TU WIEN Informatics

Stimmungsanalyse von geclusterten COVID-19 Artikel zum Thema Maskenpflicht

Ein Vergleich mit User-Kommentaren

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Lukas Burtscher
Matrikelnummer 11925939

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Ass.in Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Mitwirkung: Projektass. Dipl.-Ing. Thomas Elmar Kolb, BSc

Wien, 16. Juni 2023

Lukas Burtscher	Julia Neidhardt

TU WIEN Informatics

Exploring the Sentiment Patterns of Clustered COVID-19 News Articles on Mask Requirements

A Comparison with User Comments

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Business Informatics

by

Lukas Burtscher
Registration Number 11925939

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass.in Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Assistance: Projektass. Dipl.-Ing. Thomas Elmar Kolb, BSc

Vienna, 16th June, 2023

Lukas Burtscher	Julia Neidhardt

Erklärung zur Verfassung der Arbeit

Lukas Burtscher

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 16. Juni 2023

Lukas Burtscher

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Bachelorarbeit unterstützt und motiviert haben.

Zuerst gebührt mein Dank Frau Julia Neidhardt, die meine Bachelorarbeit betreut und begutachtet hat. Für die hilfreichen Anregungen und die konstruktive Kritik bei der Erstellung dieser Arbeit möchte ich mich herzlich bedanken.

Ich bedanke mich bei Thomas Elmar Kolb für die Mitbetreuung meiner Arbeit und dem Bereitstellen der technischen Infrastruktur.

Abschließend möchte ich mich bei meinen Eltern bedanken, welche mir mein Studium durch ihre Unterstützung ermöglicht haben und stets ein offenes Ohr für mich hatten.

Lukas Burtscher

Wien, 16.06.2023

Kurzfassung

Während die COVID-19-Pandemie weiterhin die Welt heimsucht, ist es für EntscheidungsträgerInnen und öffentliche GesundheitsbeamtInnen entscheidend, die öffentliche Stimmung und Einstellung gegenüber den Maßnahmen zur Kontrolle ihrer Ausbreitung zu verstehen. Diese Arbeit untersucht die Entwicklung der Stimmung bezüglich COVID-19-Maskenanforderungen von Januar 2020 bis Dezember 2021 durch Analyse und Clustern von Nachrichtenartikeln und usergenerierten Kommentaren. Derzeitige Studien verwenden unsupervised Clustering-Methoden, einschließlich Latent Dirichlet Allocation, K-Means-Clustering und Word-Embedding-Techniken, um Muster in Daten zu identifizieren. In dieser arbeit verwenden wir für das Clustering Word2Vec-Word-Embedding, TF-IDF und K-Means-Algorithmen. Die Vorverarbeitungsschritte umfassen Rechtschreibkorrektur, Emoticon-Behandlung und Lemmatisierung mit *SpaCy*. Diese Arbeit analysiert COVID-19 Maskenanforderungsdaten von *Der Standard*, einem österreichischen Zeitungsverlag, und zeigt einen negativen Stimmungstrend in politischen Nachrichtenartikeln sowie eine Polarisierung im Widerstand der Öffentlichkeit gegen Regierungsrichtlinien auf. Diese Studie bietet Erkenntnisse, die EntscheidungsträgerInnen und öffentliche GesundheitsbeamtInnen über die Einstellungen der Öffentlichkeit gegenüber Maskenanforderungen informieren können, um verbesserte Kommunikationsstrategien und eine bessere Kontrolle der Pandemie zu ermöglichen.

Abstract

As the COVID-19 pandemic continues to ravage the world, understanding public sentiment and attitudes towards the measures taken to control its spread is crucial for policymakers and public health officials. This thesis investigates the evolution of sentiment regarding COVID-19 mask requirements from January 2020 to December 2021 by analyzing news articles and user-generated comments. Current research employs unsupervised clustering methods, including Latent Dirichlet Allocation, K-Means clustering, and word embedding techniques, to identify patterns in the data. The cluster building involves Word2Vec word embedding, TF-IDF, and K-Means algorithms, while the preprocessing stage includes spell correction, emoticon handling, and lemmatization using *SpaCy*. The research analyzes COVID-19 mask requirement data from *Der Standard*, an Austrian newspaper publisher, revealing a negative sentiment trend in political written news articles and polarization in the public's opponents to government guidelines. This thesis offers insights that can inform policymakers and public health officials about the public's attitudes towards mask mandates, allowing for improved communication strategies and better control of the pandemic.

Contents

Kurzfassung **ix**

Abstract **xi**

Contents **xiii**

1 Introduction **1**
1.1 Motivation and Problem Statement 1
1.2 Related Work 3

2 Data Acquisition **5**

3 Text Preprocessing **7**
3.1 Extract Text Data From HTML Document Files 7
3.2 Stemming and Lemmatization 7
3.3 Building Phrases 8
3.4 Spell Correction 10
3.5 Emoticon Handling 11

4 Feature Extraction and Data Preparation **13**
4.1 Word2Vec 13
4.2 Data Preparation 16

5 Clustering **19**
5.1 Manual Labeling 22
5.2 Classified Articles 26

6 Sentiment Calculation **29**
6.1 Dealing With Sub Comments 30

7 Comparison Analysis **33**
7.1 Exploration 33
7.2 Conclusion 37
7.3 Discussion and Future Work 39

7.4 Repository 40

List of Figures **41**

List of Tables **43**

Bibliography **47**

CHAPTER 1

Introduction

In December 2019, patients in Wuhan, China reported severe respiratory infections, which were later linked to working in wet markets (Huang Chaolin, 2020). The World Health Organization initially denied human-to-human transmission but declared COVID-19 a Public Health Emergency of International Concern in January 2020 (WHO, 2020). COVID-19 stands for "Coronavirus Disease 2019." It is a highly contagious illness caused by a novel coronavirus, known as SARS-CoV-2. The term COVID-19 was coined by the World Health Organization (WHO) to describe the disease that emerged in late 2019 in Wuhan, China, and rapidly spread worldwide (Cucinotta D, 2020; WHO, 2020). The virus quickly spread globally, with over 858,000 cases and 47,192 deaths reported by the end of March 2020. The virus has affected over 166 countries with a case fatality rate that can range up to 9.26% (Khafaie Morteza Abdullatif, 2020). Isolation and self-quarantine are recommended to stop the spread of the pandemic, as demonstrated by China's lockdown. Other countries, including India, have also implemented lockdowns to control the spread of the virus (Dubey, 2021).

1.1 Motivation and Problem Statement

In Austria the COVID-19 pandemic has had a significant impact on its citizens, resulting in a range of measures such as mask requirements[1], lockdowns (Austria, 2020), and compulsory vaccinations[2]. These events have polarized public opinion, resulting in extensive commentary on social media. Policymakers and public health officials should be motivated to understand public opinion and reactions towards these measures to adjust their strategies accordingly (Andrea Ceron, 2015). One effective approach to achieve this goal is identifying sentiment trends. Analyzing the sentiment patterns of

[1] https://orf.at/stories/3160106/ last accessed 13.04.2023
[2] https://wien.orf.at/stories/3106677/ last accessed 13.04.2023

news articles and user comments over the last two years can provide valuable insights into the evolution of public opinion towards COVID-19 and related measures in Austria.

