Keyphrase Extraction on Scientific Article using Fact-based Sentiment." In: *TELKOMNIKA Indonesian Journal of Electrical Engineering* 16 (Aug. 2018), pp. 1771–1778. DOI: 10.12928/TELKOMNIKA.v16i4.5473.

[33] J. S. Justeson and S. M. Katz. "Technical terminology: some linguistic properties and an algorithm for identification in text." In: *Natural Language Engineering* 1.1 (1995), pp. 9–27. DOI: 10.1017/S1351324900000048.

[34] O. Karnalim. "Software Keyphrase Extraction with Domain-Specific Features." In: *2016 International Conference on Advanced Computing and Applications (ACOMP)*. 2016, pp. 43–50. DOI: 10.1109/ACOMP.2016.016.

[35] B. Koloski, S. Pollak, B. krlj, and M. Martinc. *Extending Neural Keyword Extraction with TF-IDF tagset matching*. 2022. arXiv: 2102.00472 [cs.CL].

[36] J. H. Lau and T. Baldwin. "An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation." In: *Proceedings of the 1st Workshop on Representation Learning for NLP*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 78–86. DOI: 10.18653/v1/W16-1609.

[37] Q. V. Le and T. Mikolov. "Distributed Representations of Sentences and Documents." In: *CoRR* abs/1405.4053 (2014). arXiv: 1405.4053. URL: http://arxiv.org/abs/1405.4053.

[38] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL].

[39] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

[40] Z. Liu, P. Li, Y. Zheng, and M. Sun. "Clustering to Find Exemplar Terms for Keyphrase Extraction." In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, Aug. 2009, pp. 257–266. DOI: 10.3115/1699510.1699544.

[41] E. Loper and S. Bird. *NLTK: The Natural Language Toolkit*. 2002. arXiv: cs/0205028 [cs.CL].

[42] D. Mahata, J. Kuriakose, R. R. Shah, and R. Zimmermann. "Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 634–639. DOI: 10.18653/v1/N18-2100.

[43] M. Martinc, B. krlj, and S. Pollak. "TNT-KID: Transformer-based neural tagger for keyword identification." In: *Natural Language Engineering* 28.4 (June 2021), pp. 409–448. DOI: 10.1017/s1351324921000127.

[44] D. McCarthy and R. Navigli. "SemEval-2007 Task 10: English Lexical Substitution Task." In: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 48–53. URL: https://aclanthology.org/S07-1009.

[45] O. Melamud, J. Goldberger, and I. Dagan. "context2vec: Learning Generic Context Embedding with Bidirectional LSTM." In: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 51–61. DOI: 10.18653/v1/K16-1006. URL: https://aclanthology.org/K16-1006.

[46] O. Melamud, O. Levy, and I. Dagan. "A Simple Word Embedding Model for Lexical Substitution." In: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 1–7. DOI: 10.3115/v1/W15-1501. URL: https://aclanthology.org/W15-1501.

[47] R. Meng, T. Wang, X. Yuan, Y. Zhou, and D. He. "General-to-Specific Transfer Labeling for Domain Adaptable Keyphrase Generation." In: *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 1602–1618. DOI: 10.18653/v1/2023.findings-acl.102. URL: https://aclanthology.org/2023.findings-acl.102.

[48] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi. "Deep Keyphrase Generation." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 582–592. DOI: 10.18653/v1/P17-1054. URL: https://aclanthology.org/P17-1054.

[49] G. Michalopoulos, I. McKillop, A. Wong, and H. Chen. "LexSubCon: Integrating Knowledge from Lexical Resources into Contextual Embeddings for Lexical Substitution." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1226–1236. DOI: 10.18653/v1/2022.acl-long.87. URL: https://aclanthology.org/2022.acl-long.87.

[50] R. Mihalcea and P. Tarau. "TextRank: Bringing Order into Text." In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 404–411. URL: https://aclanthology.org/W04-3252.

[51] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. *Distributed Representations of Words and Phrases and their Compositionality*. 2013. arXiv: 1310.4546 [cs.CL].

[52] T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient Estimation of Word Representations in Vector Space." In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2013. DOI: 10.48550/arxiv.1301.3781.

[53] G. A. Miller. "WordNet: A Lexical Database for English." In: *Commun. ACM* 38.11 (Nov. 1995), pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: https://doi.org/10.1145/219717.219748.

[54] *Oxford German Dictionary*. Oxford University Press, 2021.

[55] M. Pagliardini, P. Gupta, and M. Jaggi. "Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/n18-1049.

[56] E. Papagiannopoulou and G. Tsoumakas. "A Review of Keyphrase Extraction." In: *CoRR* abs/1905.05044 (2019). arXiv: 1905.05044. URL: http://arxiv.org/abs/1905.05044.

[57] E. Papagiannopoulou and G. Tsoumakas. *Local Word Vectors Guiding Keyphrase Extraction*. 2018. arXiv: 1710.07503 [cs.CL].

[58] J. Pennington, R. Socher, and C. Manning. "GloVe: Global Vectors for Word Representation." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

[59] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. *Deep contextualized word representations*. 2018. arXiv: 1802.05365 [cs.CL].

[60] A. Radford and K. Narasimhan. "Improving Language Understanding by Generative Pre-Training." In: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[61] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020. arXiv: 1910.10683 [cs.LG].

[62] N. Reimers and I. Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410.

[63] S. Roller and K. Erk. "PIC a Different Word: A Simple Model for Lexical Substitution in Context." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 1121–1126. DOI: 10.18653/v1/N16-1131. URL: https://aclanthology.org/N16-1131.

[64] S. Rose, D. Engel, N. Cramer, and W. Cowley. "Automatic Keyword Extraction from Individual Documents." In: *Text Mining*. John Wiley Sons, Ltd, 2010. Chap. 1, pp. 1–20. ISBN: 9780470689646. DOI: 10.1002/9780470689646.ch1. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470689646.ch1.

[65] C. Saedi, A. Branco, J. António Rodrigues, and J. Silva. "WordNet Embeddings." In: *Proceedings of the Third Workshop on Representation Learning for NLP*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 122–131. DOI: 10.18653/v1/W18-3016. URL: https://aclanthology.org/W18-3016.

[66] T. Schopf, S. Klimek, and F. Matthes. "PatternRank: Leveraging Pretrained Language Models and Part of Speech for Unsupervised Keyphrase Extraction." In: *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KDIR*. INSTICC. SciTePress, 2022, pp. 243–248. ISBN: 978-989-758-614-9. DOI: 10.5220/0011546600003335.

[67] M. Siegel and F. Bond. "OdeNet: Compiling a German Wordnet from other Resources." In: *Proceedings of the 11th Global Wordnet Conference (GWC 2021)*. 2021, pp. 192–198. URL: https://www.aclweb.org/anthology/2021.gwc-1.22.

[68] N. Teneva and W. Cheng. "Salience Rank: Efficient Keyphrase Extraction with Topic Modeling." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 530–535. DOI: 10.18653/v1/P17-2084. URL: https://aclanthology.org/P17-2084.

[69] J. Tissier, C. Gravier, and A. Habrard. "Dict2vec : Learning Word Embeddings using Lexical Dictionaries." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 254–263. DOI: 10.18653/v1/D17-1024. URL: https://aclanthology.org/D17-1024.

[70] T. Tomokiyo and M. Hurst. "A Language Model Approach to Keyphrase Extraction." In: *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*. Sapporo, Japan: Association for Computational Linguistics, July 2003, pp. 33–40. DOI: 10.3115/1119282.1119287.

[71] A. Ushio, F. Liberatore, and J. Camacho-Collados. "Back to the Basics: A Quantitative Analysis of Statistical and Graph-Based Term Weighting Schemes for Keyword Extraction." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.emnlp-main.638.

[72] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin. "Attention is All You Need." In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964.

[73] X. Wan and J. Xiao. "Single Document Keyphrase Extraction Using Neighborhood Knowledge." In: *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*. AAAI'08. Chicago, Illinois: AAAI Press, 2008, pp. 855–860. ISBN: 9781577353683.

[74] R. Wang. "Corpus-independent Generic Keyphrase Extraction Using Word Embedding Vectors." In: 2014. URL: https://api.semanticscholar.org/CorpusID:14116376.

[75] R. Wang, W. Liu, and C. Mcdonald. "Using word embeddings to enhance keyword identification for scientific publications." English. In: *Databases Theory and Applications*. Vol. 9093. 26th Australasian Database Conference ; Conference date: 04-06-2015 Through 07-06-2015. Netherlands: Springer, 2015, pp. 257–268. ISBN: 9783319195476. DOI: 10.1007/978-3-319-19548-3_21.

[76] X. Wang and H. Ning. "TF-IDF Keyword Extraction Method Combining Context and Semantic Classification." In: *Proceedings of the 3rd International Conference on Data Science and Information Technology*. DSIT 2020. Xiamen, China: Association for Computing Machinery, 2020, pp. 123–128. ISBN: 9781450376044. DOI: 10.1145/3414274.3414492.

[77] C. Wartena. *The Hanover Tagger (Version 1.1.0) - Lemmatization, Morphological Analysis and POS Tagging in Python*. en. Tech. rep. 2023, p. 24. DOI: 10.25968/opus-2457. URL: http://nbn-resolving.de/urn:nbn:de:bsz:960-opus4-24570.

[78] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. *KEA: Practical Automatic Keyphrase Extraction*. 1999. DOI: 10.1145/313238.313437. arXiv: cs/9902007 [cs.DL].

[79] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation." In: *CoRR* abs/1609.08144 (2016). arXiv: 1609.08144. URL: http://arxiv.org/abs/1609.08144.

[80] W. Zhou, T. Ge, K. Xu, F. Wei, and M. Zhou. "BERT-based Lexical Substitution." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3368–3373. DOI: 10.18653/v1/P19-1328.