Otto-von-Guericke University Magdeburg

Faculty of Computer Science

Master Thesis

A Study of Keyword and Keyphrase Extraction Techniques for Quantitative Social Science Surveys

Author:

Pawan Joshi

Supervisor:

Dr.-Ing. David Broneske

November 30, 2022

Reviewer:

Prof. Dr. rer. nat. habil. Gunter Saake
Department of Databases and Software Engineering
Otto-von-Guericke-Universität Magdeburg

Joshi, Pawan:
A Study of Keyword and Keyphrase Extraction Techniques for Quantitative Social Science Surveys
Master Thesis, Otto-von-Guericke University Magdeburg, 2022.

Abstract

This thesis work focuses on keyword and keyphrase extraction from questionnaires provided by Deutsche Zentrum für Hochschul- und Wissenschaftsforschung (DZHW). The study was focused on exploring different techniques for automatically extracting keywords and keyphrases. Readers can easily find key information in text by using keywords and keyphrases, which are vital to conveying a particular document's subject. In this work, we used three different techniques, namely, statistical (TF-IDF), graph (TextRank) and embedding-based (KeyBERT) in order to extract the relevant keywords and keyphrases. Moreover, external resources such as thesauri and controlled vocabularies from the social science domain are used in order to make the process efficient for extracting keywords and keyphrases, which improves the index search. Finally, an analysis is made on the keywords and keyphrases extracted for the questionnaires, where we selected a few questionnaires for which the methods worked best. The parameters that we use for analysis are top-k based on recall, top-k based on precision, the total number of keywords/keyphrases within ground truth, and a total number of keywords/keyphrases extracted by the methods. We conclude by arguing that there is no clear winner among the statistical, graph and embedding based. There have been indications that the length of the questionnaires and their uniqueness have a role to play in the methods used. In the end, it is upon the researchers to choose based on their requirements.

Acknowledgments

First and foremost, I want to express my gratitude to my thesis advisor, Dr.-Ing. David Broneske, for his constant supervision, guidance, and encouragement throughout my thesis which allowed me to explore the field of Natural Language Processing. I would like to express my deepest gratitude to Prof. Dr. Gunter Saake for helping me to work with my thesis topic at the DBSE research group. I would like to take the opportunity to also thank Deutsche Zentrum für Hochschul- und Wissenschaftsforschung (DZHW) for providing the necessary data and resources for my thesis.

My sincere thanks also goes to MSc. Marcus Thiel for all the brainstorming session and discussing ideas which has been really invaluable.

I would also like to thank Sankul Rathod for his contribution.

I want to thank MSc. Saijal Shahania for her immense support and motivation. This thesis wouldn't be possible without your help. I'm extremely grateful and blessed to have a mentor and a friend like you.

Last but not least, I would like to thank my tauji for always being supportive which gave me encouragement throughout my master's degree journey.

Contents

List of Figures

List of Tables

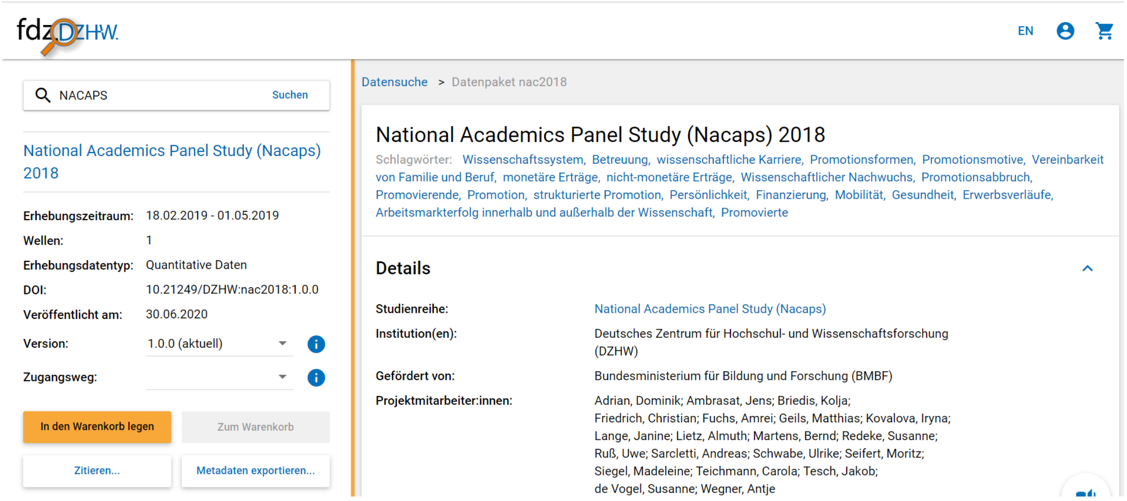
1. Introduction

This chapter briefly introduces the thesis topic, *A Study of Keyword and Keyphrase Extraction Techniques for Quantitative Social Science Surveys*. Firstly, an introduction of the thesis topic will be provided by explaining the nuances of keywords and keyphrase and the extraction process. Furthermore, in section 1.1, we dig into the motivation behind our research, where we emphasize the problems scientists are facing. Moreover, section 1.2, highlights the research questions. Finally, section 1.3 concludes this chapter by providing the structure of subsequent chapters for this thesis.

The Deutsche Zentrum für Hochschul- und Wissenschaftsforschung (DZHW) carries out empirical research with a focus on applications in the domain of science and higher education. They design and conduct qualitative and quantitative surveys. In these surveys, they ask individuals, including international students, about their experiences studying in Germany and Europe and identify effective study practices as well as the economic situation of students. For this, they have a database that hosts these surveys and has a searchable front-end [1]. The platform provided by DZHW comprises a database for university and science research. To acquire the desired metadata from this platform, one can search easily. Furthermore, they give questionnaires to researchers and scientists to inspect and use their data output from conducted surveys. Sharing the data enhances the proliferation of scientific thought and process by fostering transparency, advancing research, and assisting in improved decision-making.

A huge database has been created and made available by different scientific organizations. Figure 1.1 is an example of one such database in place by DZHW. Here we see the metadata related to the National Academics Panel Study (*Nacaps*) survey. We get the summarised information about what can be expected from the *Nacaps* through the Schlagwörter field. Tags like *monetäre Erträge* and *Promotionsformen* hint towards the motive of the study, which in this case revolves around doctoral students. Currently, these tags are human-annotated, which means the researchers spend a

[1] https://metadata.fdz.dzhw.eu/de/start



National Academics Panel Study (Nacaps) 2018

Erhebungszeitraum: 18.02.2019 - 01.05.2019
Wellen: 1
Erhebungsdatentyp: Quantitative Daten
DOI: 10.21249/DZHW:nac2018:1.0.0
Veröffentlicht am: 30.06.2020
Version: 1.0.0 (aktuell)
Zugangsweg:

National Academics Panel Study (Nacaps) 2018

Schlagwörter: Wissenschaftssystem, Betreuung, wissenschaftliche Karriere, Promotionsformen, Promotionsmotive, Vereinbarkeit von Familie und Beruf, monetäre Erträge, nicht-monetäre Erträge, Wissenschaftlicher Nachwuchs, Promotionsabbruch, Promovierende, Promotion, strukturierte Promotion, Persönlichkeit, Finanzierung, Mobilität, Gesundheit, Erwerbsverläufe, Arbeitsmarkterfolg innerhalb und außerhalb der Wissenschaft, Promovierte

Details

Studienreihe: National Academics Panel Study (Nacaps)
Institution(en): Deutsches Zentrum für Hochschul- und Wissenschaftsforschung (DZHW)
Gefördert von: Bundesministerium für Bildung und Forschung (BMBF)
Projektmitarbeiter:innen: Adrian, Dominik; Ambrasat, Jens; Briedis, Kolja; Friedrich, Christian; Fuchs, Amrei; Geils, Matthias; Kovalova, Iryna; Lange, Janine; Lietz, Almuth; Martens, Bernd; Redeke, Susanne; Ruß, Uwe; Sarcletti, Andreas; Schwabe, Ulrike; Seifert, Moritz; Siegel, Madeleine; Teichmann, Carola; Tesch, Jakob; de Vogel, Susanne; Wegner, Antje

Figure 1.1: The figure shows an example of database where the hand annotated keywords and keyphrases for *Nacaps* questionnaire is stored.

lot of time reading the entire questionnaire and coming up with meaningful tags. Given the sheer volume of these questions, the manual annotation process becomes very cumbersome. It not only requires the researchers to go through the relevant information out of voluminous documents but also takes a substantial effort/energy to summarise those ideas into the given tags.

To overcome this problem and to have a quick view of the topic, automatic extraction of keywords and keyphrases plays an essential role. A keyword/keyphrase is any word or group of words that appear more frequently in a text and determines the context of the text Scott and Tribble [2006]. Keywords and keyphrases provide a bird's eye view of the detailed and long documents. Hence, extracting and delivering good keywords and key phrases that can be used as Schlagwörter is imperative.

1.1 Problem Statement

As mentioned in the introduction, the manual annotation of keywords and keyphrases is very tedious. To overcome this problem, researchers need such a notation that they can trust, for we have to build a trustworthy automatic tool. To make a tool like this, we need a bigger dataset, but in our case, The major problem is the dataset. The dataset of DZHW is very limited, and picking questions and answers for every domain is a difficult task in itself. So, we won't have a large dataset, the dataset might be reliable for one situation. However, to ensure that the dataset is valid for all types of questionnaires, the first thing is to build upon the corpus. Another problem is to have the dataset, which the researchers already annotate. As discussed earlier, the annotation process is a very time-consuming task, and we likely have the availability of the questionnaires. Still, there is no tag annotated for that particular questionnaire. Additionally, if we use the non-annotated questionnaire data, we would have to design the problem in an unsupervised setting for automatic annotation. This is a different task compared to where we have the human-annotated data. This scope is vast, so the first thing we have to do is narrow down the scope for it being a supervised or unsupervised setting. In this thesis work, we decided to work with a supervised setting by considering the tags there are manually annotated.

Furthermore, we had also to ensure transparency in our system. Transparency can only be ensured in used approaches if the reason to use the proposed approaches has been strongly backed up. This is important to ensure that the researchers trust the system since the wrong approaches could lead to potentially incorrect tags. This, in-turn, can paint a false picture of a given survey. We choose statistical, graph and embedding-based approaches based on the current state-of-the-art. As we saw that the hand-annotated tags are both in German and English, for this, we needed an approach which is language-independent, and it is known that the statistical approaches are language-independent Singhal and Sharma [2021]. Moreover, several embedding-based approaches have been used to overcome the drawback of supervised and unsupervised keywords and keyphrase extraction. Unlike TF-IDF and TextRank, embedding-based approaches also cover the semantics of the words and phrases. EmbedRank is one such example which is an unsupervised technique that uses sentence embeddings to extract keyphrases Bennani-Smires et al. [2018].

1.2 Research Questions

We designed our system to tackle the problems mentioned above. The scope of this thesis revolves around exploring the statistical, graph and embedding-based approaches to extract the desired tags automatically. We also want to ensure that the automatic system is enriched with the existing domain knowledge via various sources. Hence, for our thesis, we come up with the following research questions:

RQ-1 How to select reasonable questions and answer options from questionnaires which are needed in keywords and keyphrase extraction?

RQ-2 How results differ in keywords and keyphrase extraction using a statistical, graph, and embedding-based approaches?

RQ-3 How can domain knowledge be incorporated for further investigation of extracted keywords and keyphrases?

1.3 Structure of this Thesis

The structure of the following chapters in this thesis is described as follows:

- In Chapter 2, the theoretical foundations of keyword and keyphrase extraction are provide. In addition, a detailed explanation of fundamental approaches and techniques for our problem are given.
- In Chapter 3, we review the existing work on keywords and keyphrase extraction and compare our work to the already existing one. We also saw the work related to the usage of thesauri and controlled vocabularies in ranking keywords and keyphrases.

- In Chapter 4, a brief overview of the datasets is given which is used to perform the experimentation. Furthermore, exploratory data analysis is performed in order to discuss the insights into the datasets. Finally, an explanation is given about the selected thesauri and controlled vocabularies which are used for experimentation.
- In Chapter 5, the methodology of the keyword and keyphrase extraction process used in this thesis work is provided. We also present the workflow of the process used in this thesis, followed by a brief explanation of each module.
- In Chapter 6, an overview is given for the implementation of the concept that is provided in Chapter 5. Furthermore, we discuss the selection of hyper-parameters for each techniques.
- In Chapter 7, first, we discuss the evaluation metrics. Furthermore, we elaborate the evaluation results for all the experimentation.
- In chapter 8, an elaborate discussion on the results of all the approaches are given with the help of qualitative evaluation. We also discuss the application for researchers which emphasises on the importance of metrics used for evaluation.
- Finally, In 9, we summarize the work in this thesis by providing some insights, limitations as well some suggestions for future research.

2. Background

This chapter comprises of the theoretical foundations of keywords and keyphrases extraction. In the beginning, we explain the meaning of keyword and keyphrase extraction, followed by a discussion of the various methods used for the research problem undertaken.

The section is structured as follows::

- In Section 2.1, a general overview of keywords and keyphrases has been given.
- In Section 2.2, a brief description of the relation of keywords and keyphrases with respect to NLP is given.
- In Section 2.3, a discussion is made on the methods used in the implementation section of this thesis.
- In Section 2.4, a description of some popular text preprocessing techniques used in natural language processing tasks is provided.

2.1 Keywords and Keyphrases

In linguistics, a keyword/keyphrase is any word or group of words that appear more frequently in a text and determines the context of the text Scott and Tribble [2006]. When a word or phrase is said to be a *key*, it can have several characteristics. For instance, in a scientific research paper, keywords/keyphrases are important concepts and topics describing the research work. For example, say if someone is new to a city and desires to find a restaurant, he/she may use a search engine to find something like ”nearby restaurant” or something similar. The keywords and keyphrases define the essence of the search terms. The distinction between keywords and keyphrases is that the keywords are single words, whereas keyphrases are composed of multiple words.

In the preceding example, if we consider only “restaurant,” that will be the keyword, but the entire search term, i.e., “nearby restaurant,” is a keyphrase. When discussing keywords/keyphrases, one should also consider the key elements or what makes a particular word key. In linguistics, keyness refers to the quality of single word or group of words being key or important in its context.

Keyness Properties

To define the term keyness, we used the same classification as Firoozeh et. el Firoozeh et al. [2020]. The *keyness* can be distinguished by several properties, such as informational, linguistic, and domain-based.

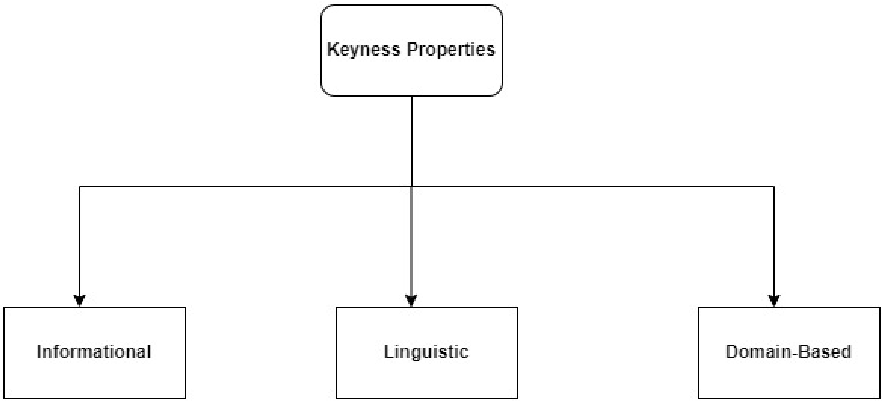


Figure 2.1: The figure gives an illustration of the various properties composing *keyness*

Informational Properties

There are several criteria that were developed Unesco. [1975] by the United Nations Educational Science and Cultural Organization UNESCO in 1975. These criteria are followed by keywords independent of any application and the way keywords are provided, which can be done manually or automatically.

Many principles exist in informational properties, and one of them is *exhaustivity*, which tells that a set of keywords should cover all of the topics inside the documents under consideration. Furthermore, for another principle that is *Specifity*, keywords are regarded as key elements of a document in contrast to other documents that may belong to a different domain. Keywords should be as specific as possible when representing the content of a document.

In *Minimality*, keywords should be distinct from one another, and the keyword set should include unique keywords with different meanings. In *Impartiality*, keywords should be as objective as possible, reflecting the informational content of documents without involving personal belief or opinion. Finally, *Representativity* asserts that Keywords are considered key elements of a document, as opposed to other keywords or keyphrases that reflect minor aspects.

Linguistic Properties:

In corpus linguistics, a keyword is a word that appears in a text more frequently than we would expect. These are small chunks of language that must adhere to certain linguistic rules. The most relevant linguistic properties of a keyword includes *well-formedness* which is significant when the extracted elements are presented to the user. It is important to have well-formed keywords as well as keyphrases. For example, some of the prohibited forms include truncated forms, such as 'gesundhei' instead of 'Gesundheit.' Another property includes *Citationess* which refers to the linguistic form of the keyword that should be retained. Keywords appear in the text in inflectional forms, but only the form without the inflectional should be retained.

Domain-based Properties:

Keyword extraction is useful not only for general language documents like journals, but also for documents in specialized domains. Documents which contain technical and scientific information that use their own vocabulary are examples of such domain-specific terminology.

Some of the properties include *Conformity*, which emphasizes that there is a proper terminology for each domain, which is specific terms for naming the concepts. Various thesaurus and domain-specific terminologies provide a list of terms that are recommended for use as keywords.

Other property which is *Homogeneity.* There are frequently occurring synonymic forms that refer to the same topic and once a keyword is selected, it should be utilized for any document within the same domain dealing with the same topic.

Finally, *Univocity* refers to how clear or unclear a keyword is to a user, keywords in specialized domains are considered less ambiguous than common words, such as cash versus money, because they are keywords or keyphrases whose meaning is consistent within a specific community and context. A few other examples are also provided by Barla et al. [2013] of such disambiguation.

2.2 Application in Natural Language Processing

This subsection will briefly discuss the theoretical foundation of Natural Language Processing(NLP) before making elaboration on several methods for extracting keywords and keyphrases from unstructured data, such as text. Due to NLP's immense variety of applications, the chapter will only concentrate on topics that are crucial to this thesis. Some suggestions for broadening the understanding of the subject are Allen [1995] Indurkhya and Damerau [2010] Chowdhary [2020].

2.2.1 Natural Language Processing

Natural language processing (NLP) Liddy [2001] Nadkarni et al. [2011] is defined as the area of linguistics, computer science, and more specifically the area of artificial intelligence that focuses on the interactions between human language and computers. It deals with enabling machines to comprehend natural language, which can be text or speech, in the same way that humans can.

There are several NLP tasks which are there in order to help the computer to understand the text and speech data it is absorbing. Some of them include speech recognition, named entity recognition, sentiment analysis and part of speech tagging [2]. It can be used for a number of tasks to aid in automating them, including automatic summarization, information retrieval, and subtasks within this field.

[2] https://monkeylearn.com/natural-language-processing/

2.2.2 Information Retrieval

”Information retrieval (IR) is finding material such as documents of an unstructured nature like text that satisfies an information need from within large collections (usually stored on computers)” Schütze et al. [2008]. The IR system helps users to locate the data they need. However, it does not explicitly give the answer to the question; it provides information about the presence and location of documents that may contain the necessary data. The documents that meet the user's needs are called relevant documents. The ideal IR system will only retrieve relevant documents.

Information retrieval Singhal et al. [2001] is typically performed using the various techniques in Natural Language Processing. One such technique is “Indexing” Kaur and Gupta [2016], which helps in obtaining the relevant information from a collection of documents. The indexing process involves extraction of essential terms from documents. These terms are sometimes referred to as keywords or keyphrases Ohsawa et al. [1998].

2.2.3 Text Analysis

Text analysis Bernard and Ryan [1998], commonly referred to as text mining Tan et al. [1999] Hotho et al. [2005], is a machine learning Mitchell and Mitchell [1997] Wang et al. [2016] technique that helps to extract information or insights from unstructured data. These insights, in turn, allows businesses to make informed data-driven decisions. Text analysis can be used to quickly and accurately evaluate various text-based sources, including emails and posts on social media. One illustration would be to analyze customer reviews on an e-commerce site and group them according to sentiment and topic. There are many different text analysis techniques available for various purposes, including text classification, text extraction, word frequency, clustering. This thesis will solely concentrate on the text extraction portion of those techniques.

- **Text Extraction:**

  Text extraction is a popular text analysis approach which uses NLP techniques [3]. Natural language processing techniques are mostly utilized to automatically scan text and extract words and phrases from unstructured text sources like surveys and news articles. It can pull out keywords and business data like names and job descriptions, product reviews, and more. Text extraction and text classification are tasks that are frequently carried out together. Feature extraction, keyword extraction, and named entity recognition are a few of the typical text extraction tasks. [4]

- **Keyword Extraction:**

  The task of keyword extraction involves automatically finding a group of words or phrases that best describe the content of a text Berry and Kogan [2010] Litvak et al. [2011] Zhou et al. [2013] . It gives readers a significant way to represent the subject of a certain document and can make it easier for them to find key

[3] https://monkeylearn.com/text-analysis/

[4] https://monkeylearn.com/blog/text-classification-vs-text-extraction

information in the text. This is frequently used for information retrieval, text analysis, and summarization tasks.

More formally, given a document D containing N words ($D = \{w1, w2, ..., wn\}$), the keyword extraction process aims at finding the smallest subset $k \subseteq D$ of words that still preserves the meaning of the text Eugenia [2018].

This can be achieved through either assignment or extraction. In the keywords assignment, the keywords are chosen from a controlled vocabulary of terms. In keyword extraction, keywords are chosen from the terms explicitly mentioned in the original text and are not restricted to a set of keywords from a predefined vocabulary.

2.3 Keyword Extraction Methods:

Keyword and keyphrase extraction are primarily divided into three categories: supervised, unsupervised and semi-supervised techniques.

In supervised learning, the extraction of keywords and keyphrases is frequently approached as a binary classification task, where candidate phrases are categorized as either positive (keywords and keyphrases) or negative (non-keywords and keyphrases), which in turn indicates whether a keyword or keyphrase is good or bad.

On the other hand, unsupervised keyword/keyphrase extraction methods are presented as a ranking problem.

Lastly, a semi-supervised keyword extraction algorithm requires a limited quantity of training data, which is used to create a keyword extraction model and extract keywords from the new text Qian et al. [2021].

The main focus of this thesis is on unsupervised keyphrase extraction methods.

2.3.1 Statistical-Based Approaches

Statistical techniques come under unsupervised methods, which are fundamental techniques,independent of domain and language which do not require training data. One of the state-of-the-art statistical techniques is term frequency -inverse document frequency (TF-IDF) Salton and Buckley [1988] which has been incorporated in this thesis. In statistical approaches Campos et al. [2020] Luthra et al. [2017] Wartena et al. [2010], keywords/keyphrases can be identified using the word statistics, examples of such statistics are n-gram, TF-IDF, and word co-occurrences. The most significant benefit of statistical techniques is that they are independent of the language in which they are applied, and as a result, the similar method can be applied to numerous. languages

1. **Term Frequency - Inverse Document Frequency (TF -IDF) :**

   TF-IDF is a combination of two distinct terms, i.e. Term Frequency and Inverse Document Frequency which is a statistical measure that determines a word's significance to a document within a group of documents or corpus. It is frequently employed as a weighting scheme in text analysis and information retrieval.

Formally, the measure is based on the frequency of occurrence of the word t in document d as well as in the entire collection of documents D. The calculation is shown in Equation 1.1:

$$tfidf(t, d, D) = tf(t, d) * idf(d, D) \tag{2.1}$$

In the above equation, tf stands for term frequency which measures how frequently a term appears in a document. It is calculated as a score between a term t and a document $d \in D$. There are various ways to calculate this, and Equations 1.2 and 1.3 show some of the frequently used functions.

$$tf(t, d) = \begin{cases} 1 & \text{if } t \in d \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

$$tf(t, d) = \sum_{w \in n} \begin{cases} 1 & \text{if w = t} \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

The idf stands for inverse document frequency, which measures how common or rare a term is in the entire data set or corpus. The closer a word's frequency is to 0, the more common it will be for a document. This metric is obtained by taking the total number of documents, dividing it by the total number of documents containing a word, and then computing the logarithm.

Below equations show two variants of frequently used functions Salton and Buckley [1988]:

$$idf(d, D) = \log(\frac{|D|}{|d|}) \tag{2.4}$$

$$idf(d, D) = \log(\frac{|D| - |d|}{|d|}) \tag{2.5}$$

The word's TF-IDF score within a document is obtained by multiplying the two metrics, term frequency (TF) and inverse document frequency (IDF). The more relevant a word is in a given document, the higher the score.

Finally, a word is considered significant if it frequently appears in a document or small set of documents and is rarely used in other documents. Only the term frequency will be considered if this metric is applied to a single document.

2. **N-gram statistic:**

The n-gram statistic is also referred to as word collocations and co-occurrences, which is another simple statistical approach that assists in determining the semantic structure of a text by allowing different words to be counted as a single unit. These are continuous sequence of n items, such as words or characters from a particular text sequence.

Words that regularly occur together are called collocations. The most frequent collocations are bi-grams which are an n-gram of size two, for example:

'wissenschaftliche Karriere', so here we can see that two terms are appeared adjacently and they form a bi-gram. Moreover, an n-gram of size three is called trigram, which is a group of three words like 'Studierende mit Kind' and finally, a unigram is an n-gram of size one. Contrarily, co-occurrences relates to words that frequently appear together in a corpus. Although they need not be close to one another, however, they might have semantic proximity.

2.3.2 Graph-Based Approaches

Graph-Based Approaches techniques are increasingly being employed as unsupervised approaches for extracting keywords and keyphrases Beliga et al. [2015] Bougouin et al. [2013] Anjali et al. [2019]. The main idea of these approaches is to create a graph from an input document in which words are represented as vertices and relation between the words is represented by edges.

Graph-based ranking algorithms basically determine the significance of a vertex inside a graph. The significance is deduced based on global data recursively gathered from the entire graph. As well how the vertices and edges propagate information to what is important and what is not is called voting or recommendation.

TextRank

TextRank is one of the graph-based approaches utilized in this thesis work for extracting keywords and keyphrases. TextRank was inspired by the widespread use of the PageRank algorithms Mihalcea and Tarau [2004].

PageRank Algorithm

Google uses the PageRank algorithm to rank websites in its search engine results which helps in measuring the importance of website pages. It is not the only algorithm Google uses to rank search engine results, but it was the first one the business ever deployed.

The web can be visualized as a directed graph, with nodes denoting web pages and edges denoting links among them.

More formally, let $G = (V, E)$ be a directed graph with the set of vertices V and set of edges E, where E is a subset of VXV. For a given vertex V_i, let $In(V_i)$ be the set of vertices that point to its predecessors, and let $Out(V_i)$ be the set of vertices that V_i points to its successors. The score of vertex V_i is defined with the equation 2.6 Mihalcea and Tarau [2004].

$$S(V_i) = (1-d) + d * \sum_{j \in In(v_i)} \frac{1}{|Out(v_j)|} S(V_j) \tag{2.6}$$

Here, $S(V_i)$ is the weight of webpage i, d is a damping factor and the usual value of damping factor lies between 0 and 1 and its role in the formula is to give a minimum score to newly added web pages.

TextRank uses a graph representation of text as the basis for extracting keywords and keyphrases. Each noun and adjective in the text is added as a vertex to the graph, and if two terms co-occur within a context window of N words, an edge is created between them. These edges can be be weighted or unweighted, directed or undirected. Once the graph is constructed each vertex is initially assigned a value of 1 and the PageRank algorithm is then used until a threshold is reached. Then, based on the number of vertices in the graph, the top T words are chosen for additional processing.

The basic difference between PageRank and TextRank is that PageRank is for webpage ranking, and TextRank is for text ranking. The webpage in PageRank is equivalent to the text in TextRank, so the basic idea is the same. A key feature of TextRank is that it is extremely adaptable to different areas, genres, or languages because it does not require in-depth linguistic expertise or annotated corpora particular to a certain subject or language.

2.3.3 Embedding-Based Approaches

Word embeddings Mikolov et al. [2013] is a method of representing texts and documents. They are the learned representations of text in an n-dimensional space where words with the same meaning are represented similarly. Word embeddings can capture the context of a word within a document, semantic and syntactic similarity and also relation with other words.

Several works have been done in the field of keywords and keyphrase extraction which used word and sentence embeddings. For example, Sent2Vec Moghadasi and Zhuang [2020] creates sentence embeddings using word n-gram characteristics. Instead of generic word vectors, it creates word and n-gram vectors that are trained to be additively concatenated into a sentence vector. EmbedRank Bennani-Smires et al. [2018] represents the candidate phrases and the document in the same high-dimensional vector space using sentence embeddings (Doc2Vec or Sent2Vec). More elaboration on the previous work related to embeddings-based approaches is given in chapter 3.

KeyBERT is one of the method that has been employed in this thesis which is a keyword extraction method that uses BERT(Bidirectional Encoder Representations from Transformers) embeddings to extract keywords and keyphrases.

BERT

In order to make models understand the contextual meaning of words, there has been significant research from ELMO (Embeddings from Language Models) to GPT (Generative Pre-trained Transformer), and the development of BERT has considerably aided NLP researchers in their quest for a solution. With BERT, it is easier to look for new words and translate between languages to a more appropriate one.

BERT stands for *Bidirectional Encoder Representations from Transformers*, which is a state-of-the-art language model Devlin et al. [2018]. BERT is created on transformer architecture which is a family of neural network architectures. Transformer architecture as a whole is based on self-attention. Self-attention involves developing the ability to weigh the significance of each item or word in relation to the other words in the input sequence Vaswani et al. [2017]. A transformer architecture includes a encoder-decoder network that uses self-attention on the encoder side and attention on the decoder side, and BERT is basically an encoder stack of transformer architecture. BERT is basically used to change the input text to numerical representation i.e. word embeddings and it is also used as an all purpose pre-trained model that is fine-tuned for specific tasks.

BERT has been pre- trained on a sizable corpus of unlabeled text, such as the entirety of Wikipedia (which has 2,500 million words) and the Book Corpus (800 million words). [5]

KeyBert

KeyBert is a minimal and easy to use keyword extraction technique which makes use of BERT embeddings and cosine similarity to create keywords and keyphrases for a document and these keywords and keyphrases are expected to be very similar to the contents of the document. [6]

KeyBert starts by embedding a document into a vector by turning a chunk of text into a fixed size vector that represents the semantics of this document, and then on the second step, it extracts keywords from this document using simple techniques such as CountVectorizer or tfidfVectorizer(cf. Chapter 6). After extracting these keywords, keybert embed each one of them using the same model used previously to embed the document and at the end we will have a list of keyword embeddings. Once keyword as well as document embeddings are obtained then Keybert will compute a similarity measure between all of these vectors and the document embeddings and this will result in an n size vector, where each value is the similarity measure between the keyword and the document. Finally, it will sort these results in decreasing order to get the top relevant keywords.

In the end when the set of keywords and keyphrases are retrieved, it might be the case that there is a little bit of redundancy in the results. In order to diversify the results there exists two techniques:

- **Max Sum Distance:**

  Max sum distance selects all the candidates and then computes a similarity matrix between all of them. The matrix is a pairwise similarity matrix and then at the end it selects the terms that are least similar to each other and yet similar to the document.

- **Maximal Marginal Relevance:**

  Maximal marginal relevance implements the same technique as above which also adds a diversity threshold that controls the diversity between the terms. Providing a high diversity value will give a very diverse set of keywords/keyphrases and otherwise if we lower the diversity we will have output with the same keywords/keyphrases that can be repeated.

[5] https://www.projectpro.io/article/bert-nlp-model-explained/558

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

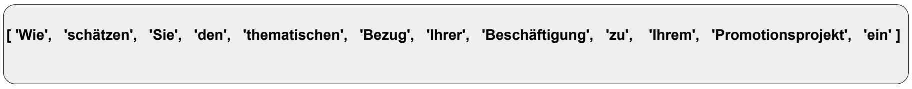
2.4 Text Pre-processing Techniques

Very often, the text data that is extracted form any domain is unstructured and noisy, and text pre-processing techniques are needed to clean the data which in-turn easier to analyze for a given task. Hence, it is required to pre-process the texts in order to convert them into formats that the computers can understand. Some of the important text pre-processing techniques are presented here.

2.4.1 Tokenization:

Tokenization is the process of breaking down the text into smaller units like sentences or words and these smaller units are called tokens. The fundamental task of tokenization is to comprehend the context of the sentence or word by analyzing the token.

For instance, if there is a sentence *"Wie schätzen Sie den thematischen Bezug Ihrer Beschäftigung zu Ihrem Promotionsprojekt ein"*, and if we pass this sentence through a tokenizer it will output result exactly as shown below:



```
[ 'Wie',  'schätzen',  'Sie',  'den',  'thematischen',  'Bezug',  'Ihrer',  'Beschäftigung',  'zu',  'Ihrem',  'Promotionsprojekt',  'ein' ]
```

Figure 2.2: The figure shows the process of tokenization based on the example given below.

2.4.2 Stemming and Lemmatization:

As the name implies, stemming is the process of breaking down a word into its root form or stem word. In stemming, the algorithms does not take into account any knowledge of the part of speech or the context of the word in the language it belongs to. In order to get around with this problem the notion of lemmatization is used.

Lemmatization, a more comprehensive variation of stemming. Lemmatization also considers the context of the word and transform each word to its "lemma", or equivalent root form. While converting a word into its root-form, lemmatization always returns the word's dictionary definition. Since lemmatization performs a morphological examination on the words, it is favored to stemming

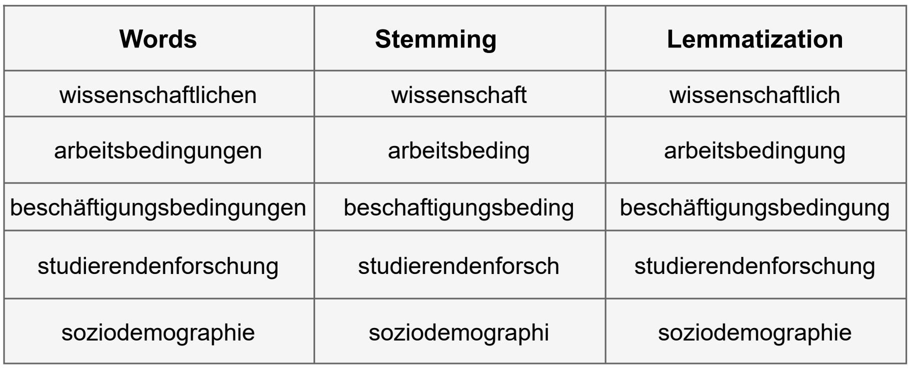
Some examples of stemming and lemmatization are mentioned below:

2.4.3 Stop Word Removal:

The most frequently used terms or words in a text that offer no useful information are called stop-words. These are typical terms that are a part of every language's grammar. Removing stop-words can help us to distinguish a keyword set better. However in other applications such as sentiment analysis removing all the stop-words can sometimes be detrimental.

for example: in German langauge, terms such as kannst, können, mein, meine, die, der are called stopwords.

In this thesis stop-words have been removed while performing the keywords extraction tasks however for keyphrase extraction we have filtered out stop-words based on a regex pattern which is discussed in chapter 5.



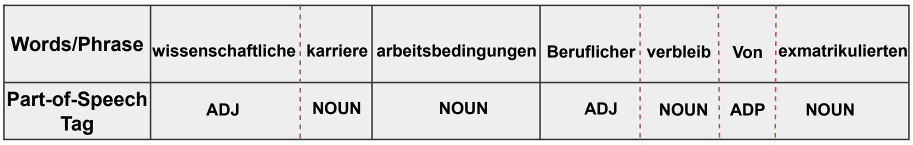
Words	Stemming	Lemmatization
wissenschaftlichen	wissenschaft	wissenschaftlich
arbeitsbedingungen	arbeitsbeding	arbeitsbedingung
beschäftigungsbedingungen	beschaftigungsbeding	beschäftigungsbedingung
studierendenforschung	studierendenforsch	studierendenforschung
soziodemographie	soziodemographi	soziodemographie

Figure 2.3: The figure depicts a differentiation between Stemming and Lemmatization.

2.4.4 Part of speech tagging:

Part-of-speech (POS) tagging, a common NLP technique, refers to classifying words in a text corpus in accordance with a specific part of speech, depending on the context and the definition of a word. Some example of parts of speech include nouns, verb, adverbs, adjectives, pronouns, conjunction and their sub-categories. It is far more difficult to identify part of speech tags compared to just simply map words to their POS tags. Depending on the context, it is entirely possible for the same word to be assigned a different part of speech in various sentences, this is why it is not possible to have a universal mapping for part of speech tags.

An illustration of POS tagging would be :



Words/Phrase	wissenschaftliche	karriere	arbeitsbedingungen	Beruflicher	verbleib	Von	exmatrikulierten
Part-of-Speech Tag	ADJ	NOUN	NOUN	ADJ	NOUN	ADP	NOUN

Figure 2.4: The figure shows an illustration of POS tagging. Here the first example is a keyphrase for which each word is tagged with its respective pos. Similarly, the second example is a keyword tagged with its pos and, finally, a keyphrase where each word is tagged with its pos tag.

3. Related Work

In this chapter, we review the existing work on keywords and keyphrase extraction and compare our work to the already existing one.

Keyword and keyphrase extraction methods are usually divided into traditional and embedding-based approaches. Therefore, we have looked at what has already been explored in the past concerning the main topic and summarized them.

3.1 Traditional work on keywords and keyphrase extraction

In this section, we differentiate already explored work between the traditional methods, including statistical-based and graph-based techniques. Furthermore, more recent ones that also use contextual and word embeddings. Using statistics is one of the simplest ways to determine the main keywords and keyphrases within a document. Statistical methods do not require prior domain expertise or training data. Another benefit of using statistical approaches is that these are language-independent. The statistics of words from the document serve as a foundation for these approaches in keyword and keyphrase extraction Singhal and Sharma [2021].

As shown in the work of Luhn [1957], how statistical properties are used in the process of extracting keywords from a given text. The method is based on zipf's distribution. This study involves counting the instances of each unique word in a text and constructing a list of all those terms in decreasing order of frequency and within this list each term is identified by either its position or Zipf's rank Herrera and Pury [2008]. A first-hand observation of Zipf's distribution states that given a list of most common words, the most common word will occur twice as often as the second most common, which will appear twice as often as the third most common, and so on Adamic and Huberman [2002]. However, the primitive method that Luhn suggested is to take the remaining cases as keywords while removing the terms at the two ends of Zipf's list. Our thesis also focuses on extracting keywords and keyphrases from the questionnaire using statistical based approach, i.e., TF-IDF.

Hulth [2003] proposed automatic keyword extraction from abstracts as a supervised approach. To do so, they not only made use of statistical features but also incorporated linguistic understanding and called it syntactic features. In this study, experimentation using the n-gram, noun phrase chunks, and phrases matching any of a set of POS tag sequences are provided. The four features utilized are the word frequency, collection frequency, relative position of the initial occurrence, and the POS tags attached to the term. Working with NP-chunks when manually assigned keywords were examined, most words were nouns or noun phrases with adjectives. Finally, compared to extracting n-grams, NP-chunks provide better precision, and POS pattern-based word extraction provided higher recall. A POS based filtering was also incorporated in our thesis for extracting relevant keywords. In order to extract keyphrases a sequential regex pattern was designed which is explained in Section 5.4

A technique for keyword extraction from a single document was put forth by Qin [2012]. The primary idea of this paper is to extract keywords using statistical information. In order to select the candidate words, they explored 3 features: word frequency, POS of words and location features of a keyword and to rank and select the candidate keywords, they use a multi-level filter such as a document word frequency filter, features of word POS and word length and finally, a location constraint logic is used. After applying this multi-level filter, the high-frequency terms are rearranged, and then the top words are chosen as keywords. Another widely used statistical approach for keywords and keyphrase extraction is tf_idf, and based on the findings of Hasan and Ng [2010] work, it was recognized as a strong baseline method with excellent performance across many datasets. Terms which have the higher TF-IDF score are considered to be important and are utilized for indexing. One significant drawback of the TF-IDF approach is that it depends on the corpus that we are taking into account which in-turn limits its applicability to dynamic collections Duari and Bhatnagar [2019] and also it does not consider the semantic similarities between words. In this thesis work we have also considered TF-IDF as a baseline to extract keywords and keyphrases from the questionnaires.

Campos et al. [2020] designed a system based on an unsupervised approach and extracted features from the text of several domains, lengths, and different languages. Basically, they use both the context and information that includes statistical information like word position and frequency in the document to extract the keywords. This work applies to tasks with significantly less training data or required external domain knowledge.

In a graph based keywords/keyphrase extraction method, given a document, first it is converted as a graph where words are represented by vertices which are nodes in the graph and the relationship between these words are represented by edges or links. Beliga et al. [2015] have mentioned several principles for making this relationship between words based on edges, such as co-occurrence relations, syntax relations, and semantic relations.

Ohsawa et al. [1998] presented KeyGraph, which is a domain-independent algorithm. They segment the graph into clusters which represent the co-occurrence between words in a document. Here they indexed the academic documents and extracts keywords which represent an overview of a document's main point. This indexing is based on data from the document, such as term frequency and location, rather than the corpus.

Mihalcea and Tarau [2004] proposed TextRank, which is a graph-based ranking model. They analyzed two unsupervised methods for keywords and keyphrase extraction. This algorithm is based on PageRank, which is elaborated in Chapter 2. TextRank functions well since it considers information recursively acquired from the entire text rather than just the local context of a text. In order to restrict the vertices that are added to the graph, they have used syntactic filters, which considers part of speech tags of some lexical units. Several experiments were carried out with a number of syntactic filters, such as nouns and only verbs. Finally, they found that nouns and adjectives alone produced the best results for keywords and keyphrases. In addition to this work, Mihalcea [2004] proposed a method which uses TextRank to provide a novel unsupervised technique for automatic sentence extraction in the context of a text summarizing task. A direct motivation of using a graph-based method for keywords and keyphrase extraction in our thesis comes from Mihalcea and Tarau [2004].

Another unsupervised graph-based keyphrase extraction algorithm that was introduced by Bougouin et al. [2013] is TopicRank . It is based on a topic representation of the document. Firstly, candidate keyphrases are grouped into topics and then used as nodes in a complete graph. In their work, noun phrases that indicate a document's major topics are extracted. Secondly, each topic is given a significance score using a graph-based ranking mechanism and in this case TextRank is used. Finally, keyphrases are extracted by selecting the top-ranked topics from the candidates.

Finally, to conclude the work related to graph and statistical based methods for keywords and keyphrase extraction, Papagiannopoulou and Tsoumakas [2020] presented an illustrative summary on unsupervised keyphrase extraction. In this study, he shows that the most popular methods are graph based. However, statistics based methods are also used widely for different research use cases.

In the context of keywords and keyphrase extraction, various kinds of word as well sentence embeddings methods are utilized. In order to cope with the drawback of supervised and unsupervised keyphrase extraction, Bennani-Smires et al. [2018] proposes EmbedRank, an unsupervised technique that uses sentence embeddings to extract keyphrases from a single document. Being a corpus independent method, EmbedRank can be implemented on top of any document embeddings only if these embeddings can encode documents of arbitrary length. They also put forth EmbedRank++ to diversify the results, using embedding-based maximal marginal relevance(MMR), which was also incorporated in KeyBERT for our experimentation.

Wang et al. [2014] presented a weighting scheme for unsupervised graph-based keyphrase extraction. In this work, they incorporated word embeddings to represent background knowledge for determining informativeness as well as local mutual information for determining *phraseness*. Additionally, they also represent the documents as a weighted undirected graph. Using the values provided by the word embeddings and statistical data, the *informativeness* and *phraseness* Tomokiyo and Hurst [2003] scores of words are computed. The calculated scores are assigned to each edge of the constructed graph as weights. In order to rank the words and extract phrases, they used a weighted Page Rank algorithm Xing and Ghorbani [2004].

Mahata et al. [2018] in their work, proposed an unsupervised method which makes use of phrase embeddings to re-rank keyphrases which are extracted from scientific papers. In order to rank they used the PageRank algorithm and called it a theme-weighted PageRank Langville and Meyer [2004]. Thematic weights are assigned to each candidate phrase once they have been retrieved, indicating how similar they are to the article's central theme or thematic representation, which was also created using the same embeddings. In our work, to extract the relevant keyphrases, we have also created phrase embeddings for the phrases extracted from questionnaires and re-ranked them.

Moreover, in the work of Wang et al. [2015], word embeddings are used for keyword extraction and generation. They propose an unsupervised graph ranking method employing word embedding vectors for extracting keywords within the document. Keyword generation is built on top of the extraction approach, which also uses word embeddings. First, they calculate the average of the word embeddings of the relevant words identified by the extraction model. Afterwards, they used the approximate nearest neighbour algorithm Indyk and Motwani [1998] to find the closest words to the average word vector.

Schopf et al. [2022] presented PatternRank, which is an unsupervised approach for keyphrase extraction. To extract keyphrases, they leverage pre-trained language models(PLMs) as well as part of speech tagging. As PatternRank is an unsupervised approach, it does not need manually annotated data and can be used in several domains. They develop a POS pattern that extracts potential keyphrases, and these potential keyphrases are then given to a pre-trained language model to rank them according to how closely they resemble the input text content. Embeddings for the document as well as candidates are also computed, and afterwards, the cosine similarity between the document and candidate embeddings is calculated. Finally, the outputted keyphrases are ranked in descending order based on the similarity score. KeyBERT is implemented in our work and also includes computing the document embeddings and calculating cosine similarity to choose top n keywords and keyphrases.

3.2 Usage of thesauri and controlled vocabulary

There are several resources, such as thesauri, controlled vocabulary and dictionary, which have been explored to improve and measure the quality of keywords and keyphrases extracted by a particular method. These external resources are domain-specific thesauri and provide supplementary information regarding the words and phrases.

In the work on Gazendam et al. [2010], an extraction and re- ranking approach was presented for keywords which make use of restricted vocabulary or thesaurus. The vocabulary was used basically for two objectives one for the annotation of chunks of the text with terms within the thesaurus and another for the ranking of those terms. In order to rank words, a weighting mechanism($tf.rr$) was created, which uses the frequency of terms within the document as well as the relationship between thesaurus terms and documents. Finally, their findings demonstrated that the new weighting system performs on par with the traditional $tf.idf$. This shows that using thesaurus relations instead of a reference corpus may be an option.

Medelyan and Witten [2005] proposed a method known as ”index term extraction,” which is a middle ground between term assignment and keyphrase extraction. They use a domain-specific thesaurus as a controlled vocabulary and a knowledge basis for semantic matching. For the extraction process, they used KEA++, which is an improvement of the original assignment algorithm that is KEA Witten et al. [1999] which combines the positive elements of keyphrase extraction and term assignment into a single scheme. They compared 200 manually indexed documents to the automatically allocated words, and the results showed that KEA++ outperformed the state-of-the-art keyphrases extraction algorithm or KEA. In our thesis, thesauri from social science domain such as Thesoz is used for matching keywords/keyphrases extracted by keyBERT. Afterwards, a re-raking scheme is used to rank those matched keywords/keyphrases. A detailed explanation is in Section 5.5.

Based on statistical information, co-occurrences of terms are a standard way of extracting keywords and keyphrases. However, incorporating external sources, such as a domain-specific thesaurus, might improve the quality of the extraction process. Hulth et al. [2001] presented an experiment where they first extracted keywords from documents and then compared those keywords with the manually annotated one as well as with a thesaurus. This process helps them to choose a good set of keywords.

In the work of He et al. [2018], prior knowledge is taken into consideration which involves controlled vocabulary. This method improves the extraction of keyphrases as well as ranking from methods like graph-based or statistical-based. The incorporated prior probability with TF-IDF and TextRank for extracting and ranking the keyphrases. The prior probability of keyphrases from controlled vocabulary is taken into consideration as base knowledge. Finally, they concluded that, when paired with the controlled vocabulary of keyphrases that is a component of past knowledge, unsupervised learning techniques like TF-IDF and TextRank are seen to perform significantly better than using TF-IDF and TextRank separately. Another work which focuses on improving the effectiveness of information retrieval is presented by Kamps [2004]. They proposed a technique based on feedback which helps re-ranking initially obtained documents using the controlled vocabulary terms assigned to them. Instead of relying on a specific thesaurus or dictionary, they computed meaning of terms inside the controlled vocabulary based on how often they appear in the corpus. They evaluated their experiments on German and French controlled vocabularies. Finally, this re-ranking method significantly increased the retrieval efficiency in domain-specific collections. This also motivated us to incorporate controlled vocabularies like CESSDA Controlled Vocabulary which is explained in Chapter 4 and Section 5.5.

4. Datasets

This chapter summarises the dataset utilized for experimentation in the thesis, followed by a quick summary of the exploratory data analysis. Finally, an overview of thesauri and controlled vocabularies is used in further experiments.

4.1 An outline of dataset

As mentioned in the introduction chapter, the DZHW conducts qualitative and quantitative surveys related to higher education and science studies which are in the form of questionnaires.

There are mainly three types of bigger surveys that we have access to:

1. **Social Survey (Sozialerhebung)**

   Several surveys have been conducted as part of the social survey about the social and economic situation of students in Germany since 1951. In addition, a cross-section of students is surveyed every three to four years about their experiences getting into college, the structural aspects of their studies, their social and economic circumstances, including their income and cost of living, and their employment.

2. **DZHW Graduate Survey Series (DZHW-Absolventenpanel)**

   One of the parts of DZHW Graduate Survey series is Graduate Panel 2009, which uses standardized surveys to gather data on studies, career entry, professional advancement and further qualifications of higher education graduates.

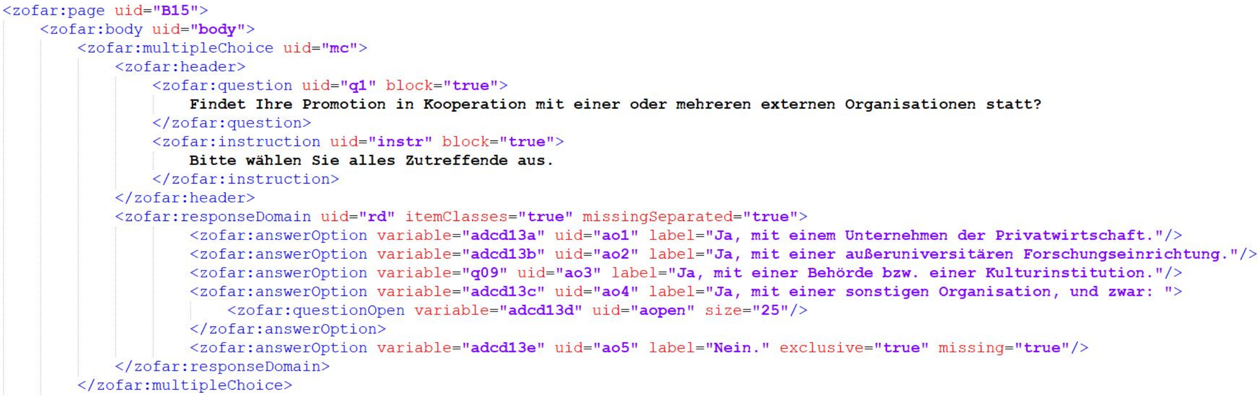
3. **DZHW Survey Series of School Leavers (DZHW-Studienberechtigtenpanel)**

   In the DZHW-Panel Study of School Leavers survey series, various standardized surveys are used to gather data on the post-school careers of school leavers with a (school) higher education admission certificate. The DZHW-Panel Study of School Leavers 2012 is a part of this survey series.

Typically, there are different waves or versions of these surveys questionnaire. For example: for our experimentation we are using a questionnaire named as *Promopanel*. We are provided different versions or waves of these questionniares namely, *Promopanel_W2*, *Promopanel_W3*, *Promopanel_W4* and *Promopanel_W5*. The difference between them is that he questions asked in one wave will differ from the questions asked in the another.

4.1.1 Text extraction from questionnaires

The surveys were available in XML format, and as our methodologies deal with plain text, parsing for these surveys was necessary. As mentioned in the introduction chapter Chapter 1 our methodology deals with extracting keywords and keyphrases from only question and answer options of the questionnaires. In order to do so, while parsing the XML files, we only retrieve question and answer options. The survey conducted by DZHW has several types of questions and answers. For instance, Likert scale, single choice, multiple choice, open ended, and matrix question. In figure 4.1, we show an example of multiple choice question.



```
<zofar:page uid="B15">
    <zofar:body uid="body">
        <zofar:multipleChoice uid="mc">
            <zofar:header>
                <zofar:question uid="q1" block="true">
                    Findet Ihre Promotion in Kooperation mit einer oder mehreren externen Organisationen statt?
                </zofar:question>
                <zofar:instruction uid="instr" block="true">
                    Bitte wählen Sie alles Zutreffende aus.
                </zofar:instruction>
            </zofar:header>
            <zofar:responseDomain uid="rd" itemClasses="true" missingSeparated="true">
                    <zofar:answerOption variable="adcd13a" uid="ao1" label="Ja, mit einem Unternehmen der Privatwirtschaft."/>
                    <zofar:answerOption variable="adcd13b" uid="ao2" label="Ja, mit einer außeruniversitären Forschungseinrichtung."/>
                    <zofar:answerOption variable="q09" uid="ao3" label="Ja, mit einer Behörde bzw. einer Kulturinstitution."/>
                    <zofar:answerOption variable="adcd13c" uid="ao4" label="Ja, mit einer sonstigen Organisation, und zwar: ">
                        <zofar:questionOpen variable="adcd13d" uid="aopen" size="25"/>
                    </zofar:answerOption>
                    <zofar:answerOption variable="adcd13e" uid="ao5" label="Nein." exclusive="true" missing="true"/>
            </zofar:responseDomain>
        </zofar:multipleChoice>
```

Figure 4.1: The figure shows the XML design for questionnaire which depicts an example of a multiple choice question.

In Figure 4.1, the tag `<zofar:page uid="B15">` stands for an id of question, whereas `<zofar:multipleChoice uid="mc">` depicts the type of question which in this case is a multiple choice question. `<zofar:question uid="q1" block="true">` represents the question tag and text inside this tag is relevant in our use case. `<zofar:answerOption` includes a few set of attributes including *variable*, *uid* which varies according to the number of answer options the respective question has and finally, *label* which comprises of answer text that we basically retrieve for our use case. Table 4.1 shows the results after parsing the XML tag for this particular multiple choice question

4.1.2 Types of survey questions

As mentioned in Section 4.1.1, there are several questions and answer options within a questionnaire which we are using for the purpose of extracting keywords and keyphrases. Table 4.2 lists several question types that exist within a given questionnaire.

Question	Answer options
Findet Ihre Promotion in Kooperation miteiner oder mehreren externen Organisationen statt?	Ja, mit einem Unternehmen der Privatwirtschaft. Ja, mit einem Unternehmen der Privatwirtschaft. Ja, mit einer Behörde bzw. einer Kulturinstitution. Ja, mit einer sonstigen Organisation, und zwar

Table 4.1: The table shows the output that is achieved after successfully parsing the example. The first column shows the question text. Second column depicts the different answer option text.

Question Type	XML Tags
Single choice questions	<zofar:questionSingleChoice>
Multiple choice questions	<zofar:multipleChoice>
Open ended questions	<zofar:questionOpen>
Matrix Questions	</zofar:matrixQuestionOpen> <zofar:matrixQuestionSingleChoice> <zofar:matrixMultipleChoice matrixQuestionMixed

Table 4.2: The table shows the type of questions with their respective XML tags that exist within a questionnaire.

For each question and answer tag in the XML file we have to custom design different functions in order to extract the text in the required format. This is because for each tag there was different type of formatting within the XML design. In figure 4.1 we show an example of a multiple choice question. There are several tags, and for each of them, there exist several attributes. Our investigation suggests that if `<zofar:answerOption>` tag has an attribute called ”variable”, then the text inside the ”label” attribute needs to be extracted as this answer option is important and contains valuable information. Similarly, for single-choice questions, we have extracted both questions and answer options. In addition, only the text inside the question tag is extracted for open-ended and matrix questions.

4.2 Exploratory data analysis

In order to obtain more insights into the dataset before using it for experimentation, we performed exploratory data analysis.

In order to see if the keywords and the keyphrases extracted have any relation with the length of the surveys, we started by checking the length of each survey (cf. Figure 4.2). For all subsequent experiments, the stopwords were removed. We see that survey *Promopanel_w3* has the highest number of words followed by *nacaps_2018*. In terms of shorter surveys, there was not one survey standing out in terms of fewer contents. Instead, the majority of the surveys shared the trend of smaller content.

Furthermore, we considered all the surveys as one corpus to check the most commonly occurring words across all the surveys. Afterwards, we plotted the top 20 frequently occurring words (cf. Figure 4.3). The frequency of words such as *arbeitgeber*, *themen* and *studium* is relatively high. Figure 4.4 also depicts the top 20 unigrams in the corpus through a word cloud.

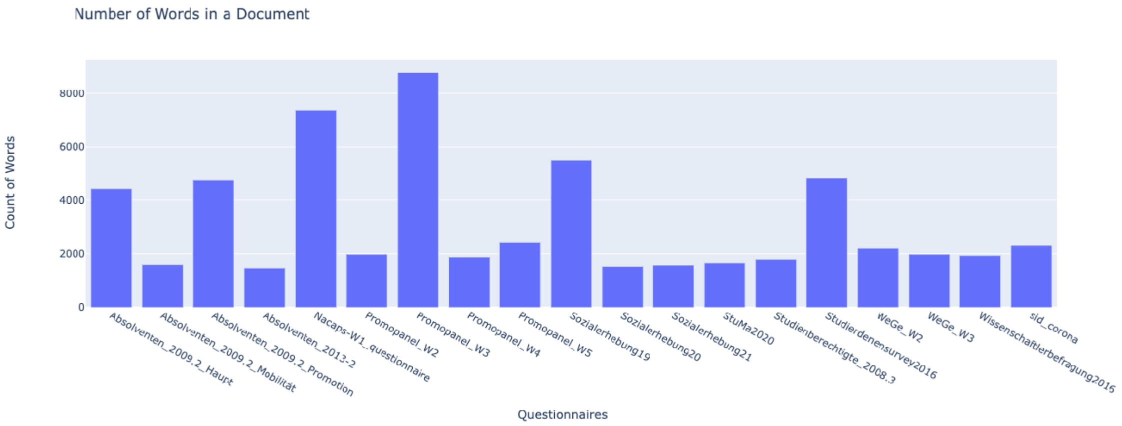


Figure 4.2: The figure shows the total number of words per document.

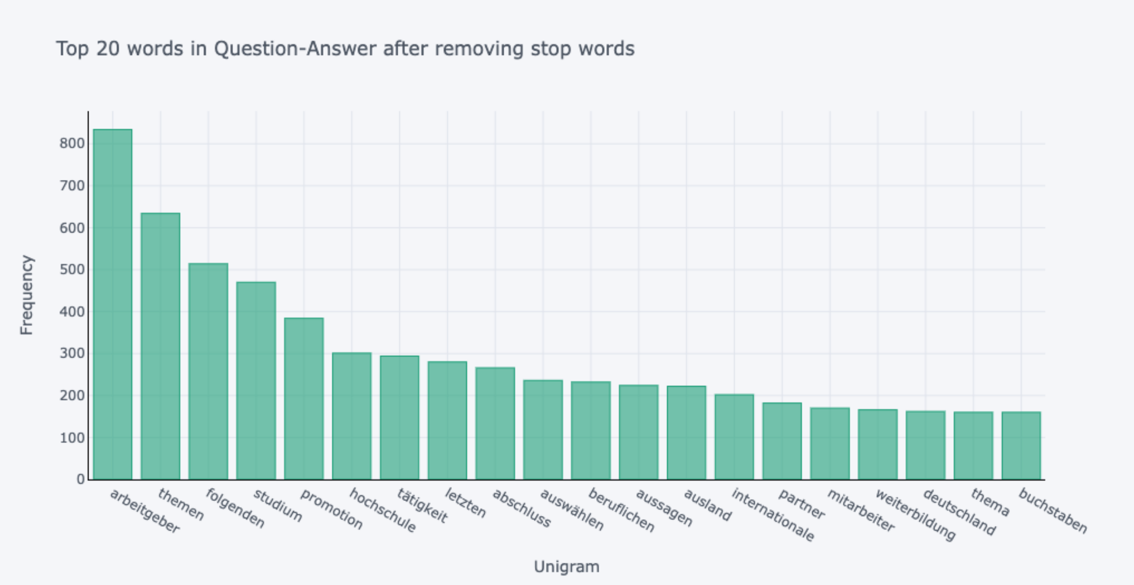


Figure 4.3: The figure shows top 20 unigram for the whole corpus.

Building on top of the same, we also want to see the most occurring bigram across the corpus (cf. Figure 4.5). In terms of bigrams, there was not a single bigram standing out, but multiple bigrams shared the popularity across the corpus. Figure 4.5 also depicts the top 20 bigram in the corpus through a word cloud.

We also wanted to see if the number of question-answer pairs has any effect on the keywords and keyphrases extracted. For this purpose, we counted the number of question-answer pair lengths across surveys(cf. Figure 4.7). The document which has the highest number of question-answer pairs is *Studierendensurvey2016* followed by *Nacaps-W1-questionnaire* and *Sozialerhebung19*.

In Figure 4.2, we saw that the questionnaire *Promopanel_w3* had a higher number of words than *Studierdenensurvey2016*, but the trend for the number of question-answer pairs is reversed. Even though there is a correlation between the words and the number of question-answer pairs, it does not necessarily mean that the survey with the maximum number of words would also have a higher number of questionnaire

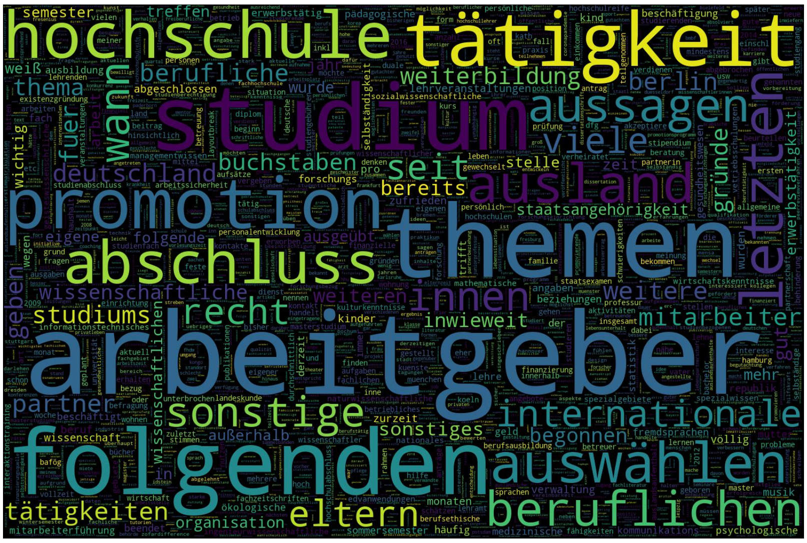


Figure 4.4: The figure shows the word cloud for top 20 words in the corpus.

pairs. This might have an effect on the keywords/keyphrases extracted by a particular method because we just take into consideration the question-answer pair and not the complete text for the process of extraction.

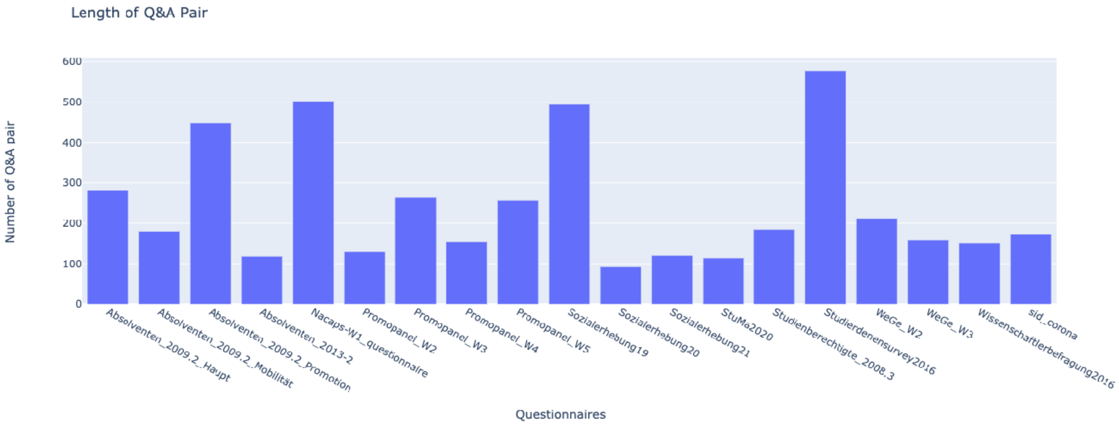


Figure 4.7: The figure depicts the total length of question answer pairs.

Based on the number of question and answer options, we have calculated the average word counts for them with respect to each document which is shown in Figure 4.8. We can see the distribution of questions and answers and say that the length of answers is always greater than the questions. One reason for this could be that apart from Likert scale questions, we also have multiple-choice questions where answer options can be more than one.

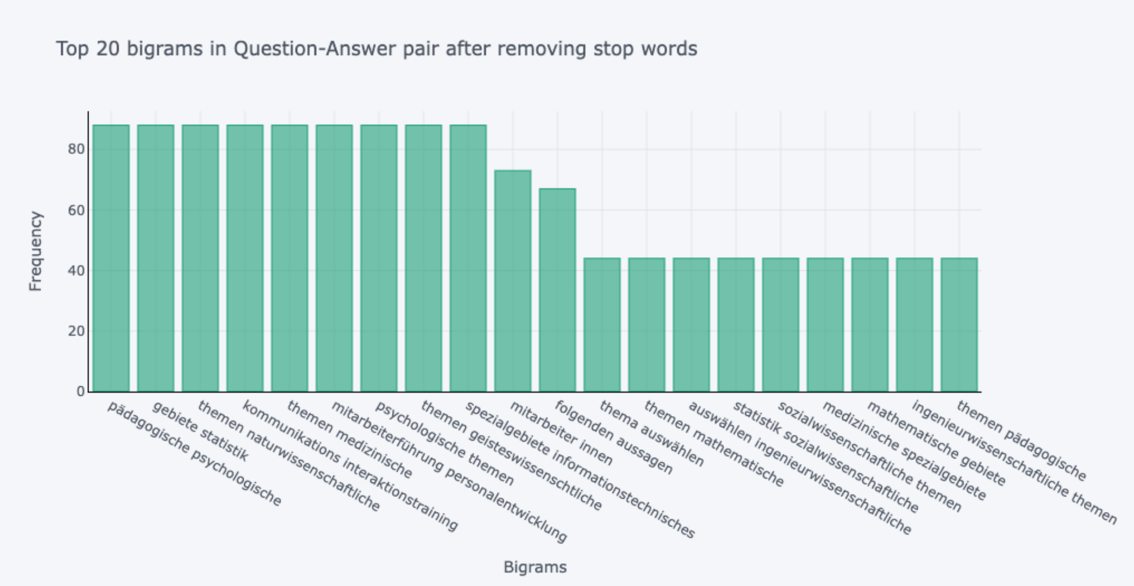


Figure 4.5: The figure shows top 20 bigram for the whole corpus.

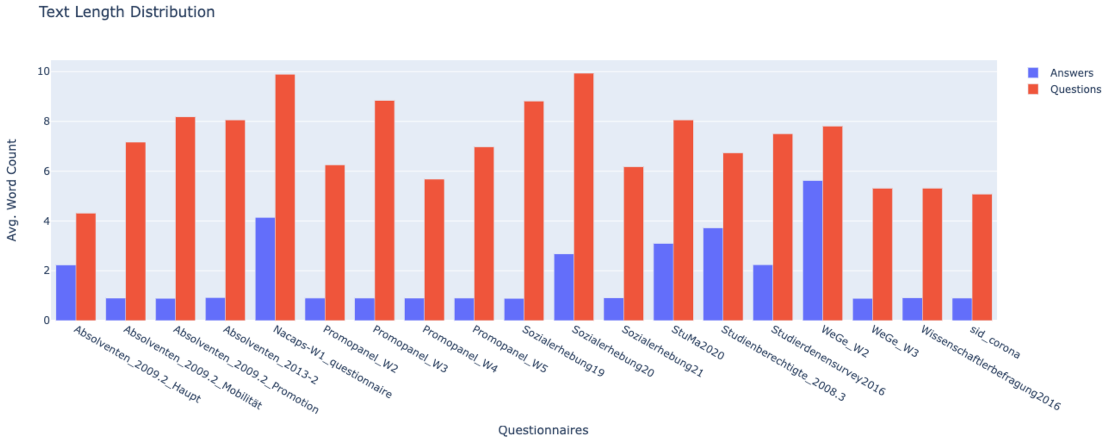


Figure 4.8: The figure depicts the average word count for questions and answers for each document

Moreover, in order to build the base towards the keywords/keyphrases extraction, we wanted to analyze the uniqueness of each questionnaire. To do so, for each questionnaire, we extracted the words that are unique to a particular survey. for example: if we have two documents d1 and d2, such that d1 is *Ich habe eine schriftliche Bewerbung eingereicht* and d2 is *Ich musste ein Auswahlverfahren durchlaufen, in dem mehrere Personen an der Entscheidung zur Besetzung der Stelle beteiligt waren.* After removing the stopwords, the unique words for d1 are 'schriftliche', 'Bewerbung', 'eingereicht' whereas, the unique word for d2 is 'Auswahlverfahren', 'Entscheidung', 'Besetzung', 'Stelle'. The uniqueness of each document is defined by the following formula:

$$Uniqueness(\%) = \frac{\text{Number of unique words}}{\text{Total number of words in a document}} * 100 \tag{4.1}$$

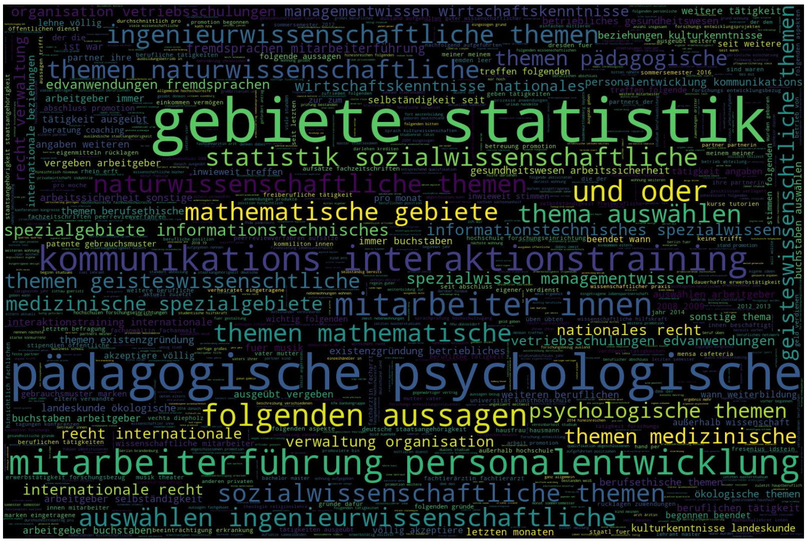


Figure 4.6: The figure shows the word cloud for top 20 bigram in the corpus.

In the example stated above, after plugging the values in equation 4.1, the uniqueness of d1 will be about 33% and for d2 it is about 22%.

Applying the same to survey data, we see that *Absolventen_2013_2* is the most unique questionnaire with a uniqueness of about 12.2% followed by *sid_corona* with a uniqueness of about 9%. On the contrary, the surveys like *Promopanel_W3*, *Sozialerhebung20* have a lot of overlapping words. Hence, there are not many unique words in those questionnaires.

We would like to analyze if this trend is also seen in the keywords/keyphrases that are extracted, meaning that surveys with less unique words have similar keywords/keyphrases and surveys with more uniqueness have unique top k extracted keywords/keyphrases.

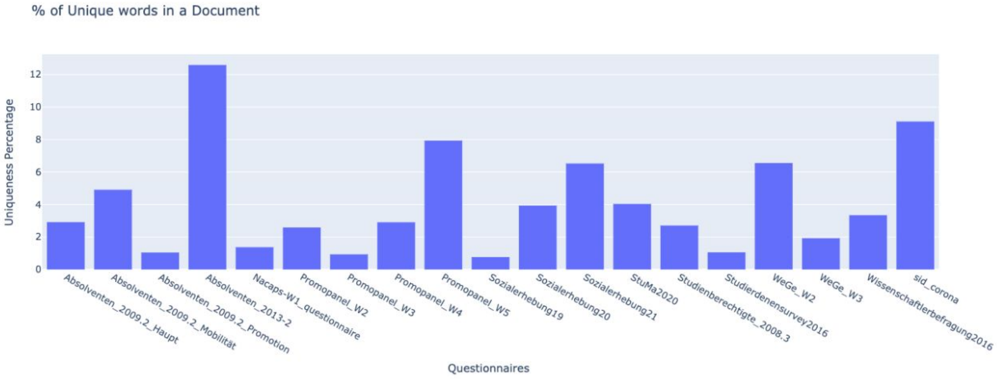


Figure 4.9: The figure shows the percentage of uniqueness of each document

4.3 Hand annotated keywords and keyphrases

In order to evaluate the experiments and effectiveness of methods a set of keywords and keyphrases are provided to us for each questionnaire. These keywords and keyphrases are manually annotated by DZHW scientists. An example of such keywords and keyphrases can be seen in the searchable front-end [7]. In the forthcoming chapters, we refer to these keywords and keyphrases as the *ground truth*. In figure 4.10, we can see that overall *ncaps_2018* has the highest number of keyphrases wherein, *Promopanel_W2*, *Promopanel_W3*, *Promopanel_W4* and *Promopanel_W5* has the highest number of keywords with the same frequency.

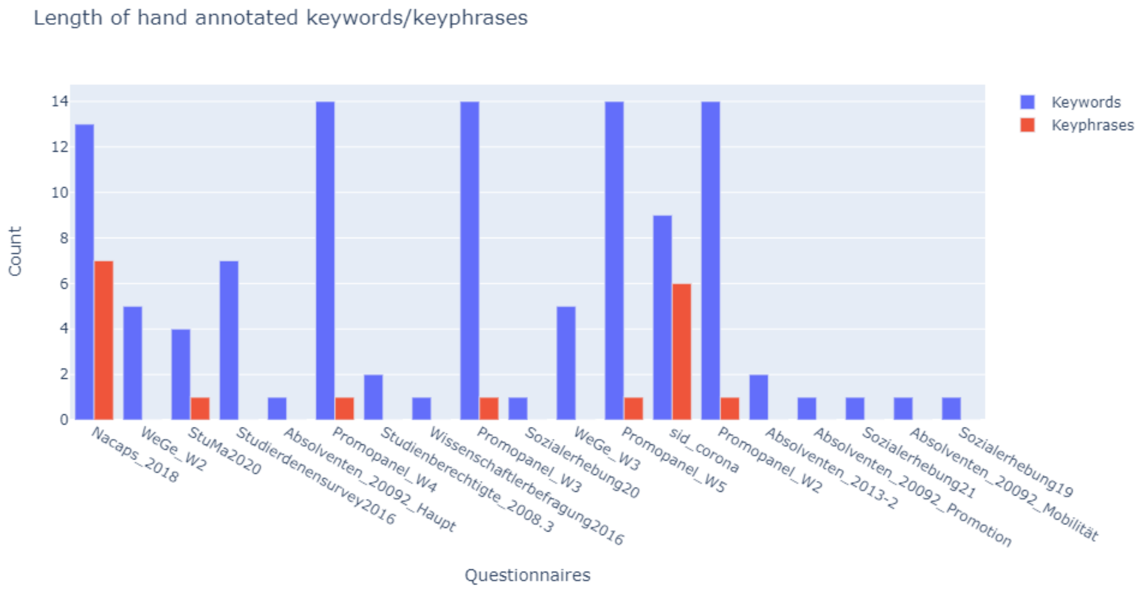


Figure 4.10: The figure shows the length for keywords and keyphrases for each questionnaire.

There were some surveys where the annotated keywords/keyphrases were not from the given survey text but were constructed from the domain knowledge of scientists. These keywords/keyphrases could not be extracted by any of the techniques that we have investigated because these techniques can only extract text that exists within the text. As a result, for our analysis, we limited the ground truth to only the keywords/keyphrases found in the text. In Figure 4.11, we can see that, out of 19 questionnaires there are only 10 questionnaires which have manually annotated keywords/keyphrases. *ncaps_2018* has the highest number of keywords and keyphrases which are within the questionnaire.

[7] https://metadata.fdz.dzhw.eu/de/data-packages/stu-nac2018?page=1&size=10&type=surveys&version=1.0.0

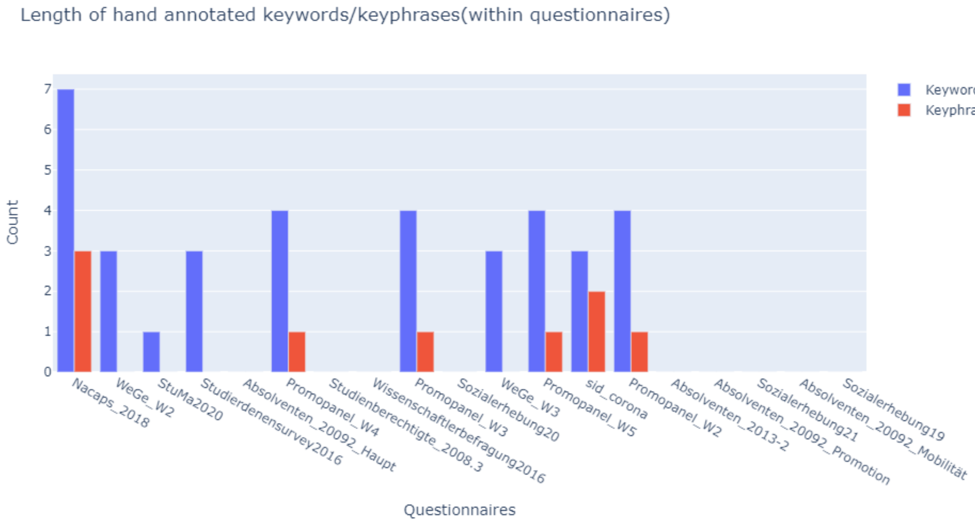


Figure 4.11: The figure shows the length of keywords and keyphrases within the text for each questionnaire.

4.4 Thesauri and Controlled Vocabularies

Thesauri are hierarchical methods that constitute a relationship for organizing knowledge which is frequently used in libraries to classify and index publications Ritze and Eckert [2012]. A thesaurus contains synonyms, other word forms, and occasionally even antonyms for each term. Whereas, controlled vocabularies are set of terms that are standardized and used to group information for tasks like information retrieval and indexing. It can make browsing and searching easier. Finally, a thesaurus is a more organized kind of controlled vocabulary that offers details about each phrase and its connections to other terms in the same thesaurus Hedden [2008].

In this thesis, thesauri and controlled vocabulary from the social science domain are chosen for the experimentation Section 5.5. More details about the selected thesauri and controlled vocabularies are mentioned below.

TheSoz - Thesaurus for the Social Sciences

The Thesaurus for the Social Sciences (TheSoz) [8] is a German thesaurus which covers the field of social sciences. It is an essential tool for indexing documents and research information as well as for search term recommendation.

TheSoz is available in three different languages: German, English and French. It contains a total of 12,000 keywords, out of which 8,000 are so-called descriptors or preferred terms for indexing documents, and 4,000 are non-descriptors, or non-preferred terms, for which preferred terms are recommended to be used instead. The thesaurus covers all areas and divisions of the social sciences, such as political science, pedagogy, employment research, and sociology Zapilko et al. [2013].

The European Language Social Science Thesaurus (ELSST)

[8] https://lod.gesis.org/thesoz/de/index

The European Language Social Science Thesaurus (ELSST) [9] is a comprehensive, multilingual thesaurus for the social sciences. It is owned and distributed by the Consortium of European Social Science Data Archives (CESSDA) and its national Service Providers. This thesaurus has approximately 3000 concepts covers the main fields of social science, including politics, sociology, economics, education, law and health.

ZBW Standard-Thesaurus Wirtschaft

The standard thesaurus for economics [10] consists of vocabulary and standardized keywords related to economics. The thesaurus includes terminology for all economic topics, with about 6,000 keywords and more than 20,000 extra synonyms to help users find specific search terms. There are also technical terminologies from fields like law, sociology, or politics as well as geographical terms.

CESSDA Controlled Vocabulary

CESSDA Controlled Vocabulary consists of vocabularies and terms which covers topics from social science domain for instance economics. Version 4.2 [11] of controlled vocabulary includes about 95 terms. The vocabularies consist of 11 different languages.

4.5 Summary

In this Chapter, first we describe the outline of the dataset where the types of surveys are mentioned. Furthermore, we have elaborated the text extraction from the questionnaires, where we take an example of XML snippets and explain each parts. After that, an emphasis is made on the types of survey questions, where we explain that several question types are exist within a given questionnaire such as single choice questions, multiple choice questions. For each of these question tag, we have created a function to extract the questions and answer option. Moreover, Exploratory data analysis is performed in order to obtain the insights about our dataset. After this, we elaborate and analyse the hand annotated keywords and keyphrases that are provided to us for experimentation. Finally, we have explained the thesauri and controlled vocabularies from social science domain that are used for our experimentation.

[9] https://thesauri.cessda.eu/elsst-3/en/

[10] https://zbw.eu/stw/version/9.12/about.de.html

[11] https://vocabularies.cessda.eu/vocabulary/TopicClassification?lang=en

5. Concept

This chapter outlines the methodology of the keyword and keyphrase extraction process used in this thesis work. Initially, we will present the workflow, followed by a brief explanation of each module. Next, a detailed explanation for each part of our concept will be presented.

1. **Text Extraction and Pre-Processing:** Relevant text is extracted by parsing the questionnaires, which was represented in XML format. A Detailed explanation of the same is given in Chapter 4 and in Section 5.1.

2. **Approaches:** In order to extract keywords and keyphrases from the questionnaires we have selected some approaches. The first approach is TF-IDF which is a statistical method. The second approach is TextRank which is a graph-based method. Finally, the last approach we have used is KeyBERT which is an embedding-based approach. Further explanation of how these methods are put together in our methodology is given in Section 5.2.

3. **Qualitative Assessment:** Once an approach results in either a set of keywords or keyphrases, we have conducted some qualitative analysis, which includes manually looking at those keywords or keyphrases and finding some insights. Elaboration of this qualitative assessment is given in Section 5.3.

4. **Fine-Tuning and Post-Processing:** A POS tagger is used, which helps in assigning each word to a part of speech category. After that, we created a sequential regex pattern to extract a good set of keyphrases. Further explanation can be found in Section 5.4.

5. **Incorporation of Domain Knowledge:** External resources such as thesauri and controlled vocabularies from the social science domain are used in order to make the process efficient for extracting keywords and keyphrases which improves the index search. More elaboration of the same can be found in Section 5.5.

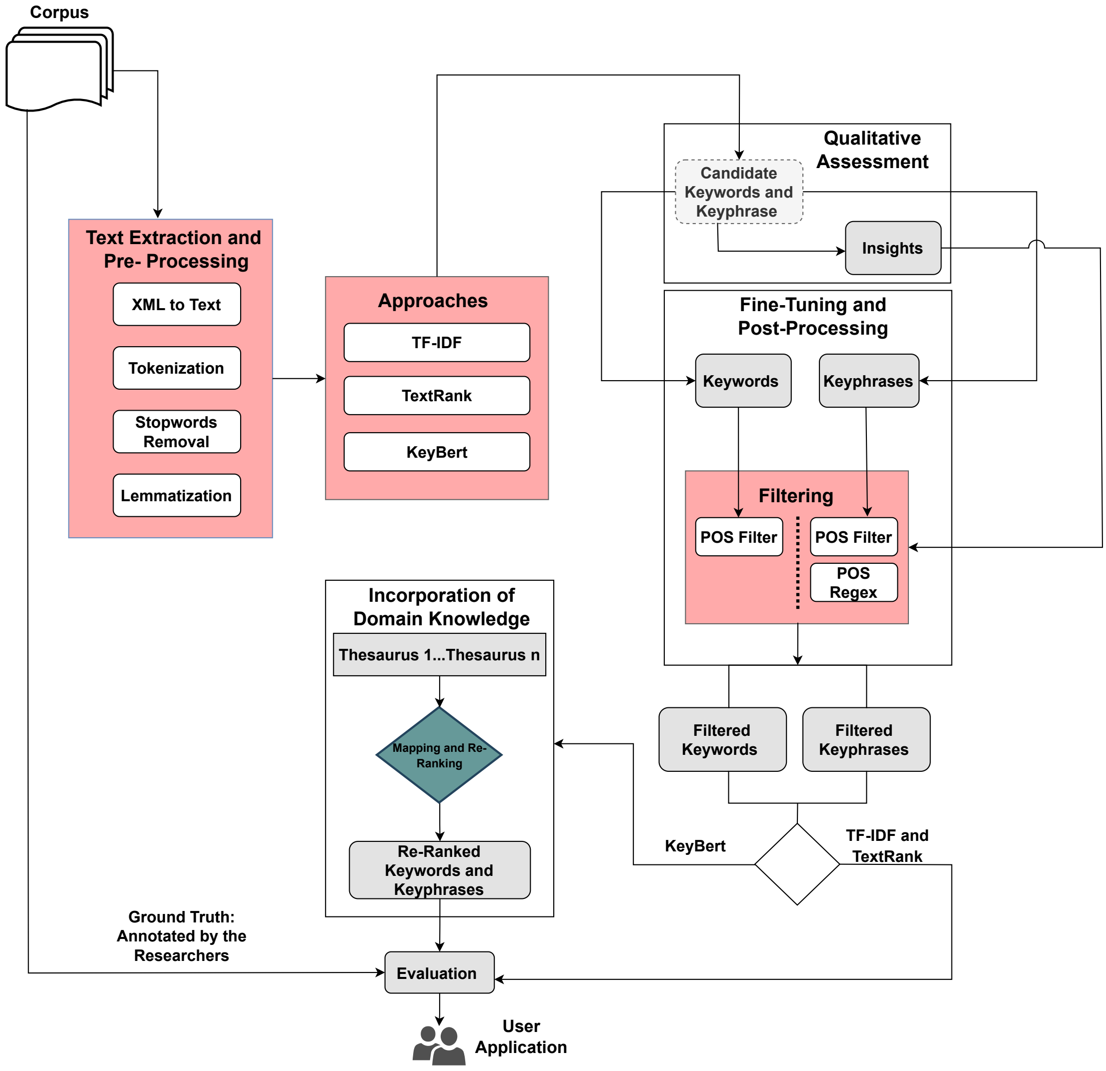


Figure 5.1: The figure shows a workflow of keyword and keyphrase extraction which consist of different modules. The first modules depicts the representation of data and pre-processing steps. The second module shows the approaches that are used. Another components used was qualitative assessment and fine-tuning and post-processing which was necessary to get refined set of keywords and keyphrases. The final components depicts the incorporation of external resources such as thesauri, controlled vocabulary which further helps in re-ranking the keywords and keyphrases.

5.1 Text Extraction and Pre-Processing

Working with most natural language processing tasks and techniques requires natural language in text format, especially when working with keyword and keyphrase extraction tasks. The dataset used in this work is in the form of questionnaires and initially stored in XML format. First, it is transformed into text format and afterwards the extraction of relevant information comes into the picture.

As mentioned in Chapter 4, questions and answer choices from the questionnaires are the main text content needed for extraction purposes. The first step is to extract both and on top of that, pre-processing is implemented because before using the data it must be as clean and structured as possible. Text pre-processing is an essential step in natural language processing tasks. It assists in filtering all of the irrelevant information from data for text to transform into a more analyzable form so that the NLP techniques can perform better.

In this work, some of the text pre-processing techniques used to clean our data are tokenization, stopword removal, and lemmatization, which are explained in Chapter 2.

After arranging the dataset in working order, three methods have been explored to extract keywords and keyphrases.

5.2 Approaches

In Figure 5.1, we visualize three different methods for extracting keywords and keyphrases. The methods includes TF-IDF, TextRank and KeyBERT. This section elaborates on how these approaches were employed in our use case.

5.2.1 TF-IDF

One of the three techniques includes $TF - IDF$, which stands for Term Frequency-Inverse Document Frequency, a popular statistical measure for term extraction, which is used as a baseline method in this thesis.

The reason why TF-IDF is considered as a baseline in this thesis is because, it is simple to comprehend, easy to compute, and it is one of the most flexible statistics for determining the relative importance of a term or phrase in a document or set of documents with the rest of the corpus.

As explained in Chapter 2, term frequency represents how frequently a particular word appears in a sentence or document.

$$TF(t) = \frac{\text{Number of times term t appears in a document}}{\text{Total Number of terms in the document}}$$

If we have a sentence, *Hat sich die Corona Pandemie auf die Einkommenssituation Ihrer Eltern ausgewirkt?* . In this particular sentence, the term frequency for the word 'Eltern' is 1/11 because the number of words in this sentence is 11, and the word 'Eltern' has occurred only once. Similarly, the term frequency for the word 'die' is 2/11 because the word 'die' has occurred twice.

As elaborated in Chapter 2, IDF stands for inverse document frequency, which indicates how frequently a term has occurred across all the documents. If a term has occurred more frequently across the different documents, then the *idf* value would be low because it is an inverse of the document frequency. Similarly, if a particular term has occurred more infrequent across different documents, then the *idf* value would be higher.

$$IDF(t) = Log\left(\frac{\text{Total Number of documents}}{\text{Number of documents containing term t}}\right)$$

For example, if we have two sentences.

- *Beratung zu allgemeinen Fragen rund um die Promotion*
- *Beratung zu Konflikten im Promotionskontext*

The idf for the word 'Beratung' will be log(2/2) because the total number of documents is two, and the word 'Beratung' has occurred in both documents. Likewise, the idf for the word 'allgemeinen' would be log(2/1) because the total number of documents is two, but the word 'allgemeinen' has occurred in only one document.

TF-IDF is nothing but the product of both term frequency(TF) and inverse document frequency(IDF). It gives less weightage to words repeated across the documents but more weightage to words used in fewer documents but repetitively in the same documents. So, if a word frequently occurs in one document but is very rare across documents, then it is an important word or keyword for that document.

To understand the concept better and how the keywords are extracted using the TF-IDF algorithm, suppose there are three documents in a corpus, and each sentence is considered a document, and the words in the sentence are tokens. The first step is to create a document-term matrix which is shown in Table 5.1. There will be a separate column for each word, and within each column, the count of that particular word is mentioned for the respective documents or sentences.

Documents	Beratung	Zur	individuellen	Karriereentwicklung	zu	Konflikten	im	Promotionskontext
Beratung zur individuellen Karriereentwicklung	1	1	1	1	0	0	0	0
Beratung zu Konflikten im Promotionskontext	1	0	0	0	1	1	1	1
Beratung zur Karriereentwicklung	1	1	0	1	0	0	0	0

Table 5.1: The table shows a document-term matrix which represents the frequency of terms that occur in a collection of documents.

Once we have the document-term matrix, the next step is calculating the term frequency. For each document, term frequency is computed as shown in Table 5.2.

Documents	Beratung	Zur	individuellen	Karriereentwicklung	zu	Konflikten	im	Promotionskontext
Beratung zur individuellen Karriereentwicklung	1/4	1/4	1/4	1/4	0	0	0	0
Beratung zu Konflikten im Promotionskontext	1/5	0	0	0	1/5	1/5	1/5	1/5
Beratung zur Karriereentwicklung	1/3	1/3	0	1/3	0	0	0	0

Table 5.2: The table depicts term frequency values for each documents.

For the first document, which is *Beratung zur individuellen Karriereentwicklung*, the word 'Beratung' has appeared once, and the total number of words in the documents is four, so the term frequency for this word will be 1/4. Similarly, the term frequency is calculated for all.

After calculating the term frequency, the next step is computing the inverse document frequency. The respective inverse document frequency has been calculated for each document, as shown in Table 5.3

Beratung	Zur	individuellen	Karriereentwicklung	zu	Konflikten	im	Promotionskontext
log(3/3)	log(3/2)	log(3/1)	log(3/2)	log(3/1)	log(3/1)	log(3/1)	log(3/1)
0	0.18	0.48	0.18	0.48	0.48	0.48	0.48

Table 5.3: The table shows inverse document frequency values for all the terms.

For example, the idf value for the word 'Karriereentwicklung' is log(3/2), which is 0.18 because the total number of documents is three and the number of documents where the word 'Karriereentwicklung' has appeared is two. Similarly, for each term the idf value is calculated. The log is used mainly for scaling purposes.

Once we have both the term frequency and inverse document frequency, the multiplication between them is performed to get the TF-IDF value shown below in Table 5.4.

Documents	Beratung	Zur	individuellen	Karriereentwicklung	zu	Konflikten	im	Promotionskontext
Beratung zur individuellen Karriereentwicklung	0	0.045	0.12	0.045	0	0	0	0
Beratung zu Konflikten im Promotionskontext	0	0	0	0	0.096	0.096	0.096	0.096
Beratung zur Karriereentwicklung	0	0.06	0	0.06	0	0	0	0

Table 5.4: The table shows TF-IDF values with respect to each documents

Table 5.4 represents TF-IDF values found at a document level. However, to summarize the TF-IDF values at a corpus level, we take the average of the TF-IDF values across the documents.

Beratung	Zur	individuellen	Karriereentwicklung	zu	Konflikten	im	Promotionskontext
0	0.035	0.04	0.035	0.032	0.032	0.032	0.032

Table 5.5: The table shows average of TF-IDF values across the documents

After computing the average of TF-IDF values across the documents, the keywords are identified based on their TF-IDF score. Apart from the word 'Beratung', the TF-IDF algorithm has identified the remaining words as keywords with the highest TF-IDF score of 0.04 for the word 'individuellen'.

Similarly, for keyphrase extraction, a POS pattern using the Regex expression is created, which will then be passed to the KeyphraseVectorizer in order to retrieve the relevant keyphrases. The elaboration of this process is further discussed in Section 5.4.

The drawback of TF-IDF is that it also consider preposition like 'zu' , 'zur' also as keywords, so sometimes non-keywords may get high TF-IDF score assigned.

There are several ways of handling this, and one way is preparing a list of stop-words that includes preposition, pronouns, and conjunctions. There are also predefined libraries such as NLTK [12] which has a list of predefined stopwords, and the stopwords can be removed from the corpus before applying the TF-IDF algorithm.

5.2.2 TextRank

The second approach that has been incorporated in this work is TextRank which is an unsupervised graph-based keywords and keyphrase extraction technique. The underlying premise of TextRank is that a word is more likely to be significant in a document if it frequently co-occurs with other significant words in the document. The primary distinction between TextRank and TF-IDF is that TextRank assigns term weights based on the context of words Lee et al. [2008]. In a given document, TextRank first creates a word graph and the connections between words show how they are related semantically, which is often determined by the co-occurrence of words in the document Ying et al. [2017]. Once the corpus is defined, the first step is to segment each document into sentences. Afterwards, for each sentence, the vertices are identified in the graph based on the part of speech tags of the text unit, which helps in providing potential nodes in the graph and then identify the relations that connect these vertices. The relations are based on the co-occurrence window, typically a window size of three, where we look at the adjacency of nouns, adjectives and repeated instances of the same words. The relations become the edges between those vertices. When the graph is finally constructed, the next step is to run the PageRank algorithm and iterate on the graph until it converges. After that, the vertices are sorted in the graph based on their score from the ranking algorithm, which will in turn be used for extracting the most referenced entities within the text, which will be either keywords or keyphrases.

For building the graph we have a taken a sample example from the questionnaire text. *"Welches Ergebnis haben Sie bei der Sprachprüfung für den Hochschulzugang DSH TestDaF oder andere Prüfung erhalten. Ich habe. Ich habe das Niveau. Ich habe die Prüfung bestanden aber weiß das Ergebnis nicht mehr. Ich habe die Prüfung nicht bestanden aber weiß das Ergebnis nicht mehr. Ich habe noch kein Prüfungsergebnis erhalten."*

In figure 5.2, a small graph is shown for the above text example. For the illustration purpose we have just taken subset from the example. In this graph, we have taken unigram words to be the vertices of graph. However, in post-processing step the candidate keyphrases are also considered.

In the table, 5.6, keywords which are extracted by TextRank from this text sample is shown. We have only mentioned the top-ranked keywords with their Textrank importance value.

5.2.3 KeyBERT

KeyBert is an embedding-based technique that also captures semantic value in the extraction process compared to a statistical and graph-based approach used in this thesis work.

[12] https://www.nltk.org/index.html

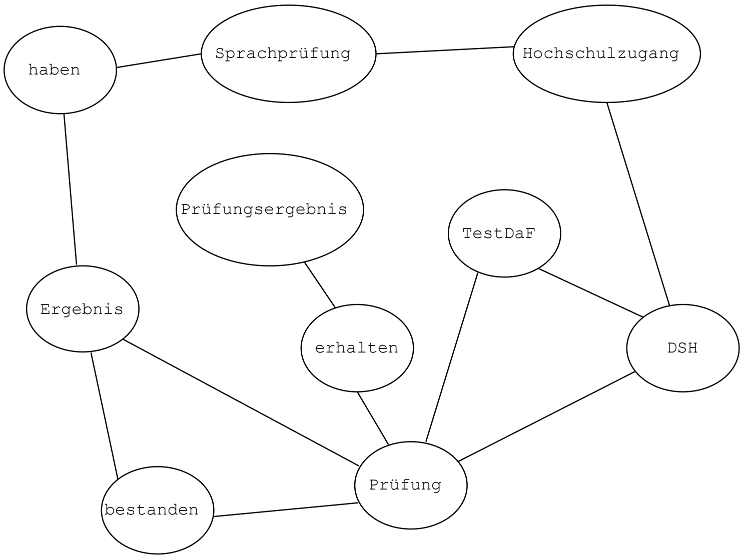


Figure 5.2: The figure depicts an illustration of TextRank by showing a sample example of graph for the given text.

Keyword extracted by TextRank	Importance score
prüfung	0.07
ergebnis	0.06
bestanden	0.052
hochschulzugang	0.042
sprachprüfung	0.037
prüfungsergebnis	0.028

Table 5.6: The table shows the lexical units or keywords extracted by TextRank with their importance score from the sample example.

It uses BERT language models in the process of extracting keywords and keyphrases. KeyBert first creates the BERT embedding of document texts, and then creates candidate keywords and keyphrases from the document by setting a predefined length of n-gram. The next step is to convert both the document as well as the candidate keywords and keyphrases to numerical representation that is embedding. The final step is to find candidates that are most similar to the document. In order to calculate the similarity between the documents and candidate keywords/keyphrases, cosine similarity is used. Once the similarity is calculated, the top n most similar candidates to the input documents result in keywords and keyphrases.

Keyword extraction

For the KeyBert approach, instead of considering complete questionnaires as a document, the document has is split into multiple smaller documents. Wherein each of the smaller documents is the question-answer pair. The basic principle of KeyBERT is to take similarity between the document as a whole and its corresponding tokens. Since in our case we have various forms of question-answer pair inside a single questionnaire(likert scale, multiple choice questions) (cf. Figure 5.4 and Figure 5.5) that differ in content as well as the stylistic aspect. We wanted to capture the local keyword that might be of importance to each of the question-answer pairs and hence decide to split into the multiple small documents.

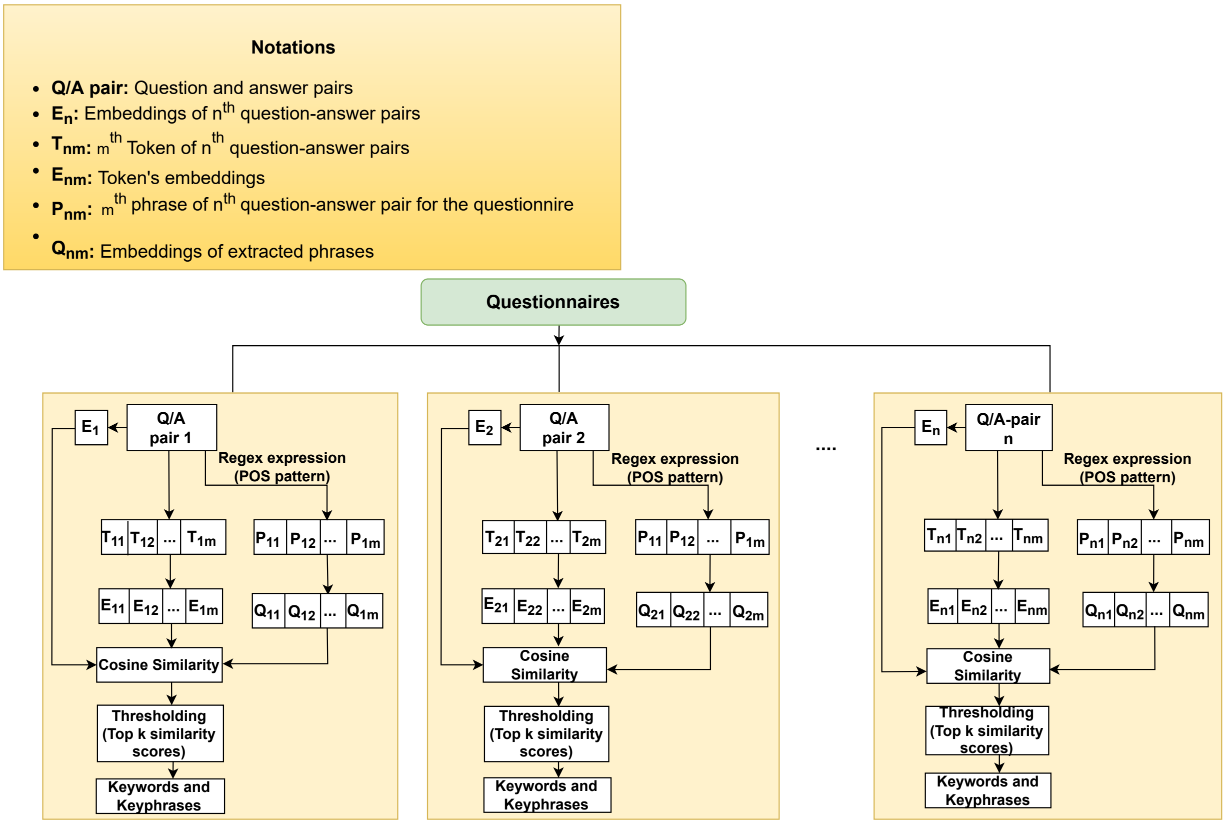
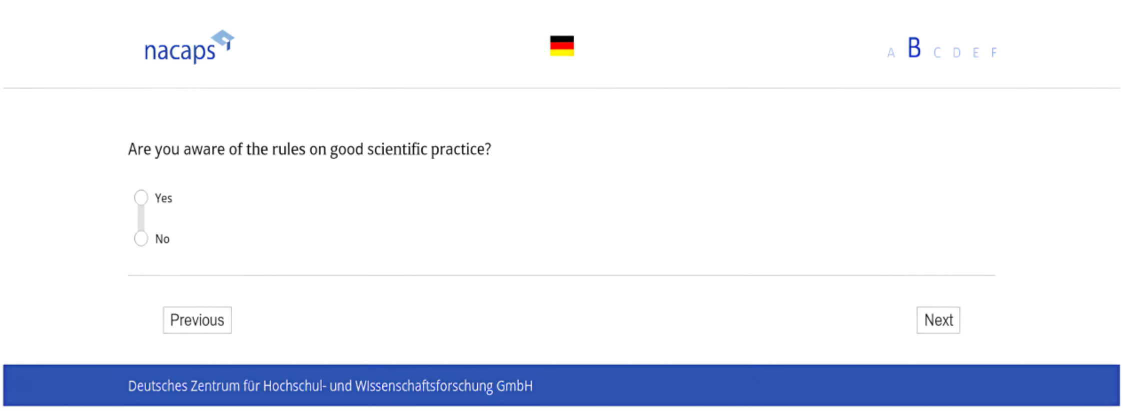

Notations
Q/A pair: Question and answer pairs
E_n: Embeddings of n^{th} question-answer pairs
T_{nm}: m^{th} Token of n^{th} question-answer pairs
E_{nm}: Token's embeddings
P_{nm}: m^{th} phrase of n^{th} question-answer pair for the questionnire
Q_{nm}: Embeddings of extracted phrases
Questionnaires
E_1
Q/A pair 1
Regex expression (POS pattern)
T_{11} T_{12} ... T_{1m}
P_{11} P_{12} ... P_{1m}
E_{11} E_{12} ... E_{1m}
Q_{11} Q_{12} ... Q_{1m}
Cosine Similarity
Thresholding (Top k similarity scores)
Keywords and Keyphrases
E_2
Q/A pair 2
Regex expression (POS pattern)
T_{21} T_{22} ... T_{2m}
P_{11} P_{12} ... P_{1m}
E_{21} E_{22} ... E_{2m}
Q_{21} Q_{22} ... Q_{2m}
Cosine Similarity
Thresholding (Top k similarity scores)
Keywords and Keyphrases
....
E_n
Q/A-pair n
Regex expression (POS pattern)
T_{n1} T_{n2} ... T_{nm}
P_{n1} P_{n2} ... P_{nm}
E_{n1} E_{n2} ... E_{nm}
Q_{n1} Q_{n2} ... Q_{nm}
Cosine Similarity
Thresholding (Top k similarity scores)
Keywords and Keyphrases

Figure 5.3: The figure shows the process of keyword and keyphrase extraction using the KeyBERT. For keyword extraction, given a question-answer pair, its embedding is first created and then tokenized. After that, embedding for these tokens is computed and then calculate the cosine similarities between the question-answer pair embeddings and token embeddings. Finally, after the thresholding, we extract the top k keywords. Similarly, for keyphrase extraction, we calculate phrase embeddings instead of token embeddings, and the rest is similar to keyword extraction.

In the next step, the embedding for these question-answer pairs is created using the pre-trained models *symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli*. It is a sentence-transformers model that helps in mapping sentences and paragraphs to a 768-dimensional dense vector space which represents the semantic aspect of the document. This model was chosen since it is known for working with several languages, and the text we are using in this thesis is in German [13].

In our case, the embeddings we get for the question-answer pairs are called document embeddings E_n cf. Figure 5.3. Afterwards, from the same document, word embeddings of the words are retrieved for N-gram range as one(N-gram=1) i.e, every question answer pairs are tokenized using the CountVectorizer and for each token

[13] https://huggingface.co/symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli



Are you aware of the rules on good scientific practice?

○ Yes

○ No

Deutsches Zentrum für Hochschul- und Wissenschaftsforschung GmbH

Figure 5.4: The figure shows an example of a single choice question from the questionnaire.

T_{nm} (m^{th} tokens of n^{th} question-answer pairs) its corresponding embedding E_{nm} (embedding ofm^{th} token from n^{th} question-answer pair) is computed using the same model which is used for embedding the documents. Furthermore, cosine similarities are computed between E_n (Embeddings of n^{th} questions-answer pair) and the embedding for all the tokens T_{nm} inside the question-answer pair.

For example: lets consider a sentence which is, *Ich habe einen Arbeitsvertrag an diesem Forschungsinstitut.*

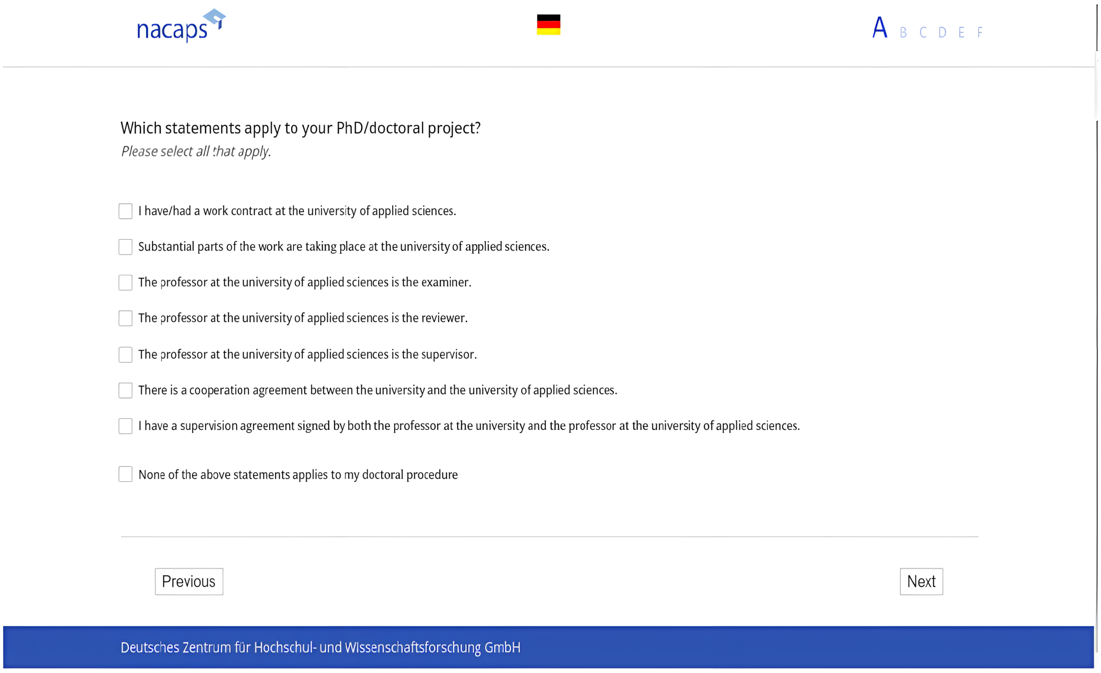
We begin by determining how similar the first word, *Ich*, is to the entire phrase. The same is repeated for the second word, which is *habe*, and similarly, when we are finished with all the words, we sort the words in the descending order of their similarity with the complete sentence (question-answer pair in this case). Finally, we conclude by choosing top k similar keywords.

Once the similarity values for each word is computed, average of all the values are taken for further process. For example, in the end, if 20 keywords are extracted and after computing the average, we get a value, lets say 0.8, we would then select the keywords within the set of those 20 that had a similarity value greater than 0.8, and that is how we define the k.

Keyphrase Extraction

The way Keyphrases are extracted using KeyBert is similar to how keywords are extracted using the method. In the first step, question-answer pair text are embedded as E_n (cf. Figure 5.3) using the pre-trained models. After that, using the KeyphraseVectorizers, phrases are extracted with part of speech pattern. Afterwards, for each candidate phrase P_{nm}, its corresponding embeddings Q_{nm} are computed. Finally, cosine similarities are calculated between document embeddings E_n and the embedding for all the phrases Q_{nm} inside the question-answer pair in order to extract keyphrases that best describe the whole document.

The only difference between extracting keywords and keyphrases is that, in order to extract keyphrase from a given questionnaire, pattern matching Rabby et al. [2018] is used by leveraging the part of speech, i.e., from the provided manually annotated keyphrases, we checked what pos pattern these keyphrases follow and based on that,


nacaps
A B C D E F
Previous
Next
Deutsches Zentrum für Hochschul- und Wissenschaftsforschung GmbH

Which statements apply to your PhD/doctoral project?
Please select all that apply.

- □ I have/had a work contract at the university of applied sciences.
- □ Substantial parts of the work are taking place at the university of applied sciences.
- □ The professor at the university of applied sciences is the examiner.
- □ The professor at the university of applied sciences is the reviewer.
- □ The professor at the university of applied sciences is the supervisor.
- □ There is a cooperation agreement between the university and the university of applied sciences.
- □ I have a supervision agreement signed by both the professor at the university and the professor at the university of applied sciences.
- □ None of the above statements applies to my doctoral procedure

Figure 5.5: The figure shows an example of multiple choice question with its answers options from the questionnaire.

made a pos regex Kaur [2014] pattern. A more detailed explanation of the same can be found in Section 5.4.

After computing the similarity values for each phrase the next step is to take the average of all the computed values. For example, in the end if a method extract 30 keyphrases and after computing the average we get a value lets say 0.4. After that, we would select the keyphrase within the set of those 30 keyphrases that had a similarity value greater than 0.4 and which in-turn helps us in defining the k.

5.3 Qualitative Assessment

Commonly, the approaches extract keywords by breaking down the document's content into a list of candidate keywords, defined as the unfiltered lists of keywords and keyphrases. A part of speech tagger (POS tagger) is typically used to filter out some terms while choosing candidate words. However, in this study, the steps that have been taken is, first, we apply the approaches, and secondly, we filter the words and output the final set of keywords and keyphrases based on POS tags and sequential regex pattern, that is why in the framework diagram (cf. Figure 5.1) it is denoted as dashed box and slightly less grey compared to others.

At the end, when an approach results in either a set of keywords or keyphrases, the next step is that we conduct some qualitative analysis in which we manually look at those keywords or keyphrases and find some insights. After that, we used both ground truth and the extracted keyphrases to make those insights. For instance, for the keyphrases, we extracted the POS tag of each of the components of each phrase(cf. Figure 5.6) and then counted the co-occurrences of consecutively occurring POS tags. After this, we designed a POS pattern using the top occurrences which is explained in the next section(cf. 5.4). Similarly, for the keywords, we manually assigned the extracted keywords and saw that the most relevant output as per our domain was noun, adjectives, and verbs.

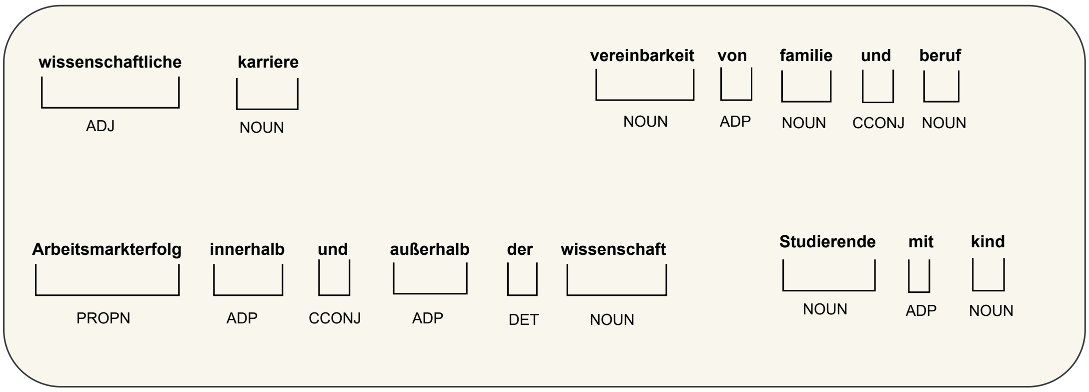


Figure 5.6: The figure shows some examples of hand annotated keyphrases that were available and used as a ground truth for experimentation.

Based on the obtained insights and findings, it was decided that which keywords or keyphrases are relevant or not relevant for our use case. More elaboration on how POS tagger and the regex pattern have been incorporated and applied on the questionnaire itself is discussed in Section 5.4.

5.4 Fine-Tuning and Post-Processing

Previous work on keywords and keyphrase extraction Jayasiriwardene and Ganegoda [2020] Rungta et al. [2020] Ushio et al. [2021] are based on methods which work either for keywords or keyphrases, however, the framework that has been designed in this thesis for keywords and keyphrase extraction consists of a dual pipeline, one is purely for extracting keywords, and the other is for extracting keyphrases, with the ability to toggle between the two. There are basically a few justifications for that.

Firstly, when we were trying to extract both keywords and keyphrases together using the approaches mentioned, the extraction process was not optimized that is we were not able to extract either good keywords or keyphrases.

Another reason that contributes to the dual pipeline architecture is the sequential regex pattern that was used which means when we tried to extract both keywords and keyphrases using the pattern, we missed out on a number of good keywords.

5.4.1 Keyword Extraction

The first phase of the pipeline involves extracting only the keywords through the approaches that are mentioned in Figure 5.1.

Once the qualitative assessment Section 5.3 is finished for keywords, afterward, we have decided to use a POS tagger which helps in assigning each word to a part-of-speech category which can be noun, adjectives, pronouns and so on.

Also as seen in Chapter 3 Hulth [2003] Mihalcea and Tarau [2004] have shown that primarily nouns and adjectives are the most relevant pos tags to extract relevant keywords, our initial experiments, as talked about in Section 5.3 also confirm this line of thinking.

However, in our system, we also consider verbs as POS tags for extracting keywords, as the combination of these POS tags resulted in a good set of keywords.

5.4.2 Keyphrase Extraction

The second phase of the pipeline consists of only extracting the keyphrases using the approaches that have been mentioned. Keyphrases are widely used in order to have a summary of documents.

Identifying phrases with POS patterns depends on linguistic analysis, which finds candidate phrases based on pre-defined POS patterns. For example, as seen in the work of Liu et al. [2009] and Wan and Xiao [2008], a POS pattern which is widely used is represented as <J.*>*<N.*>+. It extracts keyphrases with 0 or more adjectives followed by one or more nouns. However, in this thesis work, we have defined a more complex POS pattern to extract keyphrases from the questionnaires. This pattern was designed based on the experiments discussed in Section 5.3, and the same is represented using a regular expression as follows:

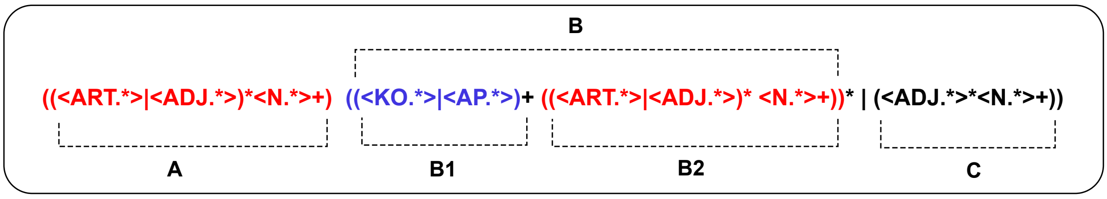


Figure 5.7: The figure show POS pattern that is created to extract keyphrases from the questionnaires.

In order to translate the created POS pattern cf. 5.7, we have split it into three parts. The 'C' part is the default pattern and passed in the parameters of KeyphraseCountVectorizer, which extracts keywords with 0 or more adjectives followed by 1 or more nouns [14]. This pattern is extended and inside that two more parts are added. The first part, 'A' tells that the keyphrases can start with an article followed by adjectives or it can only start with an adjectives, but it should always have to be followed by a noun. To get the longer keyphrases, we have incorporated part B, which again split into parts B1 and B2. The first part, B1, adds the possibility of getting conjunctions like 'und' , 'oder', or it can include adpositions like 'von', 'zu', it has to be at least one, but it can also be several. The following pattern, ' B2', is similar to pattern 'A'.

This pattern is based upon the manually annotated keyphrases that were available as ground truth for the experimentation, as seen in Figure 5.6. For example: if we want

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

to show how the regex pattern match the keyphrase which is, ”arbeitsmarkterfolg innerhalb und außerhalb der wissenschaft”.

Part A matches the first word that is *arbeitsmarkterfolg* which is just a noun, in this case a proper noun. Part *B1* is a repetitive pattern which means it has to come atleast one or many times. So from the keyphrase it matches three time(because we have '+' is the pattern) first, *innerhalb* which is an adposition, second, *und* which is a conjunction and finally, *außerhalb* which is a adposition. Part *B2* matches *der wissenschaft* because in the pattern we have an article denoted by ART* and then it can be followed by a noun. so it matches the determiner which is *der* followed by noun which is *wissenschaft*. So, for this keyphrase it matches part A, B1 and B2 completely.

5.5 Incorporation of Domain Knowledge

We saw in Chapter 4 that the number of keywords/keyphrases available for each questionnaire is limited and does not follow a consistence usage of words for the same concept. For instance, we saw the use of two different keywords, “promotion” and “promovieren”, to indicate Doctorate/PhD.

The usage of synonyms in questionnaires may affect the performance of model's output because of the lower frequency of their occurrence. For instance, a questionnaire talking about the PhD students can use words such as “Doktorand”, “promotion”, and PhD for a similar concept, but since they occur fewer times when written differently, we might miss them out, especially with techniques such as TF-IDF.

To ensure that we do not miss out on words that mean the same or fall under the same class, we wanted to incorporate a domain-specific vocabulary into our setup. The main aim of such controlled vocabulary would be to make sure that synonyms can be treated equally and then analyzed to be chosen as relevant to the document or not. Furthermore, having a standardized set of keywords can also help in grouping the questionnaires having similar keywords together. This can act as an efficient way of extracting information by improving the index search.

In our thesis, this domain knowledge is incorporated by using known thesauri which are specific from domain, such as social science, economics, and sociology and these thesauri are known to be the standard vocabularies. Therefore, the set of keywords and keyphrases we get at the end after using the approaches ideally should be within the scope of these thesauri. For instance, if the questionnaire uses “Doktorand” and *promotion* for the same concept, the relevant keyword should be the one defined by the thesaurus.

In order to work with this, we have selected four thesauri which are from the social science domain and explanation of these thesauri is mentioned in Chapter 4.

5.5.1 Mapping function between KeyBERT and thesauri

As mentioned in Section 5.5, external resources such as thesauri and controlled vocabularies from social science domain are used to get the efficient keywords and keyphrases. In this thesis, we have compared the keywords and keyphrases extracted by KeyBert with the keywords which are within the thesauri. Afterwards, we have created a mapping function between them to see if the keywords and keyphrases extracted by the method(KeyBERT) best represent them in the thesauri, which in turn give us an impression of finding good keywords and keyphrases.

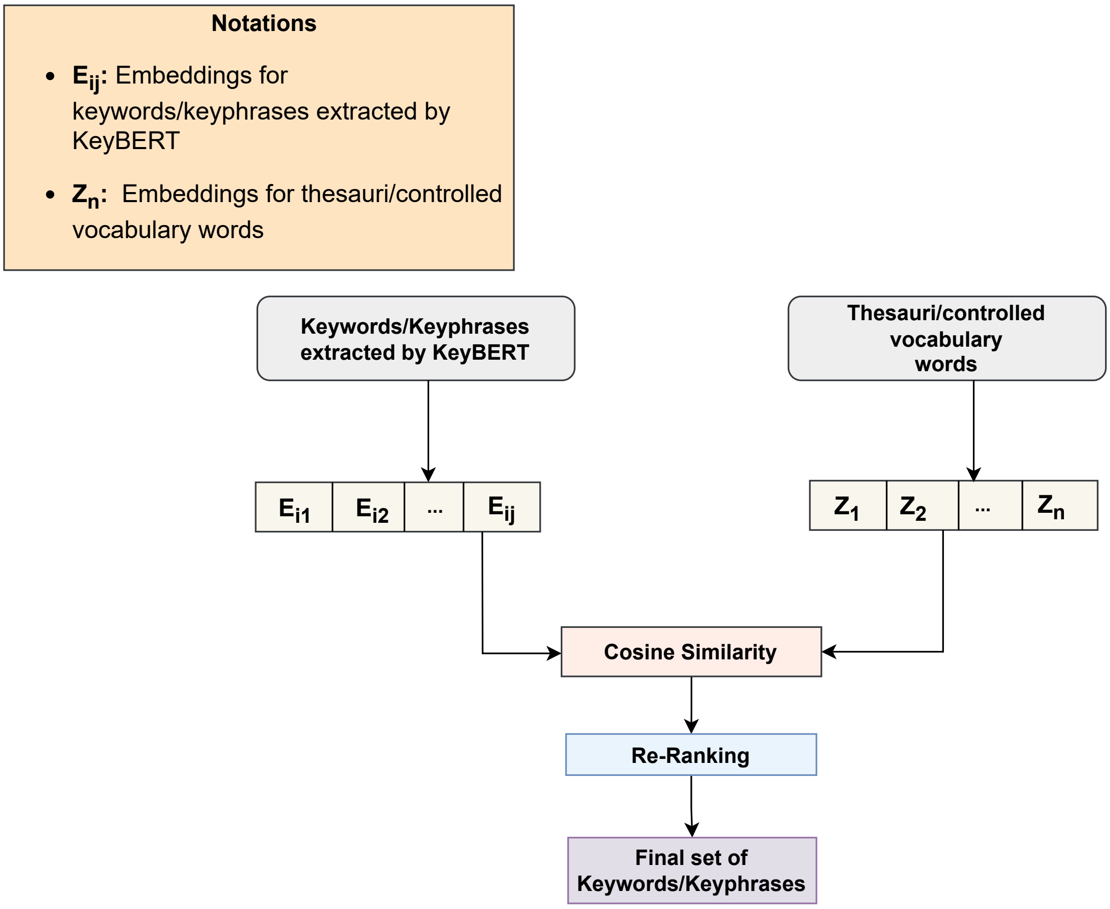

Notations
E_{ij}: Embeddings for keywords/keyphrases extracted by KeyBERT
Z_n: Embeddings for thesauri/controlled vocabulary words
Keywords/Keyphrases extracted by KeyBERT
Thesauri/controlled vocabulary words
E_{i1}
E_{i2}
...
E_{ij}
Z_1
Z_2
...
Z_n
Cosine Similarity
Re-Ranking
Final set of Keywords/Keyphrases

Figure 5.8: The figure shows the process of incorporating the thesauri and mapping the keywords/keyphrases within the thesauri to keywords/keyphrases extracted by KeyBERT.

Once the keywords/Keyphrases are extracted using KeyBert, in the next step, we create the embedding for these keywords/keyphrases. In Figure 5.8 E_{ij} depicts the embeddings for keywords/keyphrases extracted by KeyBERT. These embedding was created using the pre-trained model *symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli*. Similarly, we have calculated the embeddings for the words/phrases inside the thesauri denoted by Z_n. Finally, we compute the cosine similarities between these embeddings which tells to what extent a particular keyword/keyphrase extracted by keyBERT is similar to a word/phrase within the thesaurus.

After mapping, once we get all the keywords and keyphrases with their similarity scores, we have re-ranked keywords/keyphrases with respect to the similarity and relevance scores that we get for keywords/keyphrases extracted by keyBERT. Explanation of re-ranking phase is given in section 5.5.2.

In order to capture the semantics of keywords and keyphrases, we have incorporated word embeddings. Table 5.7 and Table 5.8 shows the results that we get after creating a mapping function between keywords/keyphrases extracted by KeyBERT and keywords/keyphrases that are within the thesaurus.

Keyword extracted by KeyBert	Keyword within thesaurus	Similarity value
beschäftigungsverhältnis	beschäftigungsverhältnis	1.0
promotions	promotion	0.91
hochschulzugangsberechtigung	hochschulabschluss	0.89
unterstützungsangebot	förderungsmaßnahme	0.71

Table 5.7: The table shows some example results after mapping. The first column shows the **keywords** extracted by KeyBERT, and the second column shows the **keywords** inside the thesaurus(Thesoz). Finally, the third column depicts the extent to which they are similar.

Keyphrases extracted by KeyBert	Keyphrases within thesaurus	Similarity value
berufliche stellung	berufliche stellung	1.0
berufliche qualifikation	berufliche qualifikationen	0.97
wissenschaftliche mitarbeiter	wissenschaftliches personal	0.94
wissenschaftlichen fortschritt	wissenschaftliche innovation	0.82

Table 5.8: The table shows some example results after mapping. The first column shows the **keyphrases** extracted by KeyBERT, and the second column shows the **keyphrases** inside the thesaurus(ELSST). Finally, the third column depicts the extent to which they are similar.

5.5.2 Re-Ranking

For experimentation purposes, we weighed keywords/keyphrases extracted by KeyBert (W_i) and the keywords/keyphrases we got after mapping (W_j). The motivation behind the same was that we wanted to see how the relevant keywords and keyphrases change by varying the weights (regularisation factors) given to W_i and W_j. There are three main scenarios to our approach, namely 1. Introducing the regularisation factor concerning only W_i. 2. Introducing the regularisation factor concerning only W_j and 3. Introducing the regularisation factors with respect to both W_i and W_j

Additionally, every scenario has been further divided into cases wherein the regularisation factor is tweaked to give higher importance to one of the variables W_i and W_j. The results are compared to the baseline setting, where no regularisation factor is considered. The details of the same can be seen as follows:

A) Baseline

The baseline was to consider both of them equal. To do that, we added the KeyBERT relevance score(R_s) to the cosine similarities values(SIM) that we calculated between keywords/keyphrase extracted KeyBERT and thesauri keywords/keyphrases and shown in Table 5.9 and 5.10.

$$\Big\{R_s + SIM \tag{5.1}$$

Where, R_s = Relevance scores of keywords/keyphrases extracted using KeyBERT

and, SIM = Similarity between thesaurus and KeyBERT keywords/keyphrases.

In table 5.9 the first column depicts the keywords extracted by KeyBERT where the first keyword which is *gesellschaftspolitisch* has a highest relevance score of 0.87 followed by *experiment* with a relevance score of 0.77 which is shown in third column. The second column shows the keywords that we get after creating a mapping function between keywords extracted by KeyBERT and keywords within thesauri Additionally, the fourth column depicts the similarity between them. Finally, after re-ranking process we get a new relevance score and based on which the keywords are ordered. These new relevance score can be seen in fifth column. The changes are highlighted with light grey color. For example: now, *experiment* has a highest score of 1.90 followed by *gesellschaftspolitisch* which has a score of 1.88.

Similarly, Table 5.10 shows re-ranking for keyphrases. It can be seen that *vereinbarkeit von promotion und familie* has a relevance score of 0.88 and for *finanzielle situation* it is 0.63. However, after re-ranking *finanzielle situation* has a higher score, i.e. 1.63 compared to *vereinbarkeit von promotion und familie*, which now has a score of 1.60.

Keywords extracted by KeyBert	Words within thesaurus	KeyBert Relevance score	Similarity value	Value after Re-Ranking
gesellschaftspolitisch	gesellschaftspolitik	0.87	0.89	1.88
experiment	experiment	0.77	1	1.90
promovieren	promotion	0.41	0.71	1.21
karriereplanung	karriereplanung	0.64	1	1.77
forschungsdatum	forschungsdokumentation	0.45	0.60	1.14

Table 5.9: The table shows the baseline assumption for re-ranking keywords. The first column depicts the keywords extracted by the KeyBERT algorithm, and then the second column shows the keywords within the thesauri. The third column depicts the relevance score for each keywords that KeyBERT extracts. The fourth column shows the similarity score between keywords extracted by KeyBERT and keywords within thesauri. Similarity value of 1 means the keywords are perfectly matched. Finally, the last column depicts the score we get after re-ranking.

Keyphrases extracted by KeyBert	Phrases within thesaurus	KeyBert Relevance score	Similarity value	Value after Re-Ranking
finanzielle situation	finanzielle situation	0.63	1	1.63
wissenschaftliche mitarbeiter	wissenschaftlicher mitarbeiter	0.62	0.97	1.59
vereinbarkeit von promotion und familie	vereinbarkeit von familie und beruf	0.88	0.72	1.60
gute wissenschaftliche praxis	wissenschaftliche erkenntnis	0.44	0.72	1.16
tätigkeit in einem etablierten betrieb unternehmen	dienstleistung für unternehmen	0.94	0.74	1.68

Table 5.10: The table shows the baseline assumption for re-ranking keyphrases.

B) Introducing the regularization factor(λ) with respect to SIM

Another re-ranking scenario includes changing the thesaurus, and KeyBERT will be as it is. To give more importance to keywords/keyphrases extracted by KeyBERT, their relevance score is multiplied with standard deviation. We are doing this because a value(or keyword/keyphrase with a relevance score) needs to be multiplied by a more significant number if we want to give it a greater weight.

The cosine similarity value we calculate between keyBert keywords/Keyphrase and Thesaurus keywords/Keyphrase will always be less than or equal to 1.

For example: If we have a number, let us say 0.8, and if we multiply it by 0.6, the original number will be reduced because it is a decimal number. To conclude, we multiply the similarity scores with the standard deviation to give higher importance to either of them. When we want to give lower importance, we multiply with the average.

B.1) Giving more weightage to keywords/keyphrases within thesauri:

Here we have taken the standard deviation of the similarity values we get after mapping is performed between KeyBert keywords/keyphrases and thesaurus keywords/keyphrases. By doing so, we are giving more importance to thesaurus keywords/keyphrases.

For example, as shown in Table 5.11, the keyword *gesellschaftspolitik*, which KeyBERT extracts, has the highest relevance score of 0.87 followed by *experiment* with a relevance score of 0.77. However, after mapping and re-ranking process *experiment* has a higher score, which is 1.90 followed by *gesellschaftspolitik*, which has a slightly lesser score which is 1.88. Similarly, Table 5.12 shows re-ranking for keyphrases. For instance, *vereinbarkeit von promotion und familie* is a keyphrase extracted by KeyBERT with relevance score of 0.88 followed by *wissenschaftliche mitarbeiter* which has a relevance score of 0.62. However, after mapping and re-ranking process *wissenschaftliche mitarbeiter* has higher score which is 1.72 compared to *vereinbarkeit von promotion und familie* which has a score of 1.69.

$$\begin{cases} R_s + \lambda * SIM \\ \text{where, } \lambda > 1 \end{cases} \tag{5.2}$$

here, $\lambda = 1 +$ standard deviation(SIM)

and, SIM = Similarity between thesaurus and keyBert keywords/keyphrases.

Keywords extracted by KeyBert	words within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
gesellschaftspolitisch	gesellschaftspolitik	0.87	0.89	1.88
experiment	experiment	0.77	1	1.90
promovieren	promotion	0.41	0.71	1.12
karriereplanung	karriereplanung	0.64	1	1.77
forschungsdatum	forschungsdokumentation	0.45	0.60	1.14

Table 5.11: The table depicts how regularization factor(λ) is used for re-ranking keywords. Here, more importance is given to to thesaurus keywords by considering standard deviation.

Keyphrases extracted by KeyBert	Phrases within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
finanzielle situation	finanzielle situation	0.63	1	1.76
wissenschaftliche mitarbeiter	wissenschaftlicher mitarbeiter	0.62	0.97	1.72
vereinbarkeit von promotion und familie	vereinbarkeit von familie und beruf	0.88	0.72	1.69
gute wissenschaftliche praxis	wissenschaftliche erkenntnis	0.44	0.72	1.26
tätigkeit in einem etablierten betrieb unternehmen	dienstleistung für unternehmen	0.94	0.74	1.78

Table 5.12: The table depicts how regularization factor(λ) is used for re-ranking keyphrases. Here, more importance is given to to thesaurus keyphrases by considering standard deviation.

B.2) Giving more weightage to keywords/keyphrases extracted by keyBERT

Here, we have considered the average of SIM, which is the similarity values we get after mapping is performed between KeyBert keywords/keyphrases and thesaurus keywords/keyphrases. Doing this gives more importance to keywords/keyphrases extracted by KeyBERT. For example: Table 5.13 and 5.14 shows the difference in scores after re-ranking.

$$\begin{cases} R_s + \lambda * SIM \\ \text{where, } \lambda < 1 \end{cases} \tag{5.3}$$

Here, $\lambda = \text{avg}(SIM)$

and, SIM = Similarity between thesaurus and keyBert keywords/keyphrases.

Keywords extracted by KeyBert	words within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
gesellschaftspolitisch	gesellschaftspolitik	0.87	0.89	1.60
experiment	experiment	0.77	1	1.59
promovieren	promotion	0.41	0.71	0.99
karriereplanung	karriereplanung	0.64	1	1.46
forschungsdatum	forschungsdokumentation	0.45	0.60	0.95

Table 5.13: The table depicts how regularization factor(λ) is used for re-ranking keywords. Here, more importance is given to keywords extracted by KeyBERT by considering average of SIM.

Keyphrases extracted by KeyBert	Phrases within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
finanzielle situation	finanzielle situation	0.63	1	1.39
wissenschaftliche mitarbeiter	wissenschaftlicher mitarbeiter	0.62	0.97	1.36
vereinbarkeit von promotion und familie	vereinbarkeit von familie und beruf	0.88	0.72	1.42
gute wissenschaftliche praxis	wissenschaftliche erkenntnis	0.44	0.72	0.99
tätigkeit in einem etablierten betrieb unternehmen	dienstleistung für unternehmen	0.94	0.74	1.50

Table 5.14: The table depicts how regularization factor(λ) is used for re-ranking keyphrases. Here, more importance is given to keyphrases extracted by KeyBERT by considering average of SIM. .

C) Introducing the regularization factor(γ) with respect to R_s

In this re-ranking scenario, the change will affect the KeyBERT while the thesaurus will remain unchanged.

C.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

To give more weightage to keywords/keyphrases extracted by KeyBERT, standard deviation of R_s is performed.

For example: Table 5.15 shows the keyword *promovieren* has a KeyBERT relevance score of 0.41 and the keyword *forschungsdatum* has a KeyBERT relevance score of 0.45. However, after mapping and re-ranking process *promovieren* got a higher score of 1.20 while score for *forschungsdatum* reduced to 1.15. Similarly, Table 5.16 shows the difference in scores for keyphrases after re-ranking.

$$\begin{cases} \gamma * R_s + SIM \\ \text{where, } \gamma > 1 \end{cases} \tag{5.4}$$

Here, $\gamma = 1 + \text{standard deviation}(R_s)$

and ,R_s = Relevance score of keyBert keywords/keyphrases.

Keywords extracted by KeyBert	words within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
gesellschaftspolitisch	gesellschaftspolitik	0.87	0.89	1.94
experiment	experiment	0.77	1	1.93
promovieren	promotion	0.41	0.71	1.20
karriereplanung	karriereplanung	0.64	1	1.77
forschungsdatum	forschungsdokumentation	0.45	0.60	1.15

Table 5.15: The table depicts how regularization factor(γ) is used for re-ranking keywords. Here, more importance is given to keywords extracted by KeyBERT by considering standard deviation of R_s.

Keyphrases extracted by KeyBert	Phrases within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
finanzielle situation	finanzielle situation	0.63	1	1.79
wissenschaftliche mitarbeiter	wissenschaftlicher mitarbeiter	0.62	0.97	1.75
vereinbarkeit von promotion und familie	vereinbarkeit von familie und beruf	0.88	0.72	1.82
gute wissenschaftliche praxis	wissenschaftliche erkenntnis	0.44	0.72	1.28
tätigkeit in einem etablierten betrieb unternehmen	dienstleistung für unternehmen	0.94	0.74	1.93

Table 5.16: The table depicts how regularization factor(γ) is used for re-ranking keyphrases. Here, more importance is given to keyphrases extracted by KeyBERT by considering standard deviation of R_s.

C.2) Giving more weightage to keywords/keyphrases within thesauri

In this case we have considered taking average of R_s which gives more importance keywords/keyphrases which are within the thesauri.

For example: Table 5.17 shows the keyword *gesellschaftspolitisch* has a KeyBERT relevance score of 0.87 and the keyword *karriereplanung* has a KeyBERT relevance score of 0.64. However, after mapping and re-ranking process *karriereplanung* got a higher score of 1.14 while score for *gesellschaftspolitisch* reduced to 1.09. Similarly, Table 5.18 shows re-ranking for keyphrases. For instance, *vereinbarkeit von promotion und familie* is a keyphrase extracted by KeyBERT with relevance score of 0.88 and another keyphrase *finanzielle situation* which has a relevance score of 0.66. However, after mapping and re-ranking process *finanzielle situation* got higher score which is 1.25 compared to *vereinbarkeit von promotion und familie* which has a score of 1.08.

$$\begin{cases} \gamma * R_s + SIM \\ \text{where, } \gamma < 1 \end{cases} \tag{5.5}$$

Here, $\gamma = \text{avg}(R_s)$

and, R_s = Relevance score of keyBert keywords

Keywords extracted by KeyBert	words within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
gesellschaftspolitisch	gesellschaftspolitik	0.87	0.89	1.09
experiment	experiment	0.77	1	1.17
promovieren	promotion	0.41	0.71	0.80
karriereplanung	karriereplanung	0.64	1	1.14
forschungsdatum	forschungsdokumentation	0.45	0.60	0.71

Table 5.17: The table depicts how regularization factor(γ) is used for re-ranking keywords. Here, more importance is given to keywords which are within thesauri by considering standard average of R_s.

Keyphrases extracted by KeyBert	Phrases within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
finanzielle situation	finanzielle situation	0.63	1	1.25
wissenschaftliche mitarbeiter	wissenschaftlicher mitarbeiter	0.62	0.97	1.22
vereinbarkeit von promotion und familie	vereinbarkeit von familie und beruf	0.88	0.72	1.08
gute wissenschaftliche praxis	wissenschaftliche erkenntnis	0.44	0.72	0.90
tätigkeit in einem etablierten betrieb unternehmen	dienstleistung für unternehmen	0.94	0.74	1.13

Table 5.18: The table depicts how regularization factor(γ) is used for re-ranking keyphrases. Here, more importance is given to keyphrases which are within thesauri by considering standard average of R_s.

D) Introducing the regularization factors(λ and γ) with respect to R_s and SIM

In this re-ranking scenario, the effect will come for both i.e. keywords/keyphrases extracted by KeyBert and keywords/keyphrases that are withing the thesauri. Either more weightage will be given to both keyBERT and Thesaurus keywords/keyphrases or less weightage will be given to both KeyBERT and Thesaurus keywords/keyphrases.

D.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

In this case more importance is given to keyBERT keywords/keyphrases and to do this we have taken standard deviation of R_s and average of SIM.

For example: Table 5.19 shows the keyword *promovieren* has a KeyBERT relevance score of 0.41 and the keyword *forschungsdatum* has a KeyBERT relevance score of 0.45. However, after mapping and re-ranking process *promovieren* got a higher score of 1.07 while score for *forschungsdatum* reduced to 1.04. Similalry, Table 5.20 shows the difference in scores for keyphrases after re-ranking.

$$\begin{cases} \gamma * R_s + \lambda * SIM \\ \text{where, } \gamma > \lambda \end{cases} \tag{5.6}$$

Here, γ = 1 + standard deviation(R_s)

and, λ = avg(SIM)

Keywords extracted by KeyBert	words within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
gesellschaftspolitisch	gesellschaftspolitik	0.87	0.89	1.78
experiment	experiment	0.77	1	1.75
promovieren	promotion	0.41	0.71	1.07
karriereplanung	karriereplanung	0.64	1	1.60
forschungsdatum	forschungsdokumentation	0.45	0.60	1.04

Table 5.19: The table depicts how regularization factors(λ and γ) is used for re-ranking keywords. Here, more importance is given to keywords which are extracted By KeyBERT. Both standard deviation(R_s) and average(SIM) is considered.

Keyphrases extracted by KeyBert	Phrases within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
finanzielle situation	finanzielle situation	0.63	1	1.55
wissenschaftliche mitarbeiter	wissenschaftlicher mitarbeiter	0.62	0.97	1.52
vereinbarkeit von promotion und familie	vereinbarkeit von familie und beruf	0.88	0.72	1.65
gute wissenschaftliche praxis	wissenschaftliche erkenntnis	0.44	0.72	1.11
tätigkeit in einem etablierten betrieb unternehmen	dienstleistung für unternehmen	0.94	0.74	1.75

Table 5.20: The table depicts how regularization factors(λ and γ) is used for re-ranking keyphrases. Here, more importance is given to keyphrases which are extracted By KeyBERT. Both standard deviation(R_s) and average(SIM) is considered.

D.2) Giving more weightage to keywords/keyphrases within thesauri

In this case more importance is given to keywords/keyphrases which are within thesauri. In order to do that we take average of R_s and standard deviation of SIM.

For example: Table 5.21 shows the keyword *gesellschaftspolitisch* has a KeyBERT relevance score of 0.87 and the keyword *karriereplanung* has a KeyBERT relevance score of 0.64. However, after mapping and re-ranking process *karriereplanung* got a higher score of 1.27 while score for *gesellschaftspolitisch* reduced to 1.20. Similarly, Table 5.22 shows re-ranking for keyphrases. For instance, *vereinbarkeit von promotion und familie* is a keyphrase extracted by KeyBERT with relevance score of 0.88 and another keyphrase *wissenschaftliche mitarbeiter* which has a relevance score of 0.62. However, after mapping and re-ranking process *wissenschaftliche mitarbeiter* got higher score which is 1.35 compared to *vereinbarkeit von promotion und familie* which has a score of 1.17.

$$\begin{cases} \gamma * R_s + \lambda * SIM \\ \text{where, } \lambda > \gamma \end{cases} \tag{5.7}$$

Here, $\gamma = \text{avg}(R_s)$

and, $\lambda = 1 + \text{standard deviation(SIM)}$

Keywords extracted by KeyBert	words within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
gesellschaftspolitisch	gesellschaftspolitik	0.87	0.89	1.20
experiment	experiment	0.77	1	1.30
promovieren	promotion	0.41	0.71	0.89
karriereplanung	karriereplanung	0.64	1	1.27
forschungsdatum	forschungsdokumentation	0.45	0.60	0.78

Table 5.21: The table depicts how regularization factors(λ and γ) is used for re-ranking keywords. Here, more importance is given to keywords which are within thesauri. Both standard deviation(SIM) and average(R_s) is considered.

Keyphrases extracted by KeyBert	Phrases within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
finanzielle situation	finanzielle situation	0.63	1	1.39
wissenschaftliche mitarbeiter	wissenschaftlicher mitarbeiter	0.62	0.97	1.35
vereinbarkeit von promotion und familie	vereinbarkeit von familie und beruf	0.88	0.72	1.17
gute wissenschaftliche praxis	wissenschaftliche erkenntnis	0.44	0.72	1.0
tätigkeit in einem etablierten betrieb unternehmen	dienstleistung für unternehmen	0.94	0.74	1.22

Table 5.22: The table depicts how regularization factors(λ and γ) is used for re-ranking keyphrases. Here, more importance is given to keyphrases which are within thesauri. Both standard deviation(SIM) and average(R_s) is considered.

D.3) Giving less weightage to keywords/keyphrases extracted by keyBERT and within thesauri

In this case less weightage is given to both KeyBERT and thesaurus keywords/keyphrases and in order to do that we have taken average of both R_s and SIM.

For example: Table 5.23 shows the keyword *gesellschaftspolitisch* has a KeyBERT relevance score of 0.87 and the keyword *karriereplanung* has a KeyBERT relevance score of 0.64. However, after mapping and re-ranking process *karriereplanung* got a higher score of 0.97 while score for *gesellschaftspolitisch* reduced to 0.93.

Similarly, Table 5.24. shows re-ranking for keyphrases. For instance, *vereinbarkeit von promotion und familie* is a keyphrase extracted by KeyBERT with relevance score of 0.88 and another keyphrase *wissenschaftliche mitarbeiter* which has a relevance score of 0.62. However, after mapping and re-ranking process *wissenschaftliche mitarbeiter* got higher score which is 0.99 compared to *vereinbarkeit von promotion und familie* which has a score of 0.90.

$$\begin{cases}\gamma * R_s + \lambda * SIM\end{cases} \tag{5.8}$$

here, $\gamma = \text{avg}(R_s)$

and, $\lambda = \text{avg(SIM)}$

Keywords extracted by KeyBert	words within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
gesellschaftspolitisch	gesellschaftspolitik	0.87	0.89	0.93
experiment	experiment	0.77	1	0.99
promovieren	promotion	0.41	0.71	0.67
karriereplanung	karriereplanung	0.64	1	0.97
forschungsdatum	forschungsdokumentation	0.45	0.60	0.60

Table 5.23: The table depicts how regularization factors(λ and γ) is used for re-ranking keywords. Here, less weightage is given to keywords which are within thesauri as well as keywords extracted by KeyBERT. Average is considered for both R_s and SIM.

Keyphrases extracted by KeyBert	Phrases within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
finanzielle situation	finanzielle situation	0.63	1	1.0
wissenschaftliche mitarbeiter	wissenschaftlicher mitarbeiter	0.62	0.97	0.99
vereinbarkeit von promotion und familie	vereinbarkeit von familie und beruf	0.88	0.72	0.90
gute wissenschaftliche praxis	wissenschaftliche erkenntnis	0.44	0.72	0.73
tätigkeit in einem etablierten betrieb unternehmen	dienstleistung für unternehmen	0.94	0.74	0.95

Table 5.24: The table depicts how regularization factors(λ and γ) is used for re-ranking keyphrases. Here, less weightage is given to keyphrases which are within thesauri as well as keyphrases extracted by KeyBERT. Average is considered for both R_s and SIM.

D.4) Giving more weightage to keywords/keyphrases extracted by keyBERT and within thesauri

In this case more weightage is given to both KeyBERT and thesaurus keywords/keyphrases and to do so we have taken standard deviation of both R_s and SIM.

For example: Table 5.25 shows the keyword *promovieren* has a KeyBERT relevance score of 0.41 and the keyword *forschungsdatum* has a KeyBERT relevance score of 0.45. However, after mapping and re-ranking process *promovieren* got a higher score of 1.29 while score for *forschungsdatum* reduced to 1.23. Similarly, Table 5.26. shows re-ranking for keyphrases. For instance, *vereinbarkeit von promotion und familie* is a keyphrase extracted by KeyBERT with relevance score of 0.88 and another keyphrase *finanzielle situation* which has a relevance score of 0.63. However, after mapping and re-ranking process *wissenschaftliche mitarbeiter* and *vereinbarkeit von promotion und familie* have the same score which is 1.92.

$$\Big\{\gamma * R_s + \lambda * SIM \tag{5.9}$$

Here, $\gamma = 1 +$ standard deviation(R_s)

and, $\lambda = 1 +$ standard deviation(SIM)

Keywords extracted by KeyBert	words within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
gesellschaftspolitisch	gesellschaftspolitik	0.87	0.89	2.06
experiment	experiment	0.77	1	2.05
promovieren	promotion	0.41	0.71	1.29
karriereplanung	karriereplanung	0.64	1	1.90
forschungsdatum	forschungsdokumentation	0.45	0.60	1.23

Table 5.25: The table depicts how regularization factors(λ and γ) is used for re-ranking keyphrases. Here, more weightage is given to keywords which are within thesauri as well as keywords extracted by KeyBERT. standard deviation is considered for both R_s and SIM.

Keyphrases extracted by KeyBert	Phrases within thesaurus	KeyBert Relevance score	Similarity value	value after Re-Ranking
finanzielle situation	finanzielle situation	0.63	1	1.92
wissenschaftliche mitarbeiter	wissenschaftlicher mitarbeiter	0.62	0.97	1.88
vereinbarkeit von promotion und familie	vereinbarkeit von familie und beruf	0.88	0.72	1.92
gute wissenschaftliche praxis	wissenschaftliche erkenntnis	0.44	0.72	1.38
tätigkeit in einem etablierten betrieb unternehmen	dienstleistung für unternehmen	0.94	0.74	2.03

Table 5.26: The table depicts how regularization factors(λ and γ) is used for re-ranking keyphrases. Here, more weightage is given to keyphrases which are within thesauri as well as keyphrases extracted by KeyBERT. standard deviation is considered for both R_s and SIM.

5.6 Summary

In this chapter, we presented the methodology of this thesis work. We start by presenting the workflow (cf. Figure 5.1), where we provide a detailed explanation of each module. The first module which is *Text Extraction and Pre-Processing* explains how relevant text is extracted by parsing the questionnaires, which was represented in XML format. The second module is *Approaches*, in which we explain selected approaches and how we used them to extract keywords and keyphrases from the questionnaires. The first approach is TF-IDF which is a statistical method. The second approach is TextRank which is a graph-based method. Finally, the last approach we have used is KeyBERT which is an embedding based approach. The third module is *Qualitative Assessment*. Once an approach results in either a set of keywords or keyphrases, some qualitative analysis were performed, which includes manually looking at those keywords or keyphrases and finding some insights. The fourth module is *Fine-Tuning and Post-Processing*, where we explained about the POS tagger which helps in assigning each word to a part of speech category. There is also a detailed explanation of the sequential regex pattern that was created for extracting the keyphrases. The final module which was explained in this chapter is *Incorporation of Domain Knowledge*, which contains the information on how external resources such as thesauri and controlled vocabularies from the social science domain are used. This incorporation makes the process efficient for extracting keywords and keyphrases which in-turns improve the index search.

6. Implementation

This chapter provides an overview of implementation for the concept that is provided in Chapter 5. Furthermore, we will discuss the selection of hyper-parameters for each techniques. This chapter begins by describing a function for text extraction from XML file. In addition, we will see the implementation snippets for TF-IDF, TextRank and KeyBERT.

6.1 Text Extraction and Pre-Processing

Text Extraction

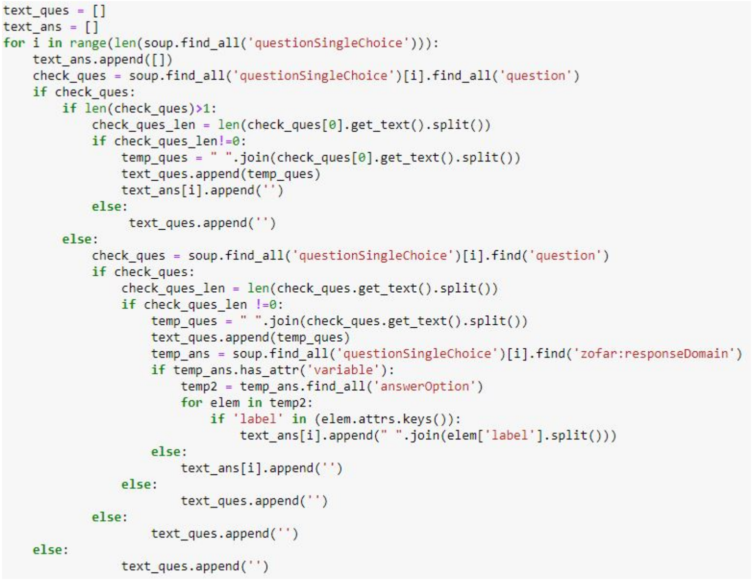
As mentioned in chapter 5, the information that was used in this study is comprised of questionnaires and was initially saved in XML format. It is converted to text first, and then identified the key information from it. Also mentioned in chapter 4, for each question and answer tag in the XML file We have created unique functions in order to extract the text in the required format. This is due to the fact that the XML design used various types of formatting for each tag. In 6.1, a python code is shown for extracting questions and answer options from xml tag `<zofar:questionSingleChoice>`. In order to parse the XML file python library which is Beautiful Soup [15] is used. Beautiful Soup is generally use to extract data from XML and HTML files.

Text Pre-Processing

After extracting the text from XML file, some basic pre-processing steps are performed to clean the data. For example, removing any HTML tags within text, any digit or single letter character in side the text is removed. Furthermore, the text is also and converted into lowercase for further experimentation. As discussed in section 5.1, there are other pre-processing techniques such as tokenization, stopword removal, and lemmatization is performed on our data. In order to tokenize the text, python nltk library is used [16]. For stopword removal also nltk library is used, from which

[15] https://fizzylogic.nl/2017/11/06/edit-jupyter-notebooks-over-ssh/

[16] https://www.nltk.org/api/nltk.tokenize.html



```
text_ques = []
text_ans = []
for i in range(len(soup.find_all('questionSingleChoice'))):
    text_ans.append([])
    check_ques = soup.find_all('questionSingleChoice')[i].find_all('question')
    if check_ques:
        if len(check_ques)>1:
            check_ques_len = len(check_ques[0].get_text().split())
            if check_ques_len!=0:
                temp_ques = " ".join(check_ques[0].get_text().split())
                text_ques.append(temp_ques)
                text_ans[i].append('')
            else:
                 text_ques.append('')
        else:
            check_ques = soup.find_all('questionSingleChoice')[i].find('question')
            if check_ques:
                check_ques_len = len(check_ques.get_text().split())
                if check_ques_len !=0:
                    temp_ques = " ".join(check_ques.get_text().split())
                    text_ques.append(temp_ques)
                    temp_ans = soup.find_all('questionSingleChoice')[i].find('zofar:responseDomain')
                    if temp_ans.has_attr('variable'):
                        temp2 = temp_ans.find_all('answerOption')
                        for elem in temp2:
                            if 'label' in (elem.attrs.keys()):
                                text_ans[i].append(" ".join(elem['label'].split()))
                    else:
                        text_ans[i].append('')
                else:
                        text_ques.append('')
            else:
                    text_ques.append('')
    else:
                text_ques.append('')
```

Figure 6.1: The figure shows the function created for `<zofar:questionSingleChoice>` for extarcting question and answer options.

German stopwords are imported. In order to work with the experimentation, the stopword list is extended and made a custom stopword list, as nltk has very less number of stowords within its list. This helped in removing the words in the text that offer no useful information for extracting keywords and keyphrases. Finally, for the lemmatization, the German spacy model pipeline is loaded (*de-core-news-lg*) [17] which comes with its lemmatizer. In addition, another lemmatizer is also used which is called *GermaLemma*. It helps in lemmatizing the POS tagged German language text [18].

6.2 Approaches

Once a cleaned set of text for each questionnaire is built, three different approaches which are TF-IDF, TextRank and KeyBERT is used in order to extract keywords and keyphrases(cf. Section 5.2).

1. **TF-IDF**

   The first step in using TF-IDF for extracting keywords and keyphrases is to create vocabularies. In order to do that, we have used a tools which is provided by scikit-learn library in python called as CountVectorizer [19]. CountVectrorizer helps in transforming the text into a vector based on the frequency of occurrence of each word in the text.

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(stop_words=stop_words,ngram_range = (1,1))
word_count_vector = cv.fit_transform(sample_data_w_lemma)
```

[17] https://spacy.io/models/

[18] https://github.com/WZBSocialScienceCenter/germalemma

[19] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

Here, cv.fit_transform build the vocabularies and output a term-document matrix which is also discussed in 5.2. In CountVectorizer, we can pass several parameters like stopwords which will not consider any stopwords while extracting keywords. ngram_range = (1,1) means that only unigram will be retrieved by the method. Alternatively, we can also pass n_gram range as (1,2) which is for bi-gram, (1.3) is for trigram.

In order to extract keyphrases, instead of CountVectorizer we have used KeyphraseCountVectorizer [20]. This package is built on top of CountVectorizer and TfidfVectorizer of scikit-learn. This package extract keyphrases from text documents using part-of-speech tags to create document and keyphrase matrix rather than n-gram tokens of a specific range.

The sequential regex pattern which is explained in section 5.4, is passed into the KeyphraseCountVectorizer to extract keyphrases.

Once vocabularies are created the next step is to compute IDF values. Here the word_count_vector is the sparse matrix which is generated using CountVectorizer which in-turn help in computing the IDF after calling the tfidf_transformer.fit() function

```
from sklearn.feature_extraction.text import TfidfTransformer
tfidf_transformer=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer.fit(word_count_vector)
```

The next step is to compute the TF-IDF and retrieve the top keywords with their relevance score. For the implementation of TF-IDF for our use case we have taken the reference from this blog [21].

2. **TextRank**

   As mentioned in Chapter 2, TextRank uses a graph representation of text as the basis for extracting keywords and keyphrases. In this implementation, we have considered POS tag, Noun, Adjectives and Verb from the text which then added as vertex to the graph. In order to build a cleaned corpus, Text pre-processing steps are applied to the text. In order to implement the TextRank for our use case we have used PyTextRank library. PyTextRank [22] is implementation of TextRank in python which is built on top of psacy pipeline extension [23]. This implementation will result in a lemma graph from the text after that we extract the top-ranked keywords and keyphrases. An example of a lemma graph from the raw is mentioned in chapter 5.

3. **KeyBERT**

   Instead of handling a full questionnaire as a single document, it is separated into numerous smaller ones, each of which comprises a question and answer pair to implement the KeyBERT method for our use case. Similar to TF-IDF and TextRank, pre-processing steps are applied to the corpus for cleaning. In order to extract keywords from the questionnaires we have used KeyBERT library [24]

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

[21] https://kavita-ganesan.com/python-keyword-extraction/

[22] https://derwen.ai/docs/ptr/

[23] https://spacy.io/universe/project/spacy-pytextrank

[24] https://maartengr.github.io/KeyBERT/index.html

which used uses BERT embeddings to retrive keywords and keyphrases which are most similar to a document.

```
from keybert import KeyBERT
kw_model = KeyBERT("symanto/sn-xlm-roberta-base-snli-mnli-anli-
                                    xnli")
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(ngram_range=(1, 1), stop_words=
                                    stop_words)
keywords = kw_model.extract_keywords(corpus)
```

Here, we first import the KeyBERT library and then we pass a pre-trained sentence-transformers model which helps in creating the embedding for the question-answer pairs. Furthermore, CountVectorizer is used and we can pass several parameters like stopwords which will not consider any stopwords while extracting keywords. Then we use vectorizer.fit() and provide our corpus to this function to build the vocabularies. Finally, in extract_keywords() function we pass the corpus as well some other parameters namely *use_mmr*, *diversity* for diversification of results and *top_n* for retrieving the top_n keywords that we have selected. The sequential regex pattern that we have explained in section 5.4, is passed into the KeyphraseCountVectorizer to extract keyphrases.

6.3 Incorporation of Domain Knowledge

In this thesis, domain knowledge is incorporated by using the external resources such as thesauri, controlled vocabularies which are specific from domain, such as social science, economics, and sociology and these thesauri are known to be the standard vocabularies. In order to implement this we created a mapping function between keywords/keyphrases extracted by KeyBERT and keywords/keyphrases within the thesauri.

1. First, keywords and keyphrases are extracted using KeyBERT as mentioned above. After that we create the embedding for each keyword and keyphrases within the extracted set.

```
from sentence_transformers import SentenceTransformer
model = SentenceTransformer('symanto/sn-xlm-roberta-base-snli-
                                    mnli-anli-xnli')
keyword_embeddings = []
#Our sentences we like to encode
for i in range(len(words)):
    print(i,flush=True)
    keyword_embeddings.append([])
    for j in range(len(words[i])):
        #Sentences are encoded by calling model.encode()
        keyword_embeddings[i].append(model.encode(words[i][j]))
```

2. keywords and keyphrases are parsed from selected thesauri. For example: Thesoz thesaurus(cf. section 4.4. After that we create the embedding for each keyword and keyphrases within the thesaurus.

```
from sentence_transformers import SentenceTransformer
model = SentenceTransformer('symanto/sn-xlm-roberta-base-snli-
                                  mnli-anli-xnli')
thesauri_keyword_embeddings = []
#Our sentences we like to encode
for i in range(len(new_thesaurus_list)):
    print(i,flush=True)
    #Sentences are encoded by calling model.encode()
    thesauri_keyword_embeddings.append(model.encode(
                                      new_thesaurus_list[i]))
```

3. Finally, a mapping function is created between keybert keywords/keyphrase and thesaurus keywords/keyphrases. This mapping compute the cosine similarities between these embeddings which tells to what extent a particular keyword/keyphrase extracted by keyBERT is similar to a keywords/keyphrases within the thesaurus.

```
cosine_scores.append(util.cos_sim(keyword_embeddings[i],
                                  thesauri_keyword_embeddings[j]
                                  ))
```

In the end, when similarity scores is obtained for each keywords and keyphrases, we have re-ranked keywords/keyphrases with respect to the similarity and relevance scores that we get for keywords/keyphrases extracted by keyBERT. Further explanation of re-ranking phase is given in section 5.5.2.

6.4 Summary

In this chapter, we have provided the implementation detail for each module of the workflow (cf. Figure5.1). First, we provided the detail on *Text Extraction and Pre-processing*. Here a code snippet is shown to explain the extraction of text from the XML file. In addition, we have also explained the text pre-processing steps that were necessary for further experimentation. The emphasis was given to some pre-peocessing steps like tokenization, stopword removal and lemmatization. We also mention the different lemmatizer that was used for the lemmatization purposes. After text extraction and cleaning, further explanation is made on the implementation of approaches. Firstly, we explain how *TF-IDF* is used for extracting keywords and keyphrases with the help of code snippets. Secondly, for *TextRank* we have summarized which package have been used for the implementation. Similarly, For *KeyBERT* an explanation is made with the help of code snippets on its library and the sentence transformer model that we have used. Finally, for *Incorporation of Domain Knowledge*, with the code snippet, we have explained how embeddings are calculated for both KeyBERT keywords/keyphrases and Thesaurus keywords and keyphrases. In addition, a description is made on the mapping function which is created between keyBERT keywords/keyphrase and thesaurus keywords/keyphrases. Finally, with the help of a code snippet, an explanation on how the cosine similarities is calculated between them.

7. Evaluation

In the previous chapters, we mentioned all the modules of our framework (cf. Figure 5.1) except *Evaluation*. In this chapter, we will elaborate on the evaluation incorporated in this thesis. This chapter begins by describing the experimental setup, which includes the tools and components which were necessary to achieve the results. Finally, the evaluation of all the experiments will be elaborated.

7.1 Experimental Setup

Most of the experiment setup was run on the server access given by the OVGU DBSE working group. Several experiments were conducted into Jupyter Notebook [25] which was accessed remotely over SSH by setting up an SSH tunnel [26]. Post the extraction process described in Section 5 and 6, we wanted to evaluate the correctness of the extracted keywords/keyphrases. Our setup was a supervised setup where we had the annotated keywords/keyphrases by the researchers for the surveys. Our evaluation strategy aimed at seeing how many of them we captured. Furthermore, we also focus on the time taken to capture the same. The metrics used to achieve the same are described as follows.

Evaluation Metrics:

Evaluation metrics are classified into statistics-based and linguistics based on the work of Sun et al. [2020]. In order to evaluate our experiments, we have made use of statistics-based evaluation metrics. The evaluation metrics, which are based on statistics, examine a technique's effectiveness by measuring the percentage of the number of different keywords and keyphrases. The proportion can be the total number of extracted keywords and keyphrases, accurate keywords and keyphrases, hand-annotated keywords and keyphrases and finally, the incorrect keywords and keyphrases. The most common statistics-based evaluation metrics include:

[25] https://jupyter.org/

[26] https://fizzylogic.nl/2017/11/06/edit-jupyter-notebooks-over-ssh/

1. **Precision**: Precision is the proportion of retrieved keywords that are relevant. In the case of keywords/keyphrases extraction, it defines how precisely a method can extract the relevant keywords and keyphrases. For example: If we are searching for a keyword/keyphrase related to "law" then our method should give precise results on top like *lawyer*, *court*, *free legal advice* instead of all the keywords and keyphrases.

$$Precision = \frac{\text{Retrieved} \cap \text{Relevant}}{\text{Retrieved}} \tag{7.1}$$

2. **Recall**: Recall, on the other hand, is the proportion of relevant keywords that are retrieved. Therefore, keywords/keyphrase extraction refers to correctly extracting keywords/keyphrases out of all the keywords/keyphrases from the ground truth. For example: if we are searching for a document that talks about postdocs, universities and science policy, in this case, instead of preciseness, we want a recall, so in this case, our system should retrieve all the keywords related to post-graduation, PhD students, and science. Search engines primarily work on recall.

$$Recall = \frac{\text{Retrieved} \cap \text{Relevant}}{\text{Relevant}} \tag{7.2}$$

3. **F-measure**: The value of precision and recall lies between 0 and 1. In order to have a trade-off between precision and recall, F-measure is used. F-measure considers both precision and recall and calculates the harmonic mean of precision and recall, which is also known as F$_\beta$ measure. Like precision and recall value of F$_\beta$ lies between 0 and 1, where 1 is the best and 0 is the worst. The β is assigned some weights based on different use cases. For example, if we want to give more importance to precision and less weightage to recall, then we assign a value of 0.5 to β. Whereas, if a value such as 2 is given to β, it provides less weightage to precision and more weightage to recall.

$$F_\beta = (1+\beta^2) * \frac{\text{Precision * Recall}}{(\ \beta^2 \text{ * Precision}) + \text{Recall}} \tag{7.3}$$

In this thesis, we have considered the value of β as 1, which is a harmonic mean of both precision and recall and gives equal weightage to both and is known as F_1 score.

$$F_1 = 2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}} \tag{7.4}$$

We evaluated our methods using all three of the metrics. The precision was used to see how precise the approaches were in extracting the keywords. The recall focuses on if all the keywords marked by the researchers are extracted and the F-measure balances both criteria. We talk about it in detail in the following subsection.

7.2 Evaluation results for keyword extraction

As mentioned in Chapter 4, for each questionnaires, some ground truth keywords are provided to us for experimentation. In order to evaluate our experiments, we compared the keywords extracted by methods with the ground truth, which are manually labelled keywords.

To evaluate our system, we have used Precision, Recall and F_1 measure. Additionally, since we are never going to extract the keywords/keyphrases which are not present in the text it would wrongly give the impression of not being able to extract the keywords/keyphrases via the low precision-recall values. So for our analysis, we have only considered ground truth keywords/keyphrases which are present in the questionnaires itself. There are total 19 questionnaires that are available to us for extracting keywords and keyphrases. Out of those 19 questionnaires, there are 10 questionnaires for which we have ground truth keywords which are within text(cf. Table 7.1).

Questionnaires	Hand annotated keywords (within questionnaires)
Nacaps	wissenschaftssystem, betreuung, promovierende, promotion, finanzierung, mobilität, gesundheit
WeGe_W2	geflüchtete, studienkolleg, studienvorbereitung
StuMa2020	masterstudium
Studierendensurvey2016	evaluation, qualifikation, studiensituation
Promopanel_W2	promotion, arbeitsbedingungen, weiterbildung, auslandsaufenthalt
Promopanel_W3	promotion, arbeitsbedingungen, weiterbildung, auslandsaufenthalt
Promopanel_W4	promotion, arbeitsbedingungen, weiterbildung, auslandsaufenthalt
Promopanel_W5	promotion, arbeitsbedingungen, weiterbildung, auslandsaufenthalt, gesundheit
WeGe_W3	geflüchtete, studienkolleg, studienvorbereitung
sid_corona	digitale lehre, finanzielle situation

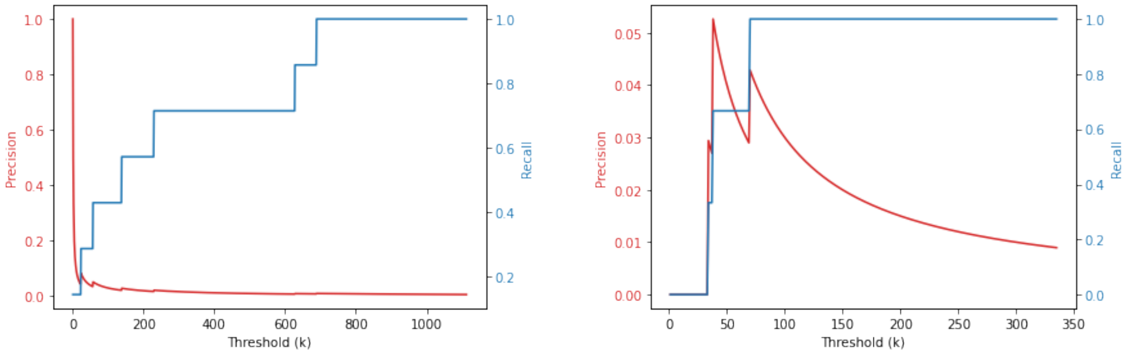
Table 7.1: The table shows the ground truth keywords which are manually annotated by scientists.

In this thesis instead of taking the problem of keywords and keyphrase extraction as a classification problem we have considered it as ranking problem. Based on this, we have evaluated the keywords and keyphrases extracted based on top-k, where k is the number of keywords/keyphrases extracted by a method. We calculate precision@k, recall@k and F_1@k

Precision@k computes the proportion of top-k keywords/keyphrases which are relevant. Recall@k indicates the proportion of relevant keywords/keyphrases which are recovered in the top-k Koboyatshwene et al. [2017]. The reason for this is because, for example, if a method extracts 500 keywords/keyphrases, from a researcher's point of view, all the keywords/keyphrases extracted by methods are not relevant. The top-k will vary with each document because the length of each document is different (cf. Chapter 4). This helps us in the best top-k for each document. Based on that, we find recall@1, which is the first keyword and recall@2, which means the first and second keyword and like this we vary the k between 1 and the number of keywords/keyphrases(N) extracted by the given method (k = [1,...., N]).

Once we get the result, we visualize it graphically and check where we get the best precision, recall and F_1 score. For example, Figure 7.1a shows a precision-recall graph for keywords extracted by TF-IDF for *nacpas* questionnaire. For this questionnaire, TF-IDF has extracted 1112 keywords. These keywords are extracted based on some order with their importance score. It can be seen from the Figure 7.1a that with every increasing k, the precision is decreasing. The highest precision we get for this is when k is 1, and then precision starts decreasing. However, the recall at this point is the lowest, with a value 0.14, after this recall began increasing. In our thesis, we have given more importance to recall compared to precision, this is because from the point of view of researchers, all the ground truth keywords must be extracted rather than how fast they are extracted. Based on that, we checked recall for each k, and at $k = 688$, recall gets saturated with a value of 1, which is the best recall value for this questionnaire. This also means for this questionnaires all the keywords which are in the ground truth have been retrieved by TF-IDF.

Similarly, Figure 7.1b depicts the graph for questionnaire *WeGe_w3*. There are 335 keywords which are extracted by TF-IDF. In this case, at $k = 33$, we started seeing the fluctuation in precision and recall curve. The highest precision that we get is 0.052 and when $k = 37$ and recall at this point is 0.66. At $k = 67$, we get the highest recall value which 1.0 and after this it gets saturated. This also means for this questionnaires all the keywords which are in the ground truth have been retrieved by TF-IDF.



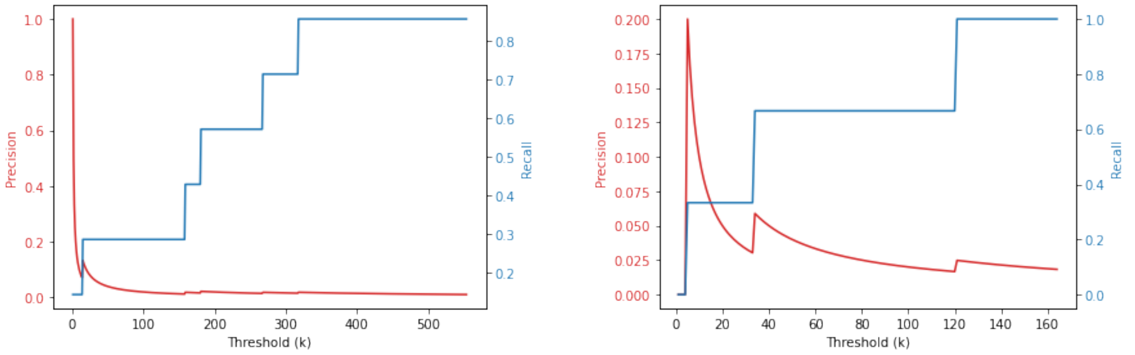
(a) This sub-figure shows a precision-recall graph for the nacaps questionnaire

(b) This sub-figure shows a precision-recall for the WeGe_W3 questionnaire

Figure 7.1: The figure depicts the precision-recall graphs for keywords extracted by TF-IDF. The x-axis represents threshold(k) which emphasizes the number of keywords extracted and also based on this, we decide the top-k. Y axis consists of two secondary axis, the axis on the right represents recall(blue curve in the graph) , and the axis on the left represents precision(red curve in the graph).

Furthermore, Figure 7.2a represents the precision-recall graph for keywords extracted by TextRank method for *nacaps* questionnaire. There are total 554 keywords which are extracted by the method. The best precision value which is 1.0 is obtained when $k = 1$ and after with increasing value of k it started decreasing and recall started increasing. At $k = 317$, the highest value of recall is 0.85 and after this it gets saturated.

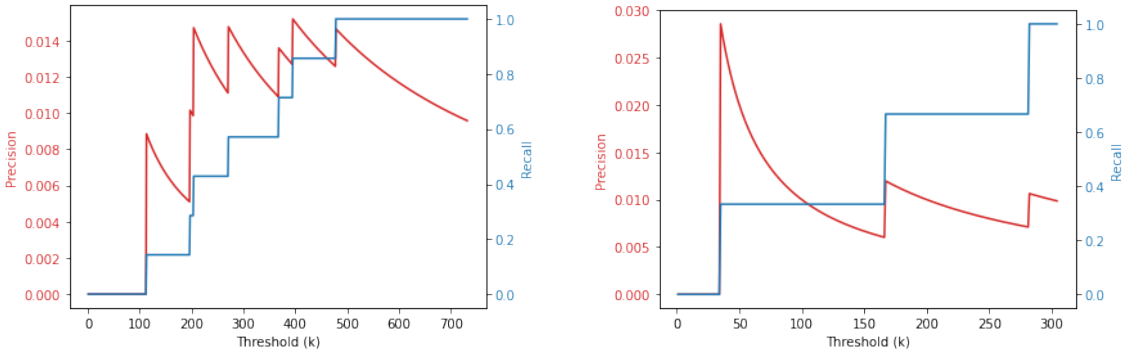
Similarly, 7.2b depicts the precision-recall graph for *WeGe_w3* questionnaire. There are total 164 keywords which are extracted by TextRank for this questionnaire. In this case, the best precision is obtained when $k = 4$. The best recall value(1.0) is



(a) This sub-figure shows a precision-recall graph for the nacaps questionnaire

(b) This sub-figure shows a precision-recall for the WeGe_W3 questionnaire

Figure 7.2: The figure depicts the precision-recall graphs for keywords extracted by TextRank



(a) This sub-figure shows a precision-recall graph for the nacaps questionnaire

(b) This sub-figure shows a precision-recall for the WeGe_W3 questionnaire

Figure 7.3: The figure depicts the precision-recall graphs for keywords extracted by KeyBERT

obtained when $k = 120$, and after that it gets saturated. This also means that for this questionnaires all the keywords which are in the ground truth have been retrieved by TextRank.

Figure 7.3a depicts the precision-recall graph for keywords extracted by KeyBERT method for *nacaps* questionnaire. There are total 731 keywords which are extracted for this questionnaire. The best precision value which is 0.015 is obtained when $k = 394$. The best recall value(1.0) is obtained at $k = 477$. This also means that for this questionnaires all the keywords which are in the ground truth have been retrieved by KeyBERT.

Finally, 7.3b shows the precision-recall graph for *WeGe_w3* questionnaire. There are total 304 keywords which are extracted by KeyBERT. In this case, the best precision is obtained when $k = 34$. The best recall value(1.0) is obtained when $k = 281$, and after that it gets saturated. This also means that for this questionnaires all the keywords which are in the ground truth have been retrieved by KeyBERT.

In table 7.2, 7.3 and 7.4, evaluation results for keyword extraction is demonstrated based on top-k recall, precision and F_1-score for TF-IDF, TextRank and KeyBERT.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	688	1	1
WeGe_W2	169	57	57
StuMa2020	1	1	1
Studierendensurvey2016	1227	93	93
Promopanel_W2	33	4	4
Promopanel_W3	35	1	1
Promopanel_W4	274	14	14
Promopanel_W5	313	2	2
WeGe_W3	69	37	37
sid_corona	401	2	2

Table 7.2: The table shows evaluation results for keywords extracted by TF-IDF.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	317	1	1
WeGe_W2	44	22	22
StuMa2020	3	3	3
Studierendensurvey2016	532	226	226
Promopanel_W2	172	5	5
Promopanel_W3	58	15	15
Promopanel_W4	124	124	124
Promopanel_W5	140	18	18
WeGe_W3	120	18	18
sid_corona	262	3	3

Table 7.3: The table shows evaluation results for keywords extracted by TextRank.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	477	394	394
WeGe_W2	313	34	34
StuMa2020	62	62	62
Studierendensurvey2016	811	215	215
Promopanel_W2	271	271	271
Promopanel_W3	200	200	200
Promopanel_W4	225	225	225
Promopanel_W5	245	245	245
WeGe_W3	281	34	34
sid_corona	257	257	257

Table 7.4: The table shows evaluation results for keywords extracted by KeyBERT.

Depending upon what is important for the researcher, in the end we can finalize the top-k for our approaches. For instance, if the priority of the researchers is to have all the keywords annotated by them as a output of the approach we would recommend extracting top-k keywords based on the recall. On the other hand, if a researcher would like to ensure the preciseness of the keywords outputted we would recommend top-k based on precision. Alternatively, researchers could also have an option to balance both the metrics.

For example: in the questionnaires *WeGe_W3*, we see that three keywords which were annotated by researchers mainly *geflüchtete*, *studienkolleg* and *studienvorbereitung* (cf. Table 7.1). If researchers considers recall as an ideal measure and the approach used is TextRank, one would have to extract 120 top keywords to ensure all keywords from ground truth are included (Recall =1). On the other hand, if the researchers decides to go with precision then one would have to extract 18 top keywords with the precision value being 0.20(cf. Figure 7.2b).

Furthermore, we also see that if we chose based on recall it gets computationally expensive as we have to extract 120 keywords in order to extract all the hand annotated keywords. On the other hand, if we chose on precision there are chances to miss out the majority of the keywords. In order to cater to this problem, we can additionally recommend k that tries to balance both precision and recall. In figure 7.2b a precision-recall graph for *WeGe_W3* questionnaire is shown and we can we see that at k = 40 we have a recall of 0.6 which in our case 2 out of 3 desired keywords have been extracted. Also, the precision value is approximately 0.075 which is still greater then 0.025@k = 120. Computationally, we would speed up since we only have to extract 40 out of 120 keywords. This could be an alternate solution if desired by the researcher. In a nutshell, we can say that our approach is flexible according to the needs of the researchers that is, our framework can be tuned if needed to include different scenarios.

7.3 Evaluation results for keyphrase extraction

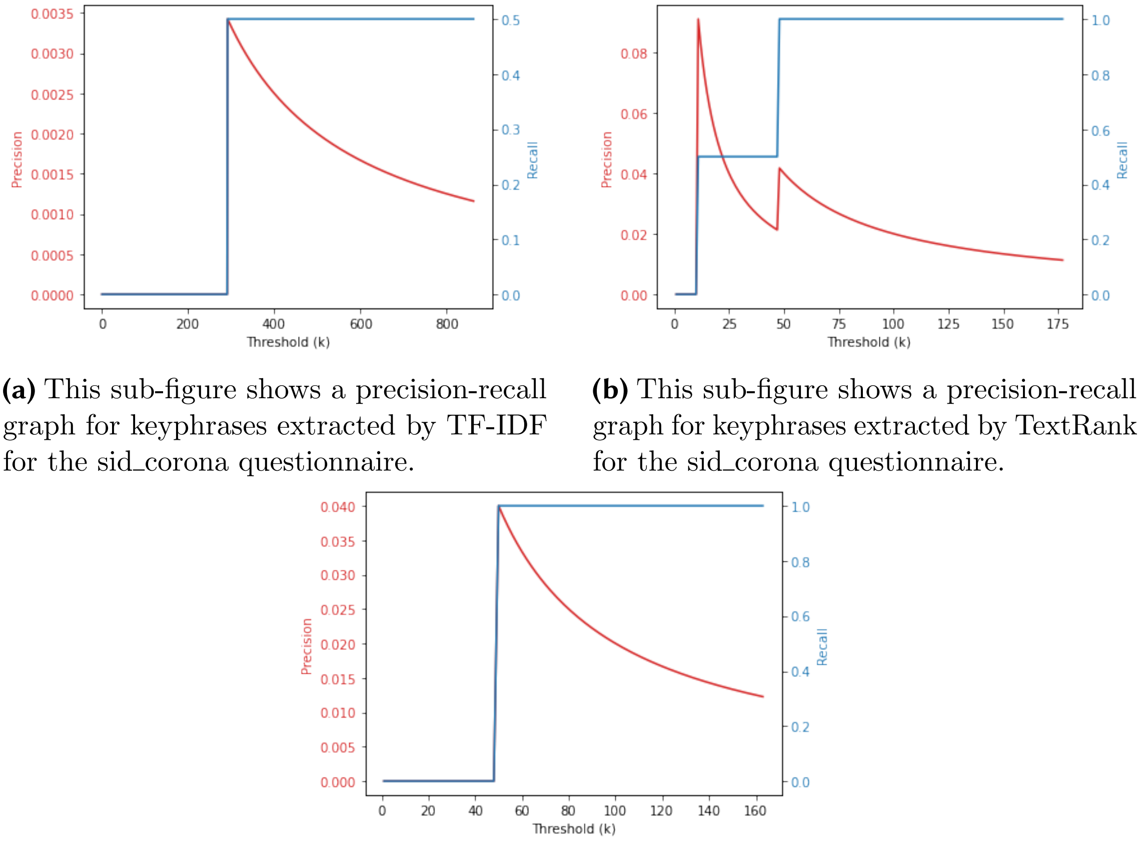
In order to evaluate keywords extracted by methods, we use precision, recall and F_1 measure. Similarly, for evaluating keyphrases, we used the same metrics and compared the ground keyphrases within the questionnaires with keyphrases extracted by a particular method.

Similar to keyword extraction, we have evaluated keyphrases based on top-k, where k is the number of keyphrases extracted by a method. Based on that, we compute precision@k and recall@k. As mentioned in Section 4, there are total 19 questionnaires that are available to us for extracting keywords and keyphrases. Out of those 19 questionnaires, there are only five questionnaires for which we have ground truth keyphrases which are within text (cf. Table 7.5).

Questionnaires	Hand annotated keyphrases (within questionnaires)
Nacaps	wissenschaftliche karriere, vereinbarkeit von familie und beruf
Promopanel_W3	wissenschaftliche aktivitäten
Promopanel_W4	wissenschaftliche aktivitäten
Promopanel_W5	wissenschaftliche aktivitäten
sid_corona	digitale lehre, finanzielle situation

Table 7.5: The table shows the ground truth keyphrases which are manually annotated by scientists.

To start with *sid_corona* questionnaire, we plotted a precision-recall graph for keyphrases extracted by TF-IDF (cf. Figure 7.4a). There are total 862 keyphrases



(a) This sub-figure shows a precision-recall graph for keyphrases extracted by TF-IDF for the sid_corona questionnaire.

(b) This sub-figure shows a precision-recall graph for keyphrases extracted by TextRank for the sid_corona questionnaire.

(c) This sub-figure shows a precision-recall graph for keyphrases extracted by KeyBERT for the sid_corona questionnaire.

Figure 7.4: The figure depicts the precision-recall graphs for keyphrases extracted by TF-IDF, TextRank and KeyBERT respectively.

which are extracted for this questionnaires. The best precision value in this case is 0.0034 and is obtained at k =291. The best recall value in this case is 0.5 and and obtained at k = 291. In this case, out of 2 keyphrases in the ground truth, only one keyphrase is extracted by TF-IDF.

Furthermore, in Figure 7.4b a precision-recall graph for is shown for keyphrases extracted by TextRank for *sid_corona* questionnaire. There are total 177 keyphrases which are extracted for this questionnaires. The best precision value in this case is 0.09 and is obtained at k =10. The best recall value in this case is 1.0 and and obtained at k = 47. This also means that for this questionnaires all the keyphrases which are in the ground truth have been retrieved by TextRank.

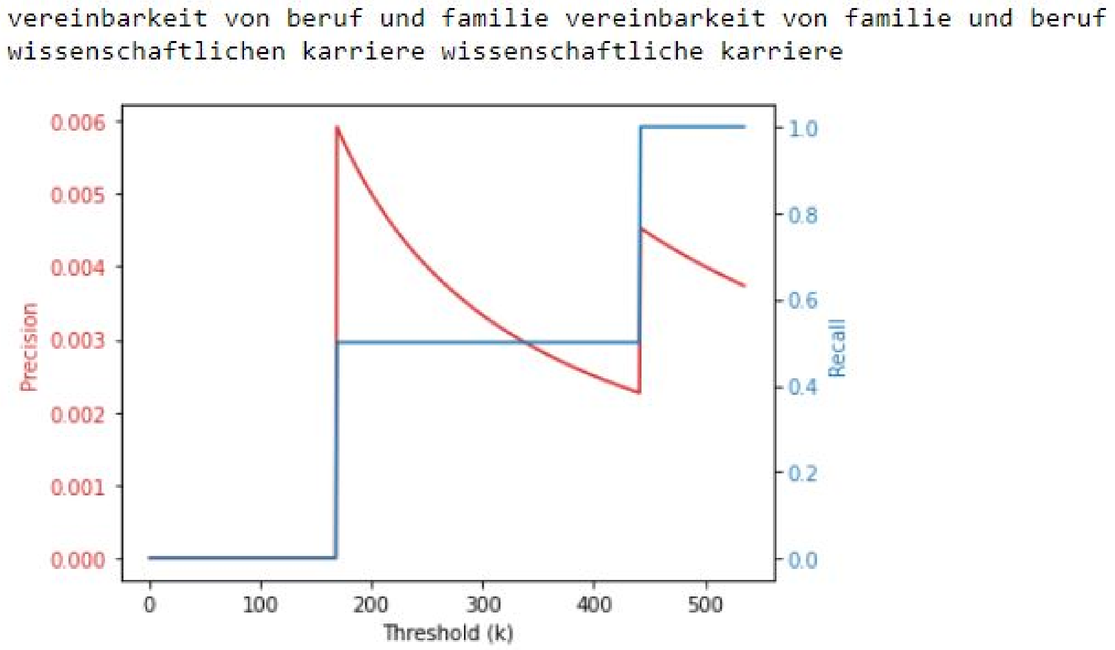
Finally, in figure 8.1 a precision-recall graph for is shown for keyphrases extracted by KeyBERT for *sid_corona* questionnaire. There are total 163 keyphrases which are extracted for this questionnaires The best precision value in this case is 0.04 and is obtained at k = 49. The best recall value in this case is 1.0 and and obtained at k = 49. This also means that for this questionnaires all the keyphrases which are in the ground truth have been retrieved by KeyBERT.

7.3.1 Evaluation based on Similarity

When extracting the keyphrases using a particular method, sometimes we were not able to retrieve the exact keyphrase. It might be the case that the keyphrase is not syntactically or lexically same. For example: table 7.5 shows the ground truth keyphrases which are within the questionnaire. For *Nacaps* questionniare, we have two keyphrases, however the methods did not retrieve these keyphrases. For the first keyphrase within the ground turth which is *wissenschaftliche Karriere* , this keyphrase was not retrieved because within the questionnaire we have *wissenschaftlichen Karriere.*

Similarly for second keyphrase within the ground truth which is *vereinbarkeit von familie und beruf*, this keyphrase was also not retrieved because within the questionnaire we have *vereinbarkeit von beruf und familie.* Since, we are more interested in semantics, calculating the similarity between the extracted keyphrases and the keyphrases which are in ground truth can help in retrieving the keyphrases.

If we evaluate it normally like we did for keyword extraction, we do not get any evaluation values for precision, recall and F_1 measure as the extracted keyphrase and ground truth keyphrase are not same. In order to overcome this, we set a threshold, for example if the extracted keyphrase is even 96% similar to what we have in the ground truth we will consider it for our evaluation. In order to compute the similarity we have considered the cosine similarity. In the end, as shown in 7.5, this helps us in retrieving these keyphrases.



```
vereinbarkeit von beruf und familie vereinbarkeit von familie und beruf
wissenschaftlichen karriere wissenschaftliche karriere
```

Figure 7.5: The figure shows a precision-recall graph for keyphrase extracted by TF-IDF for *Nacaps* questionnaire. After computing the similarity, similar keyphrases are retrieved in evaluation.

7.4 Incorporation of domain Knowledge

As mentioned in chapter 5, we made use of thesauri and controlled vocabulary. These thesauri and controlled vocabularies are from social science field and used to incorporate the domain knowledge in the extraction process of keywords and keyphrases. In order to achieve this, we have compared the keywords and keyphrases extracted by KeyBert with the keywords which are within the thesauri. Finally, we have created a mapping function between them to see if the keywords and keyphrases extracted by the method(keyBERT) best represent them in the thesauri.

7.4.1 Evaluation for Keyword Extraction

For evaluation purposes, we weighed keywords extracted by KeyBert (W$_i$) and the keywords that we get after mapping (W$_j$). We have introduces several scenarios and cases as mentioned in Chapter 5 to give higher importance to one of the variables W$_i$ and W$_j$ and evaluated accordingly. Out of four thesauri that we have selected, we will present the evaluation results for TheSoz thesaurus. The results for other can be access from Appendix. We have used precision, recall and F_1 measure for evaluation.

TheSoz - Thesaurus for the Social Sciences:

A) Baseline

In table 7.6, evaluation results for baseline are put forth. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire *WeGe_W2* is 513 and all the keywords which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 63.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	561	114	114
WeGe_W2	513	63	63
StuMa2020	168	168	168
Studierendensurvey2016	706	188	188
Promopanel_W2	176	47	47
Promopanel_W3	149	149	149
Promopanel_W4	148	39	39
Promopanel_W5	146	146	146
WeGe_W3	458	83	83
sid_corona	664	99	99

Table 7.6: The table shows evaluation results based on baseline assumption.

B) Introducing the regularization factor(λ) with respect to SIM

B.1) Giving more weightage to keywords within thesauri:

In table 7.7, evaluation results evaluation results based on the regularization factor(λ). In this case, more weightage is given to keywords within thesaurus. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire *WeGe_W2* is 497 and all the keywords which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 185 and 63 respectively.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	557	109	114
WeGe_W2	497	185	63
StuMa2020	64	64	168
Studierendensurvey2016	788	228	188
Promopanel_W2	458	227	47
Promopanel_W3	355	80	149
Promopanel_W4	310	58	39
Promopanel_W5	186	186	146
WeGe_W3	164	164	83
sid_corona	455	243	99

Table 7.7: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keywords within thesaurus.

B.2) Giving more weightage to keyword extracted by keyBERT

In table 7.8, evaluation results based on the regularization factor(λ). In this case, more weightage is given to keywords extracted by KeyBERT. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire *WeGe_W2* is 505 and all the keywords which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 183.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	534	114	114
WeGe_W2	505	183	183
StuMa2020	72	72	72
Studierendensurvey2016	855	494	494
Promopanel_W2	402	71	71
Promopanel_W3	359	65	65
Promopanel_W4	311	49	49
Promopanel_W5	206	206	206
WeGe_W3	184	184	184
sid_corona	449	272	272

Table 7.8: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keywords extracted by KeyBERT.

C) Introducing the regularization factor(γ) with respect to R_s

C.1) Giving more weightage to keywords extracted by keyBERT

In table 7.9, evaluation results based on the regularization factor(γ). In this case, more weightage is given to keywords extracted by KeyBERT. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire *WeGe_W2* is 505 and all the keywords which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 183.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	535	115	115
WeGe_W2	505	183	183
StuMa2020	72	72	72
Studierendensurvey2016	861	248	248
Promopanel_W2	402	71	71
Promopanel_W3	360	65	65
Promopanel_W4	311	51	51
Promopanel_W5	206	206	206
WeGe_W3	183	183	183
sid_corona	449	272	272

Table 7.9: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keywords extracted by KeyBERT.

C.2) Giving more weightage to keyword within thesauri

In table 7.10, evaluation results based on the regularization factor(γ). In this case, more weightage is given to keywords within thesaurus. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire *WeGe_W2* is 397 and all the keywords which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 259.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	680	265	265
WeGe_W2	397	259	259
StuMa2020	41	41	41
Studierendensurvey2016	553	162	162
Promopanel_W2	598	197	197
Promopanel_W3	335	39	39
Promopanel_W4	290	140	140
Promopanel_W5	172	172	172
WeGe_W3	224	224	224
sid_corona	447	153	153

Table 7.10: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keywords within thesaurus.

D) Introducing the regularization factors(λ and γ) with respect to R_s and SIM

D.1) Giving more weightage to keywords extracted by keyBERT

In table 7.11, evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to keywords extracted by KeyBERT. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire *WeGe_W2* is 509 and all the keywords which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 184 and 259 respectively.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	531	115	265
WeGe_W2	509	184	259
StuMa2020	73	73	41
Studierendensurvey2016	913	258	162
Promopanel_W2	365	64	197
Promopanel_W3	360	60	39
Promopanel_W4	311	42	140
Promopanel_W5	226	226	172
WeGe_W3	198	198	224
sid_corona	452	288	153

Table 7.11: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to keywords extracted by KeyBERT.

D.2) Giving more weightage to keywords within thesauri

In table 7.12, evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to keywords within thesaurus. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire *WeGe_W2* is 392 and all the keywords which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 392.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	688	216	260
WeGe_W2	392	392	392
StuMa2020	41	41	41
Studierendensurvey2016	537	322	322
Promopanel_W2	603	191	191
Promopanel_W3	336	39	39
Promopanel_W4	288	140	140
Promopanel_W5	171	171	171
WeGe_W3	227	112	227
sid_corona	440	147	147

Table 7.12: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to keywords within thesaurus.

D.3) Giving less weightage to keyword extracted by keyBERT and within thesauri

In table 7.13, evaluation results based on the regularization factor(λ and γ). In this case, less weightage is given to both keywords within thesaurus and keywords extracted by KeyBERT. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire *WeGe_W2* is 407 and all the keywords which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 255.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	665	274	274
WeGe_W2	407	255	255
StuMa2020	41	41	41
Studierendensurvey2016	573	163	163
Promopanel_W2	594	207	207
Promopanel_W3	337	39	39
Promopanel_W4	290	143	143
Promopanel_W5	178	178	178
WeGe_W3	222	222	222
sid_corona	449	77	77

Table 7.13: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, less weightage is given to both keywords within thesaurus and keywords extracted by KeyBERT.

D.4) Giving more weightage to keyword extracted by keyBERT and within thesauri

In table 7.14, evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to both keywords within thesaurus and keywords extracted by KeyBERT. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire *WeGe_W2* is 501 and all the keywords which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 182.

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F_1-score
Nacaps	546	129	129
WeGe_W2	501	182	182
StuMa2020	71	71	71
Studierendensurvey2016	837	480	480
Promopanel_W2	422	245	245
Promopanel_W3	357	72	72
Promopanel_W4	309	55	55
Promopanel_W5	197	197	197
WeGe_W3	175	175	175
sid_corona	455	258	258

Table 7.14: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to both keywords within thesaurus and keywords extracted by KeyBERT.

7.4.2 Evaluation for keyphrase Extraction

In order to evaluate the keyphrases, we weighed keyphrases extracted by KeyBert (W_i) and the keyphrases that we get after mapping (W_j). We have introduces several scenarios and cases as mentioned in Chapter 5 to give higher importance to one of the variables W_i and W_j and evaluated accordingly. As we can see from the results, out of all the questionnaires (cf. 7.5) which have the ground truth keyphrases, only two questionnaires have evaluation values.

A) Baseline

In table 7.15, evaluation results for baseline are put forth. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire*sid_corona* is 149 and all the keyphrases which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 62.

Questionnaires	**Top-k based on Recall**	**Top-k based on Precision**	**Top-k based on F_1-score**
Nacaps	2	2	2
sid_corona	149	62	62

Table 7.15: The table shows evaluation results based on baseline assumption.

B) Introducing the regularization factor(λ) with respect to SIM

B.1) Giving more weightage to keyphrase within thesauri:

In table 7.16, evaluation results based on regularization factor(λ) are put forth. In this case, more weightage is given to keyphrases within thesaurus. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire *sid_corona* is 153 and all the keyphrases which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 58.

Questionnaires	**Top-k based on Recall**	**Top-k based on Precision**	**Top-k based on F_1-score**
Nacaps	1	1	1
sid_corona	153	58	58

Table 7.16: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keyphrases within thesaurus

B.2) Giving more weightage to keyphrase extracted by keyBERT

In table 7.17, evaluation results based on regularization factor(λ) are put forth. In this case, more weightage is given to keyphrases extracted by KeyBERT. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire*sid_corona* is 146 and all the keyphrases which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 146.

Questionnaires	**Top-k based on Recall**	**Top-k based on Precision**	**Top-k based on F_1-score**
Nacaps	2	2	2
sid_corona	146	146	146

Table 7.17: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keyphrases extracted by KeyBERT

C) Introducing the regularization factor(γ) with respect to R_s

C.1) Giving more weightage to keyphrase extracted by keyBERT

In table 7.18, evaluation results based on regularization factor(γ) are put forth. In this case, more weightage is given to keyphrases extracted by KeyBERT. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire *sid_corona* is 148 and all the keyphrases which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 148.

Questionnaires	**Top-k based on Recall**	**Top-k based on Precision**	**Top-k based on F_1-score**
Nacaps	2	2	2
sid_corona	148	148	148

Table 7.18: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keyphrases extracted by KeyBERT.

C.2) Giving more weightage to keyphrase within thesauri

In table 7.19, evaluation results based on regularization factor(γ) are put forth. In this case, more weightage is given to keyphrases within thesaurus. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire*sid_corona* is 178 and all the keyphrases which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 18.

Questionnaires	**Top-k based on Recall**	**Top-k based on Precision**	**Top-k based on F_1-score**
Nacaps	1	1	1
sid_corona	178	18	18

Table 7.19: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keyphrases within thesaurus.

D) Introducing the regularization factors(λ and γ) with respect to R_s and SIM

D.1) Giving more weightage to keyphrase extracted by keyBERT

In table 7.20, evaluation results based on regularization factor(λ and γ) are put forth. In this case, more weightage is given to keyphrases extracted by keyBERT. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire*sid_corona* is 152 and all the keyphrases which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 152.

Questionnaires	**Top-k based on Recall**	**Top-k based on Precision**	**Top-k based on F_1-score**
Nacaps	2	2	2
sid_corona	152	152	152

Table 7.20: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to keyphrases extracted by KeyBERT.

D.2) Giving more weightage to keyphrase within thesauri

In table 7.21, evaluation results based on regularization factor(λ and γ) are put forth. In this case, more weightage is given to keyphrases within thesaurus. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire*sid_corona* is 177 and all the keyphrases which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 16.

Questionnaires	**Top-k based on Recall**	**Top-k based on Precision**	**Top-k based on F_1-score**
Nacaps	1	1	1
sid_corona	177	16	16

Table 7.21: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to keyphrases within thesaurus.

D.3) Giving less weightage to keyphrase extracted by keyBERT and within thesauri

In table 7.22, evaluation results based on regularization factor(λ and γ) are put forth. In this case, less weightage is given to both keyphrases within thesaurus and keyphrases extracted by KeyBERT. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire*sid_corona* is 171 and all the keyphrases which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 33.

Questionnaires	**Top-k based on Recall**	**Top-k based on Precision**	**Top-k based on F_1-score**
Nacaps	1	1	1
sid_corona	171	33	33

Table 7.22: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, less weightage is given to both keyphrases within thesaurus and keyphrases extracted by KeyBERT.

D.4) Giving more weightage to keyphrase extracted by keyBERT and within thesauri

In table 7.23, evaluation results based on regularization factor(λ and γ) are put forth. In this case, more weightage is given to both keyphrases within thesaurus and keyphrases extracted by KeyBERT. Top-k is mentioned with respect to recall, precision and F_1 measure. For example: the top-k recall for questionnaire*sid_corona* is 152 and all the keyphrases which are in the ground truth have been recovered at this point. Similarly, the precision and F_1 score for the same is 152.

Questionnaires	**Top-k based on Recall**	**Top-k based on Precision**	**Top-k based on F_1-score**
Nacaps	2	2	2
sid_corona	152	152	152

Table 7.23: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to both keyphrases within thesaurus and keyphrases extracted by KeyBERT.

7.5 Summary

In this chapter, we illustrate the evaluation phase of our thesis. We also compare some significant findings for the evaluation of different methods which are *TF-IDF*, *TextRank* and *KeyBERT*. Firstly, we discussed the Evaluation metric used, such as precision, recall and F_1 measure. Secondly, we compared the results of keyword and keyphrase extraction and compared them with the help of precision-recall graphs. In order to also capture the semantics, we have also evaluated keyphrase extraction based on cosine similarity. Finally, we concluded the chapter by discussing the evaluation results we got after incorporating the thesauri and controlled vocabularies into the extraction process.

8. Discussion

This chapter will focus on a qualitative evaluation of our workflow(cf. Figure 5.1). First, we will investigate how well the approaches performed for keyword and keyphrase extraction. For instance, whether TF-IDF, TextRank or KeyBERT has extracted all the keywords and keyphrases which were in the ground truth. In addition, we compared the approaches and saw how fast or precisely a particular technique extracted keywords and keyphrases.

8.1 Analysis of keyword and keyphrase extraction results for questionnaires

This section highlights the different scenarios wherein each approach, namely TF-IDF, TextRank and KeyBERT stood out. The parameters we use for analysis are top-k based on recall, top-k based on precision, the total number of keywords/keyphrases within ground truth, and the total number of keywords/keyphrases extracted by the method. Each of the scenarios is explained in detail as follows:

Keyword Extraction

1. Every keywords extracted by all approaches (WeGe_W3)

In figure 4.2, we see that the questionnaire *WeGe_W3* has comparatively less number of words(approx. 2000), which also translates to it having less number of question-answer pairs (approx. 150) as shown in 4.7. In addition, as shown in figure 4.9, the uniqueness of this questionnaire is also low (approx. 2%). Low uniqueness also means no new words in this questionnaire overlap with the other questionnaire. Therefore, we can assume that a lot of general-purpose words might be used in this questionnaire. These factors might have played a role in all the extracted keywords since no specialized words were used. Furthermore, the simplistic approach of TF-IDF worked well here as the ranking would have to be done between fewer words. Hence, making it more probable for the desired keywords to be extracted comparatively more quickly.

As mentioned in table 8.1, all keywords in the ground truth are retrieved with each approach, which also means that recall is 1. TF-IDF has the best top-k recall, followed by TextRank and KeyBERT. However, TextRank has the best top-k precision with a value 0.2 followed by KeyBERT whose top-k precision value is 0.02. Finally, TF-IDF, whose top-k precision value is 0.05. Although approaches were able to extract the necessary keywords on the recall front, the TF-IDF performed the best, and on the precision front, TextRank performed the best.

Methods	Top-k based on Recall	Top-k based on Precision	Total number of keywords within ground truth	Total Number of keywords extracted by method
TF-IDF	69	37	3	3
TextRank	120	4	3	3
KeyBERT	281	34	3	3

Table 8.1: The table shows the performance of keyword extraction for WeGe_W3 questionnaire. In this case, all keywords within the ground are extracted by all methods.

2. Case where TF-IDF is leading (Promopanel_W3)

Contradicting to the *WeGe_w3* questionnaire, for *Promopanel_W3* questionnaire there are more number(approx. 8200) of words (cf. 4.2). However, the number of words does not conclude enough, as we can see in figure 4.7 that it has less number of question-answer pairs (approx. 200). If the question-answer pair is less, which also means that the length of the text is not big within each question-answer pair. Similarly, uniqueness is also less in this case (approx. 1%). Low uniqueness means no new words in this questionnaire do not overlap with the other questionnaire. Therefore, we can assume that a lot of general-purpose words might be used in this questionnaire. As other factors are similar to *WeGe_w3*, we can assume that TF-IDF might work best for this questionnaire.

In this questionnaire, all the keywords are not extracted by either of the methods. In table 8.2, the best top-k based on recall is achieved using TF-IDF with a value 0.75. However, it has extracted three out of four keywords in the ground truth. TextRank is second with respect to top-k based on recall with a recall value of 0.75, which has also extracted three out of six keywords from the ground truth. For KeyBERT the recall value with respect to top-k is also 0.75 , which has extracted three out of six keywords from the ground truth. In addition, TF-IDF has the best top-k based on precision, and the value for precision is 1. TextRank has the second-best top-k based on precision with a precision value of 0.062 followed by KeyBERT with a precision value of 0.014.

Methods	Top-k based on Recall	Top-k based on Precision	Total number of keywords within ground truth	Total Number of keywords extracted by method
TF-IDF	35	1	4	3
TextRank	58	15	4	3
KeyBERT	200	200	4	3

Table 8.2: The table shows the performance of keyword extraction for Promopanel_W3 questionnaire. In this case, TF-IDF outperformed TextRank and KeyBERT based on recall and precision.

3. Case where TextRank is leading (WeGe_W2)

In figure 4.2, we see that the questionnaire *WeGe_W2* has approx. 2100 number of words, and similarly, the length of question-answer pairs is more than 200, as shown in 4.7. In addition, as shown in 4.9, the uniqueness of this questionnaire is around 6%. In this case, the trend of word count and length of question-answer pair is similar to *WeGe_W3* questionnaire. The only thing standing out is the uniqueness which is higher. This also means that in this questionnaire, the overlapping words will be fewer. In this case, we can say a very specific pattern of words is written for the questionnaires. In this questionnaire, TextRank might work best.

In this questionnaire, all the keywords in the ground truth are extracted using TF-IDF, TextRank and KeyBERT. In table 8.3, it is shown that TextRank achieves the best top-k based on recall with a value of 1. followed by TF-IDF with a recall value of 1 and similarly, for KeyBERT, the recall is also 1. In addition, TextRank has the best top-k based on precision with a value 0.08 followed by KeyBERT with a precision value of 0.02 with respect to the given top-k. Finally, for TF-IDF the precision value is 0.03 with respect to top-k.

Methods	Top-k based on Recall	Top-k based on Precision	Total number of keywords within ground truth	Total Number of keywords extracted by method
TF-IDF	169	57	3	3
TextRank	44	22	3	3
KeyBERT	313	34	3	3

Table 8.3: The table shows the performance of keyword extraction for WeGe_W2 questionnaire. In this case, TextRank outperformed TF-IDF and KeyBERT based on recall and precision.

4. Case where KeyBERT is leading (nacaps)

In figure 4.2, we see that the questionnaire *nacaps* has approx 6500 words, which also translates to it having a higher number of question-answer pairs (approx. 500) as shown in 4.7. In addition, as shown in 4.9, the uniqueness of this questionnaire is low(approx. 1%). Even after having a high number of question-answer pairs, the uniqueness is low. Low uniqueness means no new words in this questionnaire overlap with the other questionnaire. Therefore, we can assume that a lot of general-purpose words might be used in this questionnaire. In KeyBERT, for each token, the similarity is calculated with respect to the sentence. Therefore, if it had more sentences in question-answer pairs, it might have efficiently calculated the similarities. Consequently, we can assume that KeyBERT might work best for this questionnaire.

In this questionnaire, all the keywords in the ground truth are extracted using TF-IDF and KeyBERT. However, TextRank retrieved six out of 7 keywords for this questionnaire. In table 8.4, the best top-k based on recall is achieved using TextRank with a value 0.85, followed by KeyBERT with a recall value 1. Similarly, for TF-IDF the recall is 1. In addition, TF-IDF and TextRank both have same top-k based on precision with a precision value of 1. However, KeyBERT has a precision value of 0.0151 with respect to the given top-k. In this case, KeyBERT outperformed based on recall.

Methods	Top-k based on Recall	Top-k based on Precision	Total number of keywords within ground truth	Total Number of keywords extracted by method
TF-IDF	688	1	7	7
TextRank	317	1	7	6
KeyBERT	477	394	7	7

Table 8.4: The table shows the performance of keyword extraction for nacaps questionnaire. In this case, KeyBERT outperformed TF-IDF and TextRank based on recall and number of keywords extracted from ground truth.

Keyphrase Extraction

Case where TextRank is leading (sid_corona)

In figure 4.2, we see that the questionnaire *sid_corona* has approx. 2200 words, which also translates to having fewer question-answer pairs (approx. 150, cf. Figure 4.7).

In addition, as shown in figure 4.9, the uniqueness of this questionnaire is high(approx. 9%). This also means that in this questionnaire the overlapping words will be less. In this case we can say very specific pattern of words are written for the questionnaires. All of these factors might have played a role in all the keyphrases being extracted. Furthermore, TextRank might worked best for this questionnaire.

As mentioned in figure 8.5, for this questionnaire, all keyphrases in the ground truth are extracted by TextRank and KeyBERT. However, only one keyphrase is extracted using TF-IDF. As a result, the best top-k based on recall is achieved by TextRank with a value of 1 followed by KeyBERT with a recall value of 1. Similarly, for TF-IDF, the recall is 0.5. In addition, TextRank has the best top-k based on precision with a precision value of 0.09 followed by KeyBERT with a precision value of 0.04. Finally, for TF-IDF, the precision value is 0.003 with respect to top-k.

Methods	Top-k based on Recall	Top-k based on Precision	Total number of keywords within ground truth	Total Number of keywords extracted by method
TF-IDF	291	291	2	1
TextRank	47	10	2	2
KeyBERT	49	49	2	2

Table 8.5: The table shows the performance of keyphrase extraction for sid_corona questionnaire. In this case, TextRank outperformed TF-IDF and KeyBERT based on recall and precision.

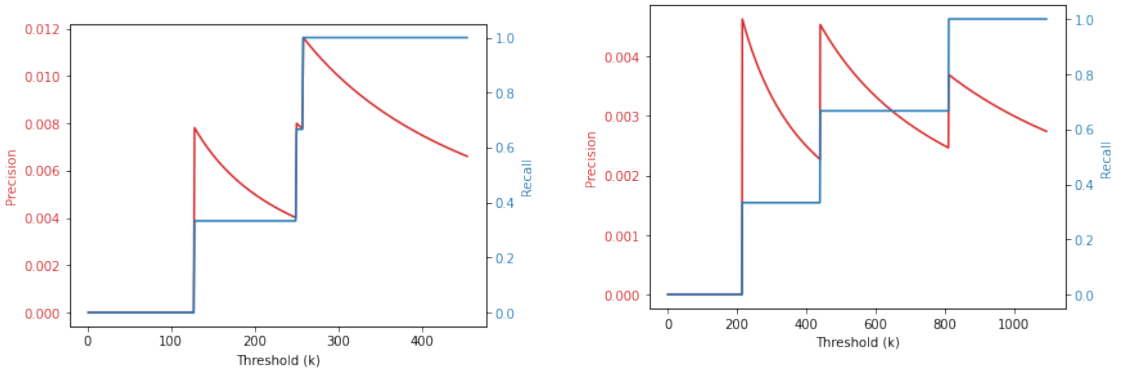
So, we tried to link the 3 parameters namely word count, question-answer pair length and uniqueness. We could not identify any fixed pattern apart from relation between lower uniqueness and length of words. For example, one indication that is observed is, if a questionnaire has high uniqueness then it prefers TextRank. If there are high number of question-answer pair for a particular questionnaires then it was preferred by KeyBERT. For lower uniqueness and less question-answer pair length, it was preferred by TF-IDF. However, from these observation it can not be conclude and generalized and this would have to be investigated further in the future work.

8.2 Application for researchers/scientists

In chapter 7, we quantitatively mentioned the importance of recall, precision and F_1-measure. The goal of this thesis is also to build a decision support system which can give assistance to researchers which in-turns helps them to choose the relevant keywords and keyphrases. Based on the requirement of researchers, some modification can be made with respect to recall, precision and F_1 measure. For instance, We advise extracting the top-k keywords based on recall if the requirements of researchers is to have all the keywords within ground truth as a result of the technique. In contrary, We suggest top-k based on precision if a researcher wants to make sure that the keywords retrived are accurate. Alternately, researchers might have the choice to balance both metrics by using F_1 measure. Overall the keywords and keyphrase extracted can cater to the direct as well as the indirect requirement for the researchers.

8.2.1 Direct requirements - no semantics involved

If the researchers are not sure about their requirements based on precision, recall and F_1 measure. For example, in figure 8.1a, the method point towards the same top-k and also extract all the keywords from ground truth. In this case, it is possible for researchers to chose based on their use case.



(a) This sub-figure shows a precision-recall graph for keywords extracted by KeyBERT for the sid_corona questionnaire.

(b) This sub-figure shows a precision-recall graph for keywords extracted by keyBERT for the Studierdenensurvey2016 questionnaire.

Figure 8.1: The figure depicts the precision-recall graphs for keywords extracted by KeyBERT

However, in figure 8.1b the top-k is for recall, precision and F_1 measure is totally different. In such scenario, the researchers have to take a call on what to choose. So, this also illustrates the flexibility of our system so that either researchers can take up the recommendation from the system or can also use their intuition.

Finally, in direct requirement of researchers where no semantics is involved, only the text of questionnaire is used, it can assist to researchers if they do not want to go outside the scope of what is written in the questionnaire.

8.2.2 Indirect requirements - semantics involved

In order to meet the indirect requirements, firstly the external domain knowledge such as thesauri, controlled vocabularies(cf. Section 5.5) is incorporated and secondly, we also tried to include the semantics.

Domain Knowledge:

Many times several keywords and keyphrases are missed or not used which are important in the field of social science. In order to have an extra edge so that we can include those missing keywords and keyphrases, we used external knowledge such as thesauri and controlled vocabularies (cf. Section 4.4). For example, if a particular word which is present in the thesaurus and not present in the questionnaire text and that keywords clearly define that particular survey. We want to recommend these type of keywords and keyphrases, because they might make more sense to the researchers for their perspective in choosing keywords and keyphrases.

Some keywords which were retrieved by using thesauri includes *forschungsthema arbeitssituation*. These keywords were not annotated by researchers but occurs in the top keywords which are retrieved from thesaurus(Thesoz (cf. Section 4.4)). Nacaps is known for the storage of the occupational data of the people oriented to research and the keywords *forschungsthema* and textitarbeitssituation perfectly matched the domain of the questionnaire.

Semantics:

When researchers build the questionnaires and ask questions, they do not follow the direct approach of asking people about a specific topic. Instead, they build the question in such a way that by asking about this topic but not specifically with the right words. For example: If we want to measure the performance of a group of students within the class by asking them questions, we would want to avoid asking straightforward questions such as, how do you stand in the class? How smart are you? That is why the researchers rather try to rephrase or find indicator questions that do not mention the topic. The rephrased question for this topic is, how do you feel that your skills are or whether your skills are better than other students? In the end, if we try to create a keyword from this, it can be “comparative students performance”. However, we will never find this keyword inside the questionnaires, Because these questionnaires are usually built in a way which the researcher calls latent features, and these latent features are not directly expressive but are kind of hidden in the text of the questionnaires. In this case, by understanding the context indirectly we need to find keywords.

As mentioned in chapter 5, in order to capture the semantics of keywords and keyphrases, we have incorporated word embeddings based on sentence transformers. Therefore, we first compute the embeddings of all the keywords/keyphrases which are extracted by KeyBERT and similarly, we compute the embeddings for keywords/keyphrases within thesauri. All the keywords which KeyBERT extracts will always be inside the thesaurus, as the thesaurus is an extended corpus. In the end, if we find keywords which are semantically similar, then we pick that keyword. However, this process was made stringent for our implementation as it needed to consider more similarities. Still, in future work, we can also play around the similarity values to pick up more semantically similar keywords or try out different embeddings.

For example: The keyphrase *vorbereitungskurs* is extracted by KeyBERT. Another keyword which is *unterrichtsvorbereitung*, is present in the thesaurus, and the semantic similarity between these two keyphrases is 0.84. This keywords describes the

WeGe_W2 questionnaire, and still being semantically similar, it was not part of the hand-annotated keyphrases.

8.3 Summary

In this section, a brief summary of the discussion chapter is provided. Firstly, an analysis is made on the keywords and keyphrase extracted for the questionnaires. This analysis was done separately for keywords and keyphrases. We chose a few questionnaire for which the methods worked best. he parameters that we use for analysis are, top-k based on recall, top-k based on precision, total number of keywords/keyphrases within ground truth, and total number of keywords/keyphrases extracted by method. Based on these parameters we have analyzed the results based on word count, question-answer pair length and uniqueness. Furthermore, an emphasis is made on the the requirement of researchers based on some modification which can be made with respect to recall, precision and F_1 measure. Two requirements were discussed, the first one is Direct requirements with no semantics where an elaboration is made on when a researcher can choose the metric importance based on their use case and the situation where they have to take a call between recall, precision . The second one is and Indirect requirements with semantics involved in which we explain the incorporation of domain knowledge as well some problem of the questionnaires and their semantic keywords and keyphrases.

9. Conclusion and Future Work

In this final chapter of the thesis, the main contribution is summarized by answering the research questions. Furthermore, emphasis is given to technical limitations related to approaches used in this work. Finally, we describe potential enhancements and suggest interesting future work directions.

9.1 Conclusion

In this thesis, we review three methods for keywords and keyphrase extraction. This thesis aims to extract keywords and keyphrases from the questionnaire using statistical, graph and embedding approaches.

To answer the **RQ-1**, different types of questions are used, such as the Likert scale, multiple choice and open-ended and extracting them from the questionnaires. This also helps in diversification, as keywords and keyphrases extracted will be from different types of questions and answers. To answer the **RQ-2**, we saw that embedding-based methods work better in scenarios where the uniqueness of words and the number of questions are also high. This might be because similar words appear and are captured by the embeddings, which can lead to a good extraction of keywords and keyphrases. On the other hand, the statistical-based approach works well in smaller vocabularies (due to limited questions, cf. Table 8.2) with a lower uniqueness of words, due to which repeating patterns can be detected. Finally, to answer the **RQ-3**, we incorporated a domain-specific thesaurus to increase the number of matching keywords and keyphrases. That way domain specific matches can also be preferred. Additionally, close-matched keywords and keyphrases such as lemmatized and synonym words are also considered using embedding-based similarities to enhance the candidate space further and capture the semantics. After answering all the research questions, on this basis, we can not conclude if either of the statistical, graph-based or embedding-based approaches is best. Every method has its advantages and disadvantages. Different techniques can be used based on the researchers' priority or for a particular scenario. Based on word length and uniqueness, we can still provide some indication that a specific method works best.

However, we can not specifically say which method works best overall. It is on the researcher to decide their requirements and based on that, they can choose a method.

9.2 Limitations

In this section, elaboration is made on the limitation that is encountered in our thesis. First, we will go through some of the technical limitations, followed by limitations related to approaches.

Technical Limitations

In this thesis, all the experimentation is performed with 19 questionnaires (cf. Chapter 4). So, the overall dataset is comparatively small to achieve a deeper analysis of the differences in the approaches.

Moreover, out of those 19 questionnaires, only ten have the ground truth keywords and keyphrases present within the text. So, The amount of keywords and keyphrases that are part of the extractable ground truth is minimal, due to which pure extractive-based methods will, by default, not perform well on the whole set of ground truth keywords.

Limitations related to approaches

Firstly, Only extractive approaches are being tested on the documents. For example, in this study, only the keywords and keyphrases within the questionnaires are considered for experimentation. Due to the nature of the data set, other abstractive approaches would need testing and comparing. Therefore, keywords/keyphrases within the text were not included in our analysis. For example, the ground keyphrase from *nacaps* questionnaire includes *"Wissenschaftlicher Nachwuchs"*. However, we did not consider this keyphrase for our experimentation because this keyphrase was not inside the questionnaire text. Therefore the results do not fully show what is possible. Furthermore, The domain knowledge incorporated into the approach is fundamental and only covers a direct comparison to a thesaurus. It is still being determined whether the thesaurus alone can cover the search space and whether the search can be directed further using expert knowledge or another source (like supporting documents).

9.3 Future Work

From the future work perspective, we can explore adding a supporting document. For example, we can add documents that describe the questionnaires from a similar field and so on. This could provide the possibility to determine better which of the matched keywords and keyphrases are more relevant, thus enhancing the ranking scheme. In addition, we can use expert knowledge to label the documents for an extractive approach (i.e. only keywords in the text are allowed) or to guide the ranking and extraction approach. The latter could be achieved by changing the ranking function or adjusting the extraction algorithm.

Furthermore, As mentioned in the technical limitations, the amount of data we have is less, and the domain knowledge we can extract is also limited. Therefore the regex pattern(cf. Section 5.4) as an example might still need improvement, and it could be extended or changed if either domain experts do it or we have more data which helps in deriving the pattern. Finally, for better qualitative analysis and better usage of keywords and keyphrases extracted by methods, a user interface can be designed in future.

Appendix

The European Language Social Science Thesaurus (ELSST)

1. Evaluation for keyword extraction

A) Baseline

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	918	654	918
WeGe_W2	237	237	237
StuMa2020	85	85	85
Studierdenensurvey2016	1231	399	399
Promopanel_W2	622	8	8
Promopanel_W3	349	10	10
Promopanel_W4	508	8	8
Promopanel_W5	439	14	14
WeGe_W3	326	326	326
sid_corona	570	570	570

Table 1: The table shows evaluation results based on baseline assumption.

B) Introducing the regularization factor(λ) with respect to SIM

B.1) Giving more weighatge to keywords within thesauri:

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	922	654	918
WeGe_W2	208	237	237
StuMa2020	81	85	85
Studierdenensurvey2016	1270	399	399
Promopanel_W2	626	8	8
Promopanel_W3	346	10	10
Promopanel_W4	514	8	8
Promopanel_W5	436	14	14
WeGe_W3	320	326	326
sid_corona	565	570	570

Table 2: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keywords within thesaurus.

B.2) Giving more weighatge to keywords extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	888	888	888
WeGe_W2	277	140	140
StuMa2020	85	85	85
Studierdenensurvey2016	1198	1198	1198
Promopanel_W2	598	4	4
Promopanel_W3	341	8	8
Promopanel_W4	498	4	4
Promopanel_W5	447	8	8
WeGe_W3	336	336	336
sid_corona	582	582	582

Table 3: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keywords extracted by KeyBERT.

C) Introducing the regularization factor(γ) with respect to R_s

C.1) Giving more weighatge to keywords extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	901	901	901
WeGe_W2	259	144	144
StuMa2020	85	85	85
Studierdenensurvey2016	1211	1211	1211
Promopanel_W2	615	8	8
Promopanel_W3	341	10	10
Promopanel_W4	500	8	8
Promopanel_W5	442	12	12
WeGe_W3	336	336	336
sid_corona	580	580	580

Table 4: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keywords extracted by KeyBERT.

C.2) Giving more weighatge to keywords within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	994	994	994
WeGe_W2	314	94	94
StuMa2020	118	118	118
Studierdenensurvey2016	1544	394	394
Promopanel_W2	653	32	32
Promopanel_W3	309	24	24
Promopanel_W4	528	27	27
Promopanel_W5	346	26	26
WeGe_W3	308	51	51
sid_corona	598	598	598

Table 5: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keywords within thesaurus.

D) Introducing the regularization factors(λ and γ) with respect to R_s and SIM

D.1) Giving more weighatge to keywords extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	882	882	882
WeGe_W2	308	137	137
StuMa2020	93	93	93
Studierdenensurvey2016	1178	1178	1178
Promopanel_W2	589	4	4
Promopanel_W3	348	6	6
Promopanel_W4	489	4	4
Promopanel_W5	448	6	6
WeGe_W3	338	338	338
sid_corona	590	590	590

Table 6: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to keywords extracted by KeyBERT.

D.2) Giving more weighatge to keywords within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	1011	1011	1011
WeGe_W2	322	94	94
StuMa2020	118	118	118
Studierdenensurvey2016	1546	386	386
Promopanel_W2	663	24	24
Promopanel_W3	298	20	20
Promopanel_W4	528	23	23
Promopanel_W5	337	20	20
WeGe_W3	309	51	51
sid_corona	607	607	607

Table 7: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to keywords within thesaurus.

D.3) Giving less weighatge to keywords extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	940	940	940
WeGe_W2	293	98	98
StuMa2020	100	100	100
Studierdenensurvey2016	1528	382	383
Promopanel_W2	650	40	40
Promopanel_W3	320	36	36
Promopanel_W4	533	37	37
Promopanel_W5	368	37	37
WeGe_W3	313	62	62
sid_corona	586	586	586

Table 8: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, less weightage is given to both keywords within thesaurus and keywords extracted by KeyBERT.

D.4) Giving more weighatge to keywords extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	909	909	909
WeGe_W2	246	246	246
StuMa2020	83	83	83
Studierdenensurvey2016	1221	1221	1221
Promopanel_W2	618	8	8
Promopanel_W3	345	10	10
Promopanel_W4	504	8	8
Promopanel_W5	442	12	12
WeGe_W3	330	330	330
sid_corona	577	577	577

Table 9: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to both keywords within thesaurus and keywords extracted by KeyBERT.

.0.1 Evaluation for keyphrase Extraction

A) Baseline

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	226	226	226

Table 10: The table shows evaluation results based on baseline assumption.

B) Introducing the regularization factor(λ) with respect to SIM

B.1) Giving more weightage to keywords/keyphrases within thesauri:

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	243	243	243

Table 11: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keyphrases within thesaurus.

B.2) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	206	206	206

Table 12: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keyphrases extracted by KeyBERT.

C) Introducing the regularization factor(γ) with respect to R_s

C.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	214	214	214

Table 13: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keyphrases extracted by KeyBERT.

C.2) Giving more weightage to keywords/keyphrases within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	309	152	152

Table 14: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keyphrases within thesaurus.

D) Introducing the regularization factors(λ and γ) with respect to R_s and SIM

D.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	194	194	194

Table 15: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to keyphrases extracted by KeyBERT.

D.2) Giving more weightage to keywords/keyphrases within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	323	149	149

Table 16: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to keyphrases within thesaurus.

D.3) Giving less weightage to keywords/keyphrases extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	273	273	273

Table 17: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, less weightage is given to both keyphrases within thesaurus and keyphrases extracted by KeyBERT.

D.4) Giving more weightage to keywords/keyphrases extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	220	220	220

Table 18: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to both keyphrases within thesaurus and keyphrases extracted by KeyBERT.

ZBW Standard-Thesaurus Wirtschaft

1. Evaluation for keyword extraction

A) Baseline

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	854	22	22
WeGe_W2	356	128	128
StuMa2020	63	63	63
Studierdenensurvey2016	1039	609	609
Promopanel_W2	545	12	12
Promopanel_W3	285	14	14
Promopanel_W4	448	9	9
Promopanel_W5	502	13	13
WeGe_W3	122	122	122
sid_corona	585	585	585

Table 19: The table shows evaluation results based on baseline assumption.

B) Introducing the regularization factor(λ) with respect to SIM

B.1) Giving more weightage to keywords/keyphrases within thesauri:

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	852	24	24
WeGe_W2	331	131	131
StuMa2020	62	62	62
Studierdenensurvey2016	1079	636	636
Promopanel_W2	547	17	17
Promopanel_W3	280	16	16
Promopanel_W4	455	11	11
Promopanel_W5	498	19	19
WeGe_W3	110	110	110
sid_corona	577	577	577

Table 20: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keywords within thesaurus.

B.2) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	842	22	22
WeGe_W2	400	133	133
StuMa2020	62	62	62
Studierdenensurvey2016	982	982	982
Promopanel_W2	512	6	6
Promopanel_W3	299	9	9
Promopanel_W4	438	5	5
Promopanel_W5	492	9	9
WeGe_W3	152	152	152
sid_corona	598	598	598

Table 21: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keywords extracted by KeyBERT.

C) Introducing the regularization factor(γ) with respect to R_s

C.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	852	24	24
WeGe_W2	378	131	131
StuMa2020	62	62	62
Studierdenensurvey2016	994	994	994
Promopanel_W2	529	8	8
Promopanel_W3	291	12	12
Promopanel_W4	441	7	7
Promopanel_W5	491	13	13
WeGe_W3	137	137	137
sid_corona	599	599	599

Table 22: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keywords extracted by KeyBERT.

C.2) Giving more weightage to keywords/keyphrases within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	836	31	103
WeGe_W2	198	198	198
StuMa2020	59	59	59
Studierdenensurvey2016	1363	220	220
Promopanel_W2	579	61	61
Promopanel_W3	209	22	22
Promopanel_W4	471	60	60
Promopanel_W5	449	70	70
WeGe_W3	106	48	48
sid_corona	503	503	503

Table 23: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keywords within thesaurus.

D) Introducing the regularization factors(λ and γ) with respect to R_s and SIM

D.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	842	20	20
WeGe_W2	427	136	136
StuMa2020	65	65	65
Studierdenensurvey2016	952	952	952
Promopanel_W2	503	4	4
Promopanel_W3	306	8	8
Promopanel_W4	441	4	4
Promopanel_W5	486	8	8
WeGe_W3	163	163	163
sid_corona	598	598	598

Table 24: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to keywords extracted by KeyBERT.

D.2) Giving more weightage to keywords/keyphrases within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	834	31	100
WeGe_W2	204	204	204
StuMa2020	61	61	61
Studierdenensurvey2016	1374	203	203
Promopanel_W2	578	61	61
Promopanel_W3	206	22	22
Promopanel_W4	467	60	60
Promopanel_W5	444	70	70
WeGe_W3	105	48	48
sid_corona	503	503	503

Table 25: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to keywords within thesaurus.

D.3) Giving less weightage to keywords/keyphrases extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	846	31	31
WeGe_W2	189	189	189
StuMa2020	48	48	48
Studierdenensurvey2016	1336	252	252
Promopanel_W2	576	71	71
Promopanel_W3	220	24	24
Promopanel_W4	469	65	65
Promopanel_W5	459	85	85
WeGe_W3	110	52	52
sid_corona	518	518	518

Table 26: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, less weightage is given to both keywords within thesaurus and keywords extracted by KeyBERT.

D.4) Giving more weightage to keywords/keyphrases extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	853	24	24
WeGe_W2	364	129	129
StuMa2020	64	64	64
Studierdenensurvey2016	1001	636	636
Promopanel_W2	541	12	12
Promopanel_W3	283	14	14
Promopanel_W4	445	9	9
Promopanel_W5	499	13	13
WeGe_W3	126	126	126
sid_corona	589	589	589

Table 27: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to both keywords within thesaurus and keywords extracted by KeyBERT.

.0.2 Evaluation for keyphrase Extraction

A) Baseline

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	164	164	164

Table 28: The table shows evaluation results based on baseline assumption.

B) Introducing the regularization factor(λ) with respect to SIM

B.1) Giving more weightage to keywords/keyphrases within thesauri:

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	168	168	168

Table 29: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keyphrases within thesaurus

B.2) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	164	164	164

Table 30: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keyphrases extracted by KeyBERT

C) Introducing the regularization factor(γ) with respect to R_s

C.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	162	162	162

Table 31: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keyphrases extracted by KeyBERT.

C.2) Giving more weightage to keywords/keyphrases within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	182	182	182

Table 32: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keyphrases within thesaurus

D) Introducing the regularization factors(λ and γ) with respect to R_s and SIM

D.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	162	162	162

Table 33: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to keyphrases extracted by KeyBERT.

D.2) Giving more weightage to keywords/keyphrases within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	188	188	188

Table 34: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to keyphrases within thesaurus.

D.3) Giving less weightage to keywords/keyphrases extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	176	176	176

Table 35: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, less weightage is given to both keyphrases within thesaurus and keyphrases extracted by KeyBERT.

D.4) Giving more weightage to keywords/keyphrases extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	163	163	163

Table 36: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to both keyphrases within thesaurus and keyphrases extracted by KeyBERT.

CESSDA Controlled Vocabulary

1. Evaluation for keyword extraction

A) Baseline

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	641	2	2
WeGe_W2	217	141	141
StuMa2020	42	42	42
Studierdenensurvey2016	587	252	252
Promopanel_W2	293	25	25
Promopanel_W3	272	23	23
Promopanel_W4	318	16	16
Promopanel_W5	362	26	26
WeGe_W3	270	57	57
sid_corona	434	434	434

Table 37: The table shows evaluation results based on baseline assumption.

B) Introducing the regularization factor(λ) with respect to SIM

B.1) Giving more weightage to keywords/keyphrases within thesauri:

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	643	2	2
WeGe_W2	204	204	204
StuMa2020	37	37	37
Studierdenensurvey2016	578	114	229
Promopanel_W2	292	26	26
Promopanel_W3	273	22	22
Promopanel_W4	319	15	15
Promopanel_W5	364	19	19
WeGe_W3	271	54	54
sid_corona	430	430	430

Table 38: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keywords within thesaurus.

B.2) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	610	4	4
WeGe_W2	311	134	134
StuMa2020	49	49	49
Studierdenensurvey2016	656	391	391
Promopanel_W2	311	40	40
Promopanel_W3	283	84	84
Promopanel_W4	298	40	40
Promopanel_W5	327	52	52
WeGe_W3	246	77	77
sid_corona	445	445	445

Table 39: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keywords extracted by KeyBERT.

C) Introducing the regularization factor(γ) with respect to R_s

C.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	637	2	2
WeGe_W2	246	142	142
StuMa2020	44	44	44
Studierdenensurvey2016	610	294	294
Promopanel_W2	301	27	27
Promopanel_W3	275	31	31
Promopanel_W4	312	18	18
Promopanel_W5	354	33	33
WeGe_W3	263	59	59
sid_corona	440	440	440

Table 40: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keywords extracted by KeyBERT.

C.2) Giving more weightage to keywords/keyphrases within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	601	2	2
WeGe_W2	210	210	210
StuMa2020	24	24	24
Studierdenensurvey2016	427	88	88
Promopanel_W2	287	17	17
Promopanel_W3	259	15	15
Promopanel_W4	303	11	11
Promopanel_W5	372	1	2
WeGe_W3	270	58	58
sid_corona	368	368	368

Table 41: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keywords within thesaurus.

D) Introducing the regularization factors(λ and γ) with respect to R_s and SIM

D.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	602	4	4
WeGe_W2	325	134	134
StuMa2020	54	54	54
Studierdenensurvey2016	669	421	421
Promopanel_W2	313	49	49
Promopanel_W3	282	82	82
Promopanel_W4	292	44	44
Promopanel_W5	311	63	63
WeGe_W3	245	81	81
sid_corona	437	437	437

Table 42: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to keywords extracted by KeyBERT.

D.2) Giving more weightage to keywords/keyphrases within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	596	2	2
WeGe_W2	217	71	217
StuMa2020	25	25	25
Studierdenensurvey2016	407	81	81
Promopanel_W2	578	61	61
Promopanel_W3	289	17	17
Promopanel_W4	258	15	15
Promopanel_W5	303	11	11
WeGe_W3	270	57	57
sid_corona	372	372	372

Table 43: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to keywords within thesaurus.

D.3) Giving less weightage to keywords/keyphrases extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	627	1	1
WeGe_W2	155	155	155
StuMa2020	24	24	24
Studierdenensurvey2016	522	137	137
Promopanel_W2	283	21	21
Promopanel_W3	269	14	14
Promopanel_W4	315	11	11
Promopanel_W5	374	8	8
WeGe_W3	276	47	47
sid_corona	412	412	412

Table 44: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, less weightage is given to both keywords within thesaurus and keywords extracted by KeyBERT.

D.4) Giving more weightage to keywords/keyphrases extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
Nacaps	637	2	2
WeGe_W2	225	140	140
StuMa2020	42	42	42
Studierdenensurvey2016	597	269	269
Promopanel_W2	597	269	269
Promopanel_W3	296	25	25
Promopanel_W4	315	16	16
Promopanel_W5	361	30	30
WeGe_W3	267	58	58
sid_corona	439	439	439

Table 45: The table shows the evaluation results based on the regularization factor((λ and γ)). In this case, more weightage is given to both keywords within thesaurus and keywords extracted by KeyBERT.

.0.3 Evaluation for keyphrase Extraction

A) Baseline

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	164	164	164

Table 46: The table shows evaluation results based on baseline assumption.

B) Introducing the regularization factor(λ) with respect to SIM

B.1) Giving more weightage to keywords/keyphrases within thesauri:

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	168	168	168

Table 47: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keyphrases within thesaurus

B.2) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	164	164	164

Table 48: The table shows the evaluation results based on the regularization factor(λ). In this case, more weightage is given to keyphrases extracted by KeyBERT

C) Introducing the regularization factor(γ) with respect to R_s

C.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	162	162	162

Table 49: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keyphrases extracted by KeyBERT.

C.2) Giving more weightage to keywords/keyphrases within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	182	182	182

Table 50: The table shows the evaluation results based on the regularization factor(γ). In this case, more weightage is given to keyphrases within thesaurus

D) Introducing the regularization factors(λ and γ) with respect to R_s and SIM

D.1) Giving more weightage to keywords/keyphrases extracted by keyBERT

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	162	162	162

Table 51: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to keyphrases extracted by KeyBERT.

D.2) Giving more weightage to keywords/keyphrases within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	188	188	188

Table 52: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to keyphrases within thesaurus.

D.3) Giving less weightage to keywords/keyphrases extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	176	176	176

Table 53: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, less weightage is given to both keyphrases within thesaurus and keyphrases extracted by KeyBERT.

D.4) Giving more weightage to keywords/keyphrases extracted by keyBERT and within thesauri

Questionnaires	Top-k based on Recall	Top-k based on Precision	Top-k based on F1-score
sid_corona	163	163	163

Table 54: The table shows the evaluation results based on the regularization factor(λ and γ). In this case, more weightage is given to both keyphrases within thesaurus and keyphrases extracted by KeyBERT.

Bibliography

Lada A Adamic and Bernardo A Huberman. Zipf's law and the internet. *Glottometrics*, 3(1):143–150, 2002. (cited on Page 17)

James Allen. *Natural language understanding*. Benjamin-Cummings Publishing Co., Inc., 1995. (cited on Page 7)

S Anjali, Nair M Meera, and MG Thushara. A graph based approach for keyword extraction from documents. In *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, pages 1–4. IEEE, 2019. (cited on Page 11)

Michal Barla, Mária Bieliková, et al. From ambiguous words to key-concept extraction. In *2013 24th International Workshop on Database and Expert Systems Applications*, pages 63–67. IEEE, 2013. (cited on Page 7)

Slobodan Beliga, Ana Meštrović, and Sanda Martinčić-Ipšić. An overview of graph-based keyword extraction methods and approaches. *Journal of information and organizational sciences*, 39(1):1–20, 2015. (cited on Page 11 and 18)

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. Simple unsupervised keyphrase extraction using sentence embeddings. *arXiv preprint arXiv:1801.04470*, 2018. (cited on Page 3, 12, and 19)

H Russell Bernard and Gery Ryan. Text analysis. *Handbook of methods in cultural anthropology*, 613, 1998. (cited on Page 8)

Michael W Berry and Jacob Kogan. *Text mining: applications and theory*. John Wiley & Sons, 2010. (cited on Page 8)

Adrien Bougouin, Florian Boudin, and Béatrice Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. In *International joint conference on natural language processing (IJCNLP)*, pages 543–551, 2013. (cited on Page 11 and 19)

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2020. (cited on Page 9 and 18)

KR1442 Chowdhary. Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649, 2020. (cited on Page 7)

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. (cited on Page 12)

Swagata Duari and Vasudha Bhatnagar. scake: semantic connectivity aware keyword extraction. *Information Sciences*, 477:100–117, 2019. (cited on Page 18)

Eugenia. Keyword extraction and classification, 2018. (cited on Page 9)

Nazanin Firoozeh, Adeline Nazarenko, Fabrice Alizon, and Béatrice Daille. Keyword extraction: Issues and methods. *Natural Language Engineering*, 26(3):259–291, 2020. (cited on Page 6)

Luit Gazendam, Christian Wartena, and Rogier Brussee. Thesaurus based term ranking for keyword extraction. In *2010 Workshops on Database and Expert Systems Applications*, pages 49–53. IEEE, 2010. (cited on Page 20)

Kazi Saidul Hasan and Vincent Ng. Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In *Coling 2010: Posters*, pages 365–373, 2010. (cited on Page 18)

Guoxiu He, Junwei Fang, Haoran Cui, Chuan Wu, and Wei Lu. Keyphrase extraction based on prior knowledge. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, pages 341–342, 2018. (cited on Page 21)

Heather Hedden. Controlled vocabularies, thesauri, and taxonomies. *Indexer*, 26(1), 2008. (cited on Page 31)

Juan P Herrera and Pedro A Pury. Statistical keyword detection in literary corpora. *The European Physical Journal B*, 63(1):135–146, 2008. (cited on Page 17)

Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. A brief survey of text mining. In *Ldv forum*, volume 20, pages 19–62, 2005. (cited on Page 8)

Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223, 2003. (cited on Page 17 and 44)

Anette Hulth, Jussi Karlgren, Anna Jonsson, Henrik Boström, and Lars Asker. Automatic keyword extraction using domain knowledge. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 472–482. Springer, 2001. (cited on Page 21)

Nitin Indurkhya and Fred J Damerau. *Handbook of natural language processing.* Chapman and Hall/CRC, 2010. (cited on Page 7)

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998. (cited on Page 20)

Thiruni D Jayasiriwardene and Gamage Upeksha Ganegoda. Keyword extraction from tweets using nlp tools for collecting relevant news. In *2020 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, pages 129–135. IEEE, 2020. (cited on Page 43)

Jaap Kamps. Improving retrieval effectiveness by reranking documents based on controlled vocabulary. In *European Conference on Information Retrieval*, pages 283–295. Springer, 2004. (cited on Page 21)

Gaganpreet Kaur. Usage of regular expressions in nlp. *International Journal of Research in Engineering and Technology IJERT*, 3(01):7, 2014. (cited on Page 42)

Harpreet Kaur and Vishal Gupta. Indexing process insight and evaluation. In *2016 International Conference on Inventive Computation Technologies (ICICT)*, volume 3, pages 1–5. IEEE, 2016. (cited on Page 8)

Tshepho Koboyatshwene, Moemedi Lefoane, and Lakshmi Narasimhan. Machine learning approaches for catchphrase extraction in legal documents. In *FIRE (Working Notes)*, pages 95–98, 2017. (cited on Page 65)

Amy N Langville and Carl D Meyer. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380, 2004. (cited on Page 20)

Jung-Tae Lee, Sang-Bum Kim, Young-In Song, and Hae Chang Rim. Bridging lexical gaps between queries and questions on large online q&a collections with compact translation models. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 410–418, 2008. (cited on Page 38)

Elizabeth D Liddy. Natural language processing. 2001. (cited on Page 7)

Marina Litvak, Mark Last, Hen Aizenman, Inbal Gobits, and Abraham Kandel. Degext—a language-independent graph-based keyphrase extractor. In *Advances in intelligent web mastering–3*, pages 121–130. Springer, 2011. (cited on Page 8)

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 257–266, 2009. (cited on Page 44)

Hans Peter Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317, 1957. (cited on Page 17)

Shruti Luthra, Dinkar Arora, Kanika Mittal, and Anusha Chhabra. A statistical approach of keyword extraction for efficient retrieval. *International Journal of Computer Applications*, 168(7):31–36, 2017. (cited on Page 9)

Debanjan Mahata, John Kuriakose, Rajiv Shah, and Roger 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)*, pages 634–639, 2018. (cited on Page 20)

Olena Medelyan and Ian H Witten. Thesaurus-based index term extraction for agricultural documents. 2005. (cited on Page 21)

Rada Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL interactive poster and demonstration sessions*, pages 170–173, 2004. (cited on Page 19)

Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004. (cited on Page 11, 19, and 44)

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. (cited on Page 12)

Tom M Mitchell and Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997. (cited on Page 8)

Mahdi Naser Moghadasi and Yu Zhuang. Sent2vec: A new sentence embedding representation with sentimental semantic. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 4672–4680. IEEE, 2020. (cited on Page 12)

Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011. (cited on Page 7)

Yukio Ohsawa, Nels E Benson, and Masahiko Yachida. Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *Proceedings IEEE International Forum on Research and Technology Advances in Digital Libraries-ADL'98-*, pages 12–18. IEEE, 1998. (cited on Page 8 and 18)

Eirini Papagiannopoulou and Grigorios Tsoumakas. A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2): e1339, 2020. (cited on Page 19)

Yili Qian, Chaochao Jia, and Yimei Liu. Bert-based text keyword extraction. In *Journal of Physics: Conference Series*, volume 1992, page 042077. IOP Publishing, 2021. (cited on Page 9)

Ying Qin. Applying frequency and location information to keyword extraction in single document. In *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, volume 3, pages 1398–1402. IEEE, 2012. (cited on Page 18)

Gollam Rabby, Saiful Azad, Mufti Mahmud, Kamal Z Zamli, and Mohammed Mostafizur Rahman. A flexible keyphrase extraction technique for academic literature. *Procedia computer science*, 135:553–563, 2018. (cited on Page 41)

Dominique Ritze and Kai Eckert. Thesaurus mapping: a challenge for ontology alignment? In *OM*, 2012. (cited on Page 31)

Mukund Rungta, Rishabh Kumar, Mehak Preet Dhaliwal, Hemant Tiwari, and Vanraj Vala. Transkp: Transformer based key-phrase extraction. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020. (cited on Page 43)

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988. (cited on Page 9 and 10)

Tim Schopf, Simon Klimek, and Florian Matthes. Patternrank: Leveraging pretrained language models and part of speech for unsupervised keyphrase extraction. *arXiv preprint arXiv:2210.05245*, 2022. (cited on Page 20)

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008. (cited on Page 8)

Mike Scott and Christopher Tribble. *Textual patterns: Key words and corpus analysis in language education*, volume 22. John Benjamins Publishing, 2006. (cited on Page 2 and 5)

Aakanksha Singhal and DK Sharma. Keyword extraction using renyi entropy: a statistical and domain independent method. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 1970–1975. IEEE, 2021. (cited on Page 3 and 17)

Amit Singhal et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001. (cited on Page 8)

Chengyu Sun, Liang Hu, Shuai Li, Tuohang Li, Hongtu Li, and Ling Chi. A review of unsupervised keyphrase extraction methods using within-collection resources. *Symmetry*, 12(11):1864, 2020. (cited on Page 63)

Ah-Hwee Tan et al. Text mining: The state of the art and the challenges. In *Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases*, volume 8, pages 65–70, 1999. (cited on Page 8)

Takashi Tomokiyo and Matthew Hurst. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment*, pages 33–40, 2003. (cited on Page 19)

Unesco. Unisist indexing principle sc.75/ws/58. 1975. (cited on Page 6)

Asahi Ushio, Federico Liberatore, and Jose Camacho-Collados. Back to the basics: a quantitative analysis of statistical and graph-based term weighting schemes for keyword extraction. *arXiv preprint arXiv:2104.08028*, 2021. (cited on Page 43)

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. (cited on Page 12)

Xiaojun Wan and Jianguo Xiao. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, volume 8, pages 855–860, 2008. (cited on Page 44)

H Wang, Z Lei, X Zhang, B Zhou, and J Peng. Machine learning basics. *Deep learning*, pages 98–164, 2016. (cited on Page 8)

Rui Wang, Wei Liu, and Chris McDonald. Corpus-independent generic keyphrase extraction using word embedding vectors. In *Software engineering research conference*, volume 39, pages 1–8, 2014. (cited on Page 19)

Rui Wang, Wei Liu, and Chris McDonald. Using word embeddings to enhance keyword identification for scientific publications. In *Australasian Database Conference*, pages 257–268. Springer, 2015. (cited on Page 20)

Christian Wartena, Rogier Brussee, and Wout Slakhorst. Keyword extraction using word co-occurrence. In *2010 workshops on database and expert systems applications*, pages 54–58. IEEE, 2010. (cited on Page 9)

Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 254–255, 1999. (cited on Page 21)

Wenpu Xing and Ali Ghorbani. Weighted pagerank algorithm. In *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.*, pages 305–314. IEEE, 2004. (cited on Page 20)

Yan Ying, Tan Qingping, Xie Qinzheng, Zeng Ping, and Li Panpan. A graph-based approach of automatic keyphrase extraction. *Procedia Computer Science*, 107: 248–255, 2017. (cited on Page 38)

Benjamin Zapilko, Johann Schaible, Philipp Mayr, and Brigitte Mathiak. Thesoz: A skos representation of the thesaurus for the social sciences. *Semantic Web*, 4(3): 257–263, 2013. (cited on Page 31)

Zhi Zhou, Xiaojun Zou, Xueqiang Lv, and Junfeng Hu. Research on weighted complex network based keywords extraction. In *Workshop on Chinese Lexical Semantics*, pages 442–452. Springer, 2013. (cited on Page 8)

I herewith assure that I wrote the present thesis independently, that the thesis has not been partially or fully submitted as graded academic work and that I have used no other means than the ones indicated. I have indicated all parts of the work in which sources are used according to their wording or to their meaning.

Magdeburg, November 30, 2022