This thesis aims to identify sentiment trends and interesting events by analyzing a subset of COVID-19 articles related to the mask requirement policy in Austria. To achieve this, Natural Language Processing (NLP) strategies were applied to cluster articles into subgroups, and the sentiment scores of user comments were compared with the sentiment of the articles and the sentiment of different subgroups at specific time points. NLP is a field of artificial intelligence that focuses on enabling computers to understand, interpret, and generate human language in a way that facilitates human-computer interaction and information extraction (Diksha Khurana, 2022). The identification of such trends and events can provide valuable insights into public opinion and attitudes towards the mask requirement policy in Austria. The NLP techniques used in this study provide a tool for analyzing large volumes of text data, which can reveal nuanced patterns and insights that are not easily discernable through traditional data analysis techniques.

Performing NLP on text data presents several technical challenges. One of the primary challenge is ensuring the accuracy of the text data through various preprocessing techniques such as emoticon handling and spell correction (Diksha Khurana, 2022). Additionally, embedded text data often contains a large number of numerical features, making it challenging to analyze and extract meaningful information. Dimensionality reduction techniques such as principal component analysis (PCA) can help to address this challenge by reducing the number of features while preserving the most significant information (Rodionova, 2021). Another important challenge is identifying the underlying patterns and structures within the text data, which can be accomplished through clustering techniques (Wenchuan Mu, 2022). Finally, accurately determining the sentiment of text data poses a significant challenge, as it can be influenced by factors such as sarcasm, irony, and cultural differences (Diksha Khurana, 2022).

This thesis utilized a data set of anonymized user comments posted directly on the news forum of the Austrian newspaper company *Standard Verlagsgesellschaft m.b.H (Der Standard)* from February 2020 until November 2021. In contrast to *Facebook*, users are more anonymous on the forum because they only have to register with an e-mail and set a community nickname to comment. Additionally, the variety and number of news posts are greater, and users do not only post under the snippet of an article but directly view the whole text.

To extract sentiment patterns for different COVID-19 subgroups, NLP techniques are employed. This thesis presents a comprehensive preprocessing pipeline, including tokenization, stopword removal, and lemmatization. Additionally, emoticons need to be handled and converted into sentiment for the user comments. Spell correction is also a significant challenge due to the unique Austrian accent, which differs from the German language (Muhr, 2008). To group similar COVID-19 articles based on the words and phrases used in the text, n-grams are built and word embedding is applied for numerical feature extraction. A cluster analysis is performed using appropriate techniques to determine the appropriate number of topics for article separation. A dictionary-based

sentiment approach is then utilized to extract the sentiment of user comments in each topic group. The sentiment of the final discussion is also influenced by comment voting and sub-conversations. Thus, this thesis tackles various computational challenges to uncover the sentiment patterns in the end.

It address the following research questions:

- **RQ1:** Which preprocessing techniques are suitable when clustering text data and performing sentiment analysis?
- **RQ2:** Based on the textual content of each article, what methods can be employed to categorize news articles into distinct topics?
- **RQ3:** To what extent can sentiment scores be computed for user comments while taking into account both user ratings and sub-comments?
- **RQ4:** What trends and patterns in sentiment can be investigated by comparing different groups of articles over a specific time frame?

1.2 Related Work

Sentiment analysis and opinion mining have recently gained popularity for examining COVID-19 (Ponmani, 2022; Shoaib, 2021; Dubey, 2021; Khafaie Morteza Abdullatif, 2020). However, the majority of research has concentrated on analyzing social media data related to COVID-19, rather than officially released news data and user comments. In Akash Dutt Dubey's work (Dubey, 2021) word clouds were used to represent the sentiment of a country based on *Twitter* postings. The findings revealed that countries such as Belgium, India, and Australia had positive sentiments in their COVID-19-related tweets, while China exhibited negative sentiments. Moreover, analyzing the word clouds of various countries revealed that individuals used different emotions when tweeting about pandemic-related topics such as *Death, Quarantine, Hope, Stay Safe, Government, Political, Fight, and Masks.*
This thesis focuses on clustering and analysing the sentiment in each cluster. For this, the work of Dylan Castillo (Castillo, 2021) provides information about clustering text data using word embedding techniques. Here the focus lies on text preprocessing and a Word2Vec implementation in Python. The focus of Ahmed Md Shoaib's study (Shoaib, 2021) is on identifying sentiment dynamics within clustered groups of *Twitter* users regarding trending COVID-19 topics. The paper introduces a model that aims to identify the dynamic sentiment of users towards top-k trending sub-topics related to COVID-19, while also identifying the top active users based on their involvement scores in those trending topics. The approach involves preprocessing the postings and applying a topic modelling method to calculate the trending topics and users' involvement scores. To perform topic modelling, the paper utilized a Latent Dirichlet Allocation (LDA) method (Zhao WX, 2011). For sentiment calculation, a rule-based lexicon is employed. Min Shi and Jianxun Liu (Min Shi, 2017) conducted a comparison between LDA and a word

embedding technique with K-Means clustering. Based on the findings, an advanced WE-LDA model was proposed, which incorporated high-quality word embeddings to enhance the performance of the topic model. The results showed that the utilization of word embeddings led to a significant improvement in the clustering accuracy of their proposed approach. At present, only a limited number of studies have investigated the sentiment trends related to the requirement of masks during the COVID-19 pandemic. Nonetheless, Ponmani K. (Ponmani, 2022) discussed the significance of having a vaccine for COVID-19 and how individuals' decisions regarding vaccination can be influenced by their sentiments. To analyze sentiment, a dictionary-based approach was used along with a random forest classifier. A straightforward Term Frequency-Inverse Document Frequency (TF-IDF) method was employed for feature extraction. TF-IDF is a numerical representation used in information retrieval and text mining to assess the importance of a term in a document by considering its frequency in the document and inversely weighing it based on its frequency across the entire corpus (Aizawa, 2002). More on that in Section 5

This thesis prioritizes simplicity and interpretability, so it avoids NLP techniques that utilize neural networks (Janiesch, 2021). K-Means clustering and Word2Vec embedding are preferred over deep learning models because they are more explainable (Meyer, 2016). Additionally, a dictionary-based approach is chosen for sentiment analysis, based on recommendations from (Dubey, 2021; Janiesch, 2021).

CHAPTER 2

Data Acquisition

To gather textual data pertaining to COVID-19 pandemic and the mask mandate process in Austria, the database of Austria's Newspaper *Der Standard* is used. *Der Standard* is an independent Austrian daily newspaper founded in 1988 by Oskar Bronner with a 50% participation by the German Springer group. Nowadays the Standard Medien AG is the 100% owner of the Standard Verlagsgesellschaft m.b.H. and derStandard.at GmbH[1]. In 2011, the newspaper had a daily circulation of 72,700. In 1995, it became the first German-speaking daily newspaper to have an online presence[2].

This chapter provides an overview of the data used in this research thesis. The productive database of *Der Standard* is accessed to explore the released print media articles, their user comments, and user data. The first step is to query over all related news articles.

The HTML content of web pages is stored in a *Content* table, along with their corresponding ID. In addition, the *Content* table contains further information about the content such as author, topic, and release date.

To reduce the amount of data to a computable number, more constraints are applied to obtain fewer but more targeted COVID-19 related articles. In addition to searching for articles where "Maskenpflicht" was mentioned to focus on the mask requirement situation during the COVID-19 pandemic since 2020, another constraint is used to select only articles where the *ressort name* is "Coronavirus" to focus specifically on COVID-19 related articles. This study primarily examines articles that are published without a specified author. Consequently, the majority of the news content relies on data collected from the *Austrian Presse Agentur* (APA), rather than personal opinions. This serves to compare supposedly objectively written articles that contain little own opinion with user comments.

[1] https://about.derstandard.at/unternehmen/standard-medien-ag/ last accessed 04.04.2023

[2] http://www.demokratiezentrum.org/bildung/ressourcen/lexikon/der-standard/ last accessed 04.04.2023

Query 2.1 represents the logic discussed:

Algorithm 2.1: Retrieve Top 10 Articles Related to COVID-19 and Mask Mandates

```
1 SELECT TOP (10) t.,c.author, c.RessortName, c.source, c.sourceCategory,
    c.ObjectType, c.publishingDate, c.titletext
2 FROM ContentArtikel t, Content c
3 WHERE t.documentXML.exist('//[text()[contains(.,"Corona")]]') = 1
4 AND t.documentXML.exist('//*[text()[contains(.,"Maskenpflicht")]]') = 1
5 AND c.ID_GodotObject = t.ID_GodotObject
6 AND c.RessortName = 'Coronavirus'
7 AND c.author = '(keine Angabe)'
```

The subset used for this analysis comprises 103 COVID-19 articles and 421,572 user comments. A subset is selected because the mask requirement has affected many groups in society, making it a suitable subset for conducting a sentiment analysis without having to consider all COVID-19 articles. The objective of this thesis is to compare user sentiment data with news articles that are written without personal opinions. To achieve this, we will focus solely on articles that were published by the newspaper itself.

However, some of the constraints were imposed due to computational limitations. The multiprocessing preprocessing pipeline took 33.7 hours to process the 421,572 user comments on 40 available logical processors. The analysis was performed using an Intel CPU E5-2630 with 20 cores and a base speed of 2.20 GHz.

CHAPTER 3

Text Preprocessing

As previously discussed in the introduction, the aim is to understand the sentiments and opinions expressed in the text data through the use of NLP and unsupervised clustering. To achieve this, the raw text data needs to be prepared. In this section, this will be explained in detail.

3.1 Extract Text Data From HTML Document Files

The HTML structure is a data format that encompasses meta-information regarding the presentation of text on a web page. To isolate the text from the HTML string object, the *BeautifulSoap* (Richardson, 2019) Python module is utilized. With *BeautifulSoap*, the majority of text can be extracted from the HTML tags. However, not all tags can be removed, so a more straightforward approach is employed by utilizing a regular expression to remove the HTML tag characters.

BeautifulSoap inserts spaces between the text and tags, enabling words to be separated. For instance, *HelloWorld* should be recognized as "Hello World" not as "HelloWorld". However, in some instances, the third-party library cannot correctly place the spaces. To remedy this, the Python *search()* and *split()* functions are used to insert a space character between two German nouns, as *BeautifulSoap* is designed to work with English syntax, where nouns are not capitalized.

3.2 Stemming and Lemmatization

Stemming and lemmatization are important techniques in NLP preprocessing as they can help standardize the text data by reducing each word to its core form, thereby reducing the dimensionality of the text data and removing some of the noise. This, in turn, can lead

to more accurate and effective feature extraction and modeling (Vimala Balakrishnan, 2014).

Stemming involves removing the suffixes (and in some cases prefixes) of words to arrive at a common base form. For example, the stemming of the words "running", "runner", and "ran" would result in the common base form "run" (Vimala Balakrishnan, 2014).

Lemmatization, on the other hand, goes a step further and reduces a word to its base form using vocabulary and morphological analysis, which takes into account the context and part of speech of the word. For example, the lemma of the word "running" would be "run" (Vimala Balakrishnan, 2014). In another example of the word "studies", lemmatization would identify the information of the word "studies" as to whether it is a singular or third person point of view. In this example, "studies" is a present tense of the verb "study," and thus the resulting lemma would be "study" (Rio Pramana, 2022).

This thesis uses the well-known *SpaCy* (Honnibal, 2014) library for lemmatizing texts in Python. To load the german vocabulary in a *SpaCy* NLP language model this thesis chooses the *de_core_news_md* (Honnibal, 2014) package by the *spacy.load("de_core_news_md")* command. The model offers 500.000 keys and 20.000 unique word vectors for the german language. In *SpaCy*, the lemmatization of German words is performed using the morphological analysis of the words and their Part-of-Speech (POS) tags. The lemmatizer uses the rule-based approach to identify the base form of a word, which is known as the lemma. For example, in German, the words "laufen" (in English: run) and "gelaufen" (in English: ran) have the same lemma "laufen" (Honnibal, 2014). Morphological analysis is the process of identifying the root form of a word and its grammatical structure, including its inflection, tense, and other features. In other words, it involves breaking down a word into its constituent parts and analyzing the meaning and function of each part (Wartena, 2019). Part-of-Speech (POS) tags, on the other hand, are labels assigned to words to indicate their grammatical category, such as nouns, verbs, adjectives, adverbs, etc. This is used to identify the role of a word in a sentence and to better understand the meaning of a text (Wartena, 2019).

3.3 Building Phrases

Currently, each article is represented as a list of lemmatized tokens, each representing one word. To find more specific topics this thesis uses *bigrams* and *trigrams* to build common phrases in the text data.

Bigrams and *trigrams* are two types of n-grams, which are sequences of n consecutive words in a text. A *bigram* is a sequence of two words and a *trigram* is a sequence of three words (Wenchuan Mu, 2022). These n-grams provide valuable information about the relationship between words in a text and the context in which they are used. By analyzing the frequency of *bigrams* and *trigrams* in a corpus of COVID-19 news articles, it can be identified which combinations of words are frequently used together in relation to

the pandemic. This information can then be used to group articles into related topics and provide a better understanding of the most commonly discussed aspects of the pandemic.

For example, consider the following sentence:

"Austria requires masks in all public places to slow the spread of the coronavirus."

After tokenization, the sentence can be represented as the following list of tokens:

["Austria", "requires", "masks", "in", "all", "public", "places", "to", "slow", "the", "spread", "of", "the", "coronavirus"]

To extract *bigrams*, they concatenate pairs of consecutive tokens like:

["Austria requires", "requires masks", "masks in",...]

for *trigrams*, they consider groups of three consecutive tokens like:

["Austria requires masks", "requires masks in", "masks in all",...]

The choice of n-gram size depends on the desired level of granularity for the analysis. Larger values of n result in more specific n-grams that capture more information about the relationship between words in a text, while smaller values of n result in more general n-grams that capture less information.

The *trigram* model was found to achieve a slightly higher accuracy score in a comparative study (Tiffani, 2020). Given that the use of *trigrams* is common in word clustering (Sven Martin, 1997; David Johnson, 2006), this approach is also employed in this thesis. Specifically, a *trigram* model is constructed based on a simpler *bigram* model in order to capture common *bigrams* and any further *trigrams* that may occur. This is in line with suggestions made in a topic modelling article (CR, 2020).

This thesis uses the *gensim.models.Phrases* (Rehurek, 2020) class to create a *bigram* model. The model is created with a minimum count of five, meaning that only *bigrams* that occur at least five times in the text will be considered. The threshold of 100 means that *bigrams* that have a score greater than 100 will be considered, resulting in fewer hits. The model can then be used to predict phrases in a new sentence (Rehurek, 2020). Based on the found *bigrams* further *trigrams* are constructed by training another model with the phrases found for each token in the *bigram* model.

The function *construct_trigrams()* is defined. It takes a list of text documents as input, and returns a list of the same documents with *trigrams* constructed using the *bigram* and *trigram* models. The *trigrams* are constructed by first applying the *bigram* model and then the *trigram* model to each document.

The final result of this code is a list of documents represented by the *data_words* variable, where *bigrams* and *trigrams* have been added to the original tokens.

Building n-grams after lemmatization is a common practice in NLP and is considered to be superior to building n-grams before lemmatization for several reasons. Firstly, lemmatization allows for a more accurate representation of the underlying semantic

meaning of the words in a text, as it maps inflected forms of words to their base form (Vimala Balakrishnan, 2014). This improved semantic representation of words leads to a more meaningful representation of the text as a whole, which is useful in NLP tasks such as text classification, sentiment analysis, and language modelling (Rio Pramana, 2022).

Secondly, lemmatization reduces the number of unique words in a text by mapping inflected forms to their base form. This reduction in the number of unique words results in a smaller n-gram vocabulary, which can lead to improved computational efficiency when building n-grams (Castillo, 2021). Additionally, the reduced dimensionality of the n-gram vocabulary can result in improved n-gram frequency, as the frequency of n-grams is more likely to accurately represent the underlying meaning of the text.

In conclusion, building n-grams after lemmatization is considered to be a best practice in NLP, as it leads to improved semantic representation, reduced dimensionality, improved n-gram frequency, and better generalization to new texts (David Johnson, 2006).

3.4 Spell Correction

A spell checker is an important component in preprocessing text data for sentiment analysis as it helps improve the accuracy and reliability of the analysis (James Kavanagh, 2023). When working with user-generated comment text data, it's common for errors, such as typos or misspelled words, to be present. These errors can negatively impact the results of the sentiment analysis by leading to incorrect or ambiguous interpretations of the text (Singh, 2019).

By incorporating a spell checker, these errors can be corrected, resulting in clearer, more accurate text data for the sentiment analysis to process (Singh, 2019). This, in turn, leads to improved accuracy and reliability of the analysis and the results it produces (James Kavanagh, 2023).

Incorporating a spell checker into the preprocessing step of the sentiment analysis process is a professional way of ensuring that the data being analyzed is of the highest quality and that the analysis results are accurate and meaningful. Additionally, in the context of sentiment analysis, it's important to consider regional dialects and variations in language, as they can also negatively impact the results (Singh, 2019). For example, if a spell checker is designed to recognize German words, it may not accurately identify words in an Austrian dialect that deviate from standard German spelling. This could lead to incorrect spellings being corrected and valuable information being lost.

The Python spellchecker *psyspellchecker* (Barrus, 2023) is employed, because it also corrected some easy dialect expressions to high German, for example, "wos" has been corrected to "was".

Psyspellchecker is a Python spell-checking library based on Peter Norvig's algorithm[1]. It uses the Levenshtein Distance to calculate the difference between words and find spelling

[1] https://norvig.com/spell-correct.html last accessed 20.06.2023

mistakes within an edit distance of two (Barrus, 2023). The library then compares these permutations to a word frequency list and suggests the most likely correction based on the frequency of the words. This library supports multiple languages including English, Spanish, German, French, and Portuguese and is compatible with Python three. The Levenshtein Distance can be set to a maximum of two, but for longer words, it's recommended to use a distance of one for better results (Barrus, 2023). Since user data is processed, staying on the default distance of two (usual users won't write long words in their posting) is reasonable.

3.5 Emoticon Handling

The user comments included various emoticons as a means of conveying emotions. To aid in sentiment analysis at a later stage, the most commonly used emoticons are replaced with their corresponding textual descriptions.

Replacing emoticons with their textual descriptions in text preprocessing makes sense for several reasons. Firstly, standardization is achieved by this process. Emoticons can have different forms and representations across platforms and devices. For example, the same emoticon can be represented by different Unicode characters, or even as ASCII art. By replacing emoticons with their textual descriptions, the representation of emotions across all text inputs is standardized, making it easier to process and analyze (Dandannavar, 2019).

Secondly, improved text representation is another reason. Emoticons are not words, and they are not easily recognizable by NLP algorithms (Tripti Agrawal, 2019). By replacing emoticons with their textual descriptions, the representation of emotions in text is improved, making it easier for NLP algorithms to understand and process them (Dandannavar, 2019).

Finally, readability is improved as well. Emoticons can be difficult to understand, especially for those who are not familiar with them. Replacing emoticons with their textual descriptions improves the readability of text, making it easier for humans to understand the emotions expressed in the text (Tripti Agrawal, 2019).

For the replacement process, the inspiration comes from the *emot*[2] library-published Unicode dictionary. To start, a subset of 2000 comments is used to check in the dictionary for the presence of any emoticons. A self-constructed dictionary is used, to pair common emoticons with their German translations. For example, we replace ":D" with the German word for laugh. Finally, a simple loop replaces all emoticons in each user comment with the corresponding German descriptions. In the user comment preprocessing pipeline, this step happens right in the beginning followed by the misspelled correction.

[2]https://github.com/NeelShah18/emot/blob/master/emot/emo_unicode.py last accessed 03.05.2023

CHAPTER 4

Feature Extraction and Data Preparation

This section is about, how to use word embeddings to extract numerical features out of COVID-19 news article text data and how to transform the features to prepare the topic clustering step.

Word embedding refers to a feature learning method that transforms the vocabulary words into low-dimensional vectors of continuous real numbers (Zhang, 2018). This technique creates a distributional vector representation of words, also known as a semantic vector space, which captures the semantic (meaning) and syntactic (structure) information of words based on their contextual usage.

In the field of NLP, the two widely used models for generating word embeddings are Word2Vec and Global Vectors for Words Representation (GloVe) (Mikolov, 2013; Pennington, 2014). Word2Vec, developed by Mikolov et al. (Mikolov, 2013), is a shallow neural network-based technique that learns word embeddings. On the other hand, GloVe, proposed by Pennington et al. (Barhoumi, 2017), is an unsupervised learning algorithm that performs exceptionally well in preserving context. Both techniques have demonstrated effectiveness in various NLP applications (Mikolov, 2013; Pennington, 2014; Sara Elshobaky, 2018; Marwa Naili, 2017). However, in the context of sentiment analysis, Word2Vec is more commonly used as it has shown superior performance (Sara Elshobaky, 2018; Marwa Naili, 2017).

4.1 Word2Vec

Word2Vec creates dense, low-dimensional vectors that capture the semantic and syntactic relationships between words in a corpus. These vectors can then be used as input to

a variety of machine learning algorithms, such as clustering, to identify patterns and relationships within the data (Meyer, 2016).

In the case of news articles, Word2Vec can capture the overall meaning and context of the words in the articles, as well as relationships between the words themselves. For example, words that appear frequently in similar contexts (such as "politics" and "government") will be represented by similar vectors, and words that are semantically related (such as "apple" and "fruit") will be positioned close to each other in the embedding space (Tomas Mikolov, 2013).

These vectors are able to capture the semantic meaning of words effectively. For example, consider the following analogy "Night is to day as right to left". It turns out that:

$$v_{day} - v_{nighht} + v_{right} \approx v_{left}$$

where $v_{day}, v_{night}, v_{right}, v_{left}$ are the word vectors of the analogy (Meyer, 2016). The two main types of Word2Vec models are CBOW (Continuous Bag of Words) and Skip-Gram. CBOW predicts a word based on its context, while Skip-Gram predicts the context based on a given word (Tomas Mikolov, 2013).

CBOW is a type of Word2Vec model that uses a feed forward neural network to predict a target word based on a list of context words (Tomas Mikolov, 2013). Given the phrase: "In school masks are mandatory", and the context *["In","school","are","mandatory"]*, the predicted target word would be "masks". Figure 4.1 shows an example architecture.

The Skip-Gram model is a neural network, which has a single hidden layer, that predicts the likelihood of a word being present when another word is present. The input word is taken and the model attempts to predict the surrounding context words with accuracy (Meyer, 2016). In contrast to the CBOW model, the Skip-Gram model focuses on learning and predicting context words. Studies have demonstrated that the quality of predictions improves as the number of word vectors increases, though it also leads to an increase in computational demands (Tomas Mikolov, 2013).

To perform this word embedding algorithm, the implementation provided in the *Gensim* (Rehurek, 2020) Python package is applied. It is a highly optimized and scalable implementation of the Word2Vec algorithm. The inspiration for this implementation comes from Dylan Castillo (Castillo, 2021). Firstly, Word2Vec is trained with the corpus of all article texts. Each article in the list is represented by a list of all tokens contained in that article.

In a further method, the list of documents gets vectorized and stores a vector per document in a NumPy array. The Word2Vec model returns a vector for each word, so an algorithm is needed to construct one vector per COVID-19 article. In the implementation by Dylan Castillo (Castillo, 2021), a simple mean calculation is utilized for all the word vectors in each document. However, in this approach, a weighted average calculation is used in order to give more emphasis to words that are deemed more relevant or significant in the document.

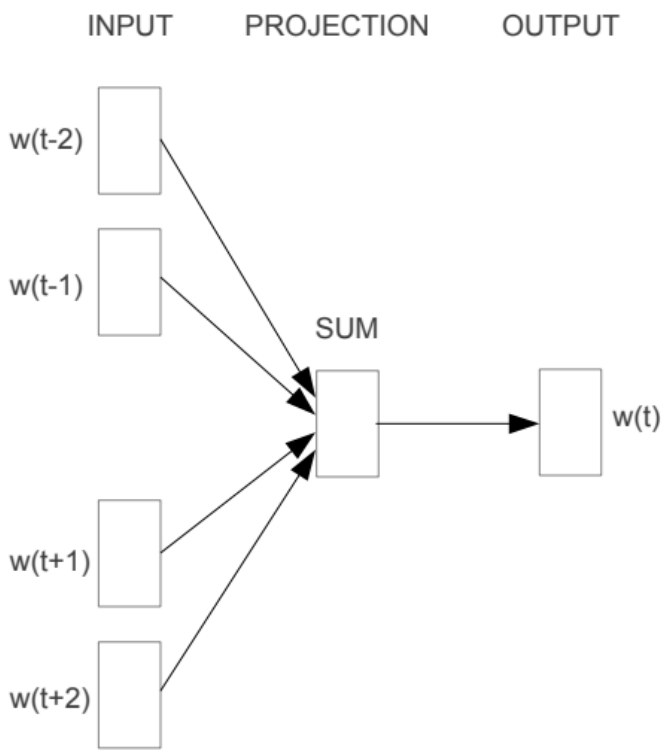

Figure 4.1: CBOW Architecture (Tomas Mikolov, 2013)

To achieve this, this thesis utilizes the *TfidfVectorizer*[1] from the *Sklearn* library, which calculates the mean TF-IDF score for each word across all documents. This TF-IDF score serves as a weight for each word vector that was extracted from the word embedding model. Each word vector x_i is multiplied by its respective weight w_i and then add all the weighted word vectors together. To obtain the final document vector, this sum is divided by the total sum of weights calculated during the process of investigating the words in the document.

$$WeightedMean = \frac{\sum_{i=1}^{n}(x_i * w_i)}{\sum_{i=1}^{n} w_i} \tag{4.1}$$

In conclusion, by utilizing a weighted average approach in creating the document vectors, this thesis gives greater importance to the words that are deemed more meaningful or significant in each document (Aizawa, 2002), resulting in a more nuanced and semantically rich representation. After that feature extraction, a vector with 100 numerical features for each of the 103 documents is obtained.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html last accessed 03.05.2023

4.2 Data Preparation

In this section, the numerical feature vectors will be prepared for unsupervised clustering.

As a first step, multicollinear columns must be dropped. This is an important step in data preprocessing because it helps to reduce the risk of overfitting and improve the interpretability of the model (Shrestha, 2020). Multicollinearity refers to the high correlation between two or more predictor variables in a statistical model. When predictor variables are highly correlated with one another, it can be difficult to determine the individual contribution of each variable to the outcome (Shrestha, 2020). Moreover, dropping multicollinear features can help to reduce the dimensionality of the data, making it easier for the KNN algorithm to process and run faster. The KNN algorithm uses distance measures to find the nearest neighbours, and high dimensionality can cause the algorithm to become computationally expensive. On the right of Figure 4.2 features that have a correlation $> |95\%|$ can be seen.

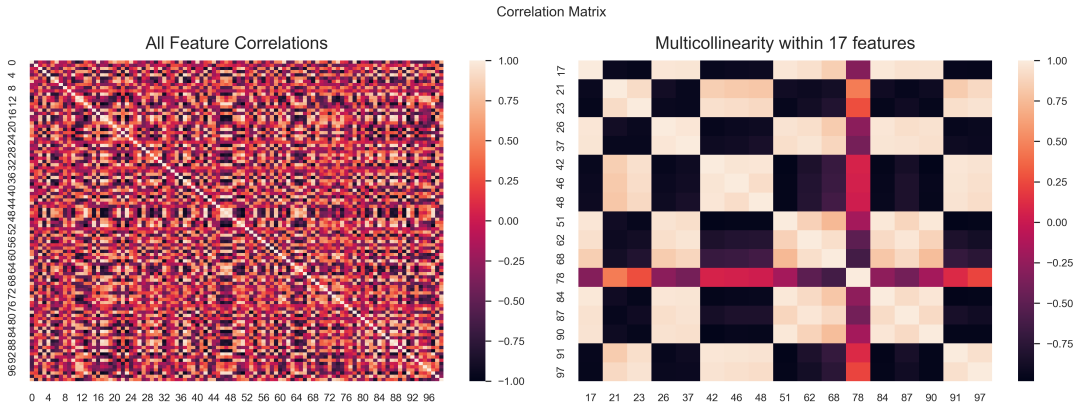

Figure 4.2: Correlation within the Document's Features

Secondly, Principal Component Analysis (PCA) is used in order to reduce the dimensions of the features. This is a statistical technique used to reduce the dimensionality of a dataset while preserving the most important information by transforming the data into a new set of uncorrelated variables called principal components. With PCA reduction it is able to achieve higher silhouette scores when performing the clustering algorithm (more in the next section).

The idea behind PCA is to reduce the number of features in a dataset while retaining as much information as possible. This is achieved by transforming the original features into a new set of features, called principal components, which are linear combinations of the original features. These principal components are ranked in order of importance, based on the amount of variance they explain in the original data (Rodionova, 2021). The matrix is decomposed using a formula:

$$X = T_A P_A^t + E \tag{4.2}$$

where A is the number of principal components, T and P are matrices containg scores and loadings for each component and E is the matrix of residuals (Rodionova, 2021). Before performing PCA it is important to scale and center the data because they ensure that all the features in the dataset are on a similar scale, which is crucial for accurate interpretation of the results.

Scaling is the process of transforming the values of a feature to a common scale (Rodionova, 2021). For example, if one feature has values that range from 0 to 1000, and another feature has values that range from 0 to 1, it would be appropriate to scale the values of both features to the same range, such as 0 to 1. This ensures that each feature has an equal impact on the calculation of the principal components. μ represents the mean of all x in one feature column, while σ represents the standard deviation in that column.

$$z_i = \frac{x_i - \mu}{\sigma} \tag{4.3}$$

Centering is the process of subtracting the mean of each feature from its values. This step is important because it helps to remove any systematic biases in the data (Rodionova, 2021). For example, if one feature has a mean value of 100, and another feature has a mean value of 0, subtracting the mean from each feature would result in both features having a mean value of 0. This helps to ensure that the first principal component truly represents the direction of maximum variance in the data, rather than being influenced by a systematic bias in one of the features. In Figure 4.3 the individual and cumulative explained variance explained per principal component can be seen. In this thesis, the first 5 components are chosen for dimensional reduction, to clearly explain more than 80% of the variance in the data.

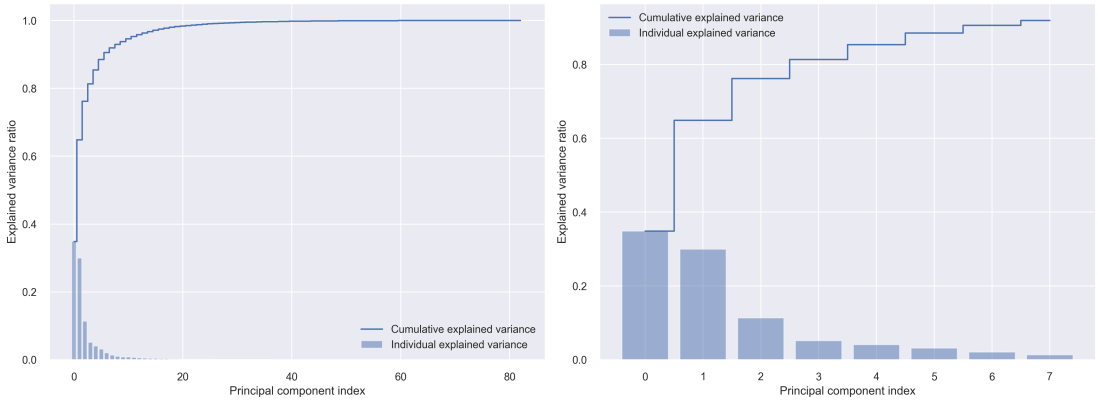

Figure 4.3: Explained Variance in PCA

CHAPTER 5

Clustering

Now K-Means is applied as an unsupervised clustering method to find similarities between the test documents. K-Means clustering is simple and intuitive to understand, making it easy to interpret the results. The algorithm works by finding the k nearest neighbours to each article, based on their word embeddings, and grouping similar articles together into clusters (Hartigan, 1979).

The sensitivity of this algorithm to skewed data is another reason why PCA is employed in the preprocessing stage. By standardizing the data before reducing its dimensions, it can now be immediately utilized for clustering without any issues. If the data is skewed, meaning that some features have much larger values than others, the mean of the data points will be heavily influenced by the large values, which can lead to biased clusters (Hartigan, 1979).

For example, if one feature has values ranging from 0 to 100, and another feature has values ranging from 0 to 1, K-Means clustering will place more weight on the feature with the larger values, which will cause the cluster assignments to be skewed towards that feature.

The K-Means clustering algorithm begins by randomly initializing the *centroids*, which serve as the centers of the clusters (Hartigan, 1979). It then computes the Euclidean Distance between each data point and the centroids to assign each data point to a cluster (Sinaga, 2020). After all the data points have been assigned, the algorithm adjusts the centroids to be situated at the center of these data points. This process is repeated until the centroids no longer move or experience only minimal movement (Hartigan, 1979). The distance between a data point and a centroid is calculated using the *Euclidean Distance* metric.

$$d(x, c) = \sqrt{\sum_{i=0}^{n} (x_i - c_i)^2} \tag{5.1}$$

Here, the calculation of the distance between a data point x and a centroid c is needed. The variable i represents a single dimension out of the total n dimensions, where n is the number of features used for clustering. To calculate the distance, the difference between the two points in each dimension is squared, sum the results, and then take the square root of this sum (Sinaga, 2020).

In K-Means it is needed to choose the number of clusters to be specified in advance. Determining the optimal number of clusters can be challenging and may require multiple runs of the algorithm (Humaira, 2020). In contrast to supervised learning, they do not have labelled data to train and test the model. Choosing the appropriate number of clusters is important for several reasons. First, it ensures that the clusters are clear and distinct, and the instances within a cluster are similar to each other, which is the primary goal of clustering (Humaira, 2020). Second, if the number of clusters is too high, there may not be enough data in each cluster to accurately represent the instances within it, which may result in overfitting. Furthermore, selecting too few clusters can result in unbalanced clusters, where one cluster has extremely more instances than the others.

With the silhouette score, the number of clusters can be determined. It takes into account both the within-cluster similarity and the between-cluster dissimilarity (Sukavanan Nanjundan, 2019). The silhouette score provides a simple, yet effective way to visualize the quality of the clustering solution. By calculating the silhouette score for different values of k (the number of clusters), the optimal number of clusters can be identified that produce the highest silhouette scores (Sukavanan Nanjundan, 2019). Here the *silhouette_score* [1] from *sklearn* is used for that.

$$ss = \frac{(b - a)}{max(a, b)} \tag{5.2}$$

It is calculated by determining the mean intra-cluster distance a and the mean nearest-cluster distance b for each sample. The silhouette coefficient for a sample is then obtained by subtracting the mean intra-cluster distance from the mean nearest-cluster distance and dividing the result by the maximum of the two, as shown in the formula above.

To understand the silhouette coefficient, it is important to note that b represents the distance between a sample and the closest cluster to which it does not belong. The silhouette coefficient provides a simple, yet effective way to evaluate the quality of the clustering solution by comparing the within-cluster similarity to the between-cluster dissimilarity (Sukavanan Nanjundan, 2019). The silhouette coefficient ranges from -1 to 1, with a high score indicating a good match between the sample and its own cluster, and a low score indicating a poor match (Sukavanan Nanjundan, 2019). Figure 5.1 shows the scores for the data frame with PCA preprocessing, while Figure 5.2 shows the score for each k cluster when fitting the model with the scaled data only. Considering this plot it can be clearly seen, that reducing the dimension to five principal components is a good idea. Nevertheless, both plots suggest a cluster size of three (when only the average score is important).

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html last accessed 03.05.2023

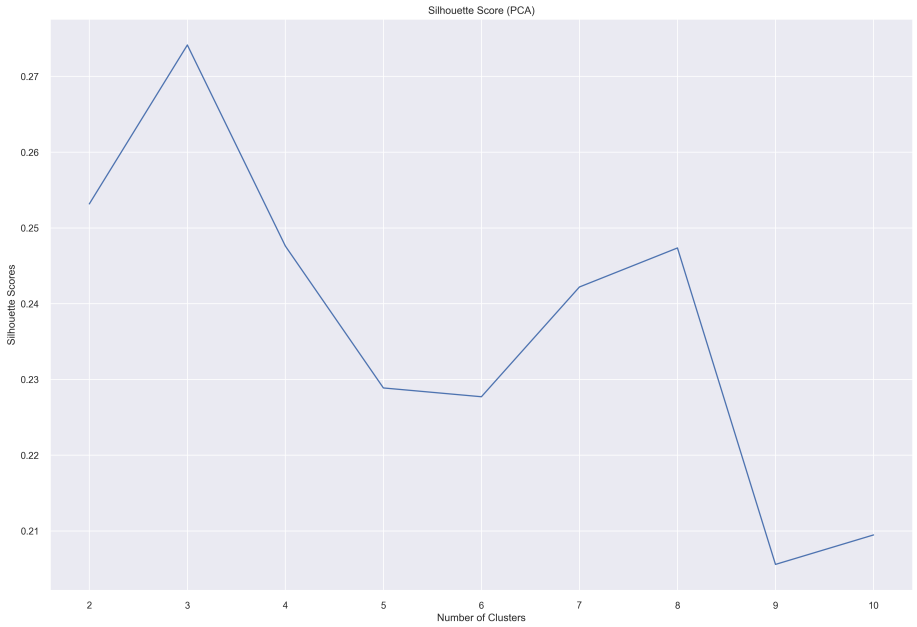

Figure 5.1: Silhouette Scores (with PCA)

Another, but way more superficial approach is the Elbow Method. It works by plotting the sum of squared distances (also known as inertia) of the data points to their closest cluster centre for different values of k, where k is the number of clusters (Humaira, 2020).

The resulting curve typically has a shape that resembles an elbow, where the "elbow point" is the value of k at which the decrease in sum of squared distances starts to level off (Humaira, 2020). This point represents the optimal number of clusters for the K-Means algorithm, as it strikes a balance between minimizing the sum of squared distances and preventing overfitting. Figure 5.3 shows these inertia scores for each k, but does not clearly provide a good selection for k. A point where the slope flattens out sharply is needed, but that point doesn't seem to be a definite one. Therefore the results of the silhouette analysis are more important.

In order to delve deeper into the results of the silhouette analysis, more sophisticated silhouette plots are created for the different values of k (for evaluation). In Figure 5.4, the cluster structure is displayed for cluster size three and eight, since those scored the highest in average. Each horizontal, color-coded line represents the silhouette coefficient value for a single instance within a cluster. The instances of each cluster are arranged in descending order based on this value. The vertical, red dotted line indicates the average silhouette score. For a desirable silhouette plot, clusters that are approximately equal in size (number of instances) and have enough values that reach at least the average

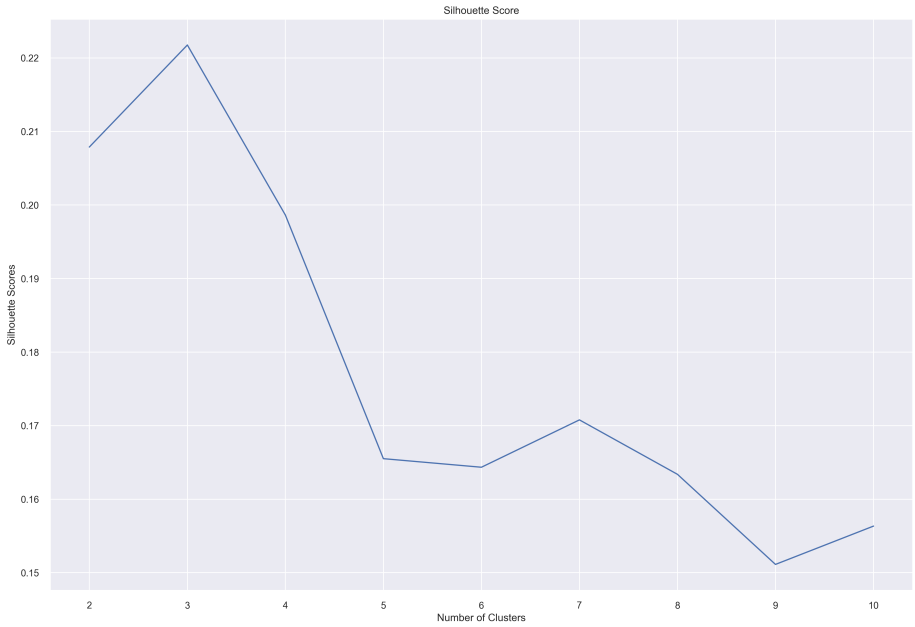

Figure 5.2: Silhouette Scores (without PCA)

silhouette score are perfect. To select the final cluster size, a look at the words in each cluster needs to be taken to decide. The following section will explain this approach.

5.1 Manual Labeling

This section will examine the words in each cluster manually, to get a better understanding of the opinions that are grouped together in each cluster and to see if the algorithm is able to separate well enough for not overlapping topics.

The unsupervised clustering algorithm can only identify patterns in the data based on the inputs and parameters that are provided. By manually examining the words in each cluster, they can leverage their own expertise and domain knowledge to identify patterns that may not be captured by the algorithm alone.

To find the most important words and phrases for each cluster, the TF-IDF score is applied.

TF-IDF is a widely used metric in the field of text analysis and information retrieval (Aizawa, 2002). It calculates the importance of a word in a document relative to a collection of documents, also known as a corpus (Aizawa, 2002). The TF in TF-IDF stands for term frequency, which is calculated as the number of occurrences of a word in

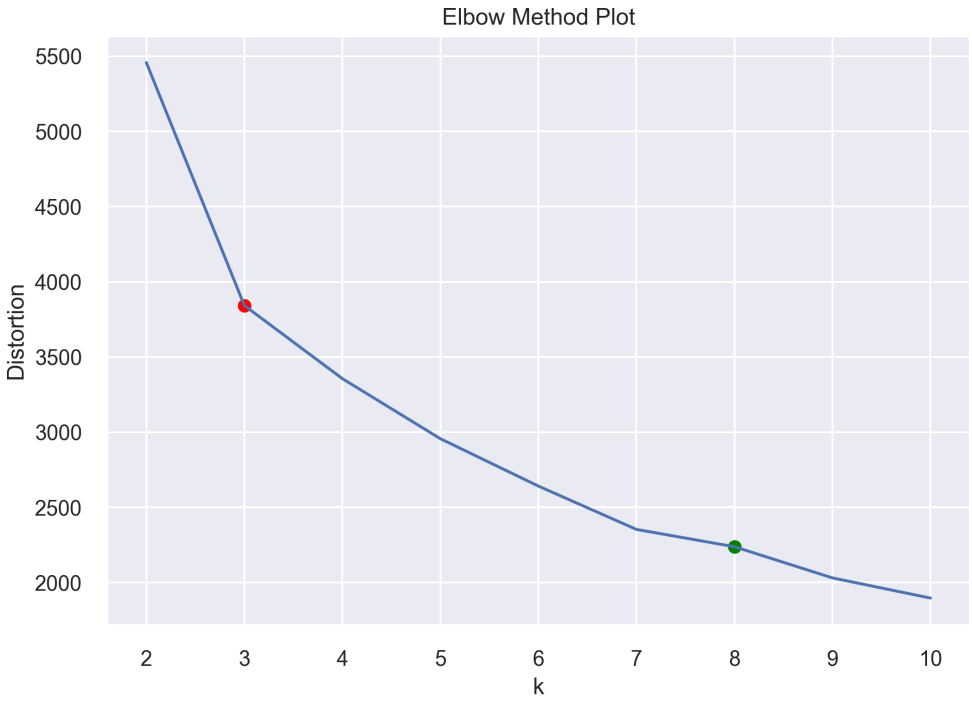

Figure 5.3: Elbow Method Plot

a document. On the other hand, the IDF in TF-IDF represents the inverse document frequency, which measures the rarity of a word across the entire corpus (Aizawa, 2002).

The fundamental concept behind TF-IDF is that words that are frequent within a document but rare across the entire corpus are considered to be more informative and relevant. By combining term frequency with inverse document frequency, TF-IDF provides a means to identify the most significant words in a document while taking into account their rarity in the corpus (Aizawa, 2002).

To compute the TF-IDF score of a word in a document, the term frequency is first calculated by counting the number of occurrences of the word. Then, the inverse document frequency is calculated by dividing the total number of documents in the corpus by the number of documents containing the word and taking the logarithm of this result. Finally, the TF-IDF score for a word in a document is obtained by multiplying the term frequency by the inverse document frequency (Aizawa, 2002).

To analyze the text clusters, a large text corpus for each cluster is constructed. Then, the *TfidfVectorizer* class from the *scikit-learn* library vectorizes the data. The vectorizer is fed with the merged text for each cluster by using the *fit_transform()* method. The result is a matrix containing the scores of each word in each cluster. This allows seeing which words are unique to each cluster and distinguishable from the text in other clusters.

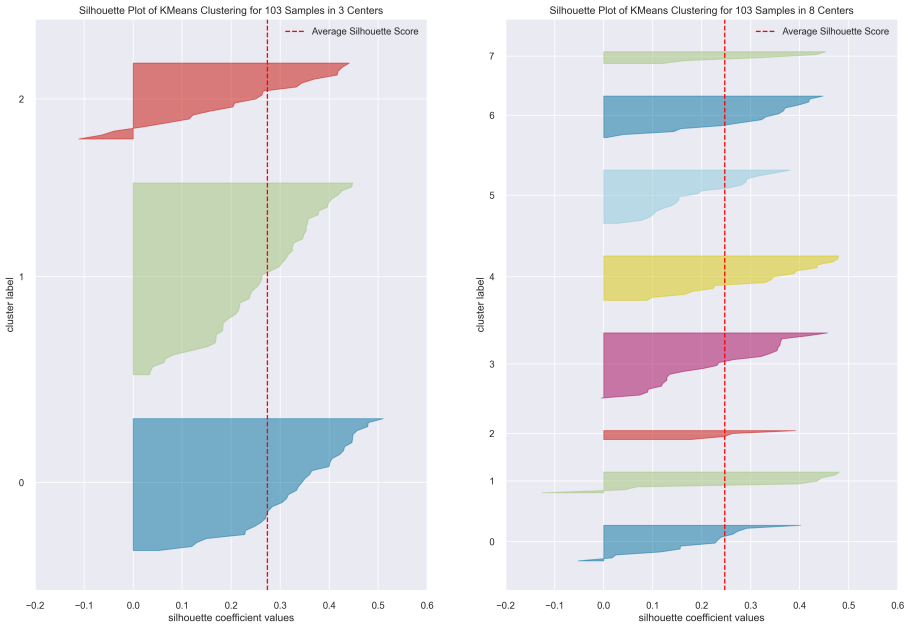

Figure 5.4: Silhouette Plot

Figure 5.5 depicts this information. To better compare the words, the axis is reformated so that the words are sorted in descending order based on their scores in each cluster. This makes it easier to compare the words and understand the results. The index column consists of words found in all documents, arranged alphabetically with numeric strings appearing first. The following eight columns correspond to individual clusters, displaying the TF-IDF score associated with each word in its respective cluster.

	index	0	1	2	3	4	5	6	7
0	0000	0.009875	0.0	0.0	0.000000	0.0	0.00000	0.0	0.0
1	006121	0.009875	0.0	0.0	0.000000	0.0	0.00000	0.0	0.0
2	006721	0.009875	0.0	0.0	0.000000	0.0	0.00000	0.0	0.0
3	024	0.000000	0.0	0.0	0.000000	0.0	0.00898	0.0	0.0
4	033	0.000000	0.0	0.0	0.008308	0.0	0.00000	0.0	0.0

Figure 5.5: Raw TF-IDF Scores Table (8 Clusters)

The thesis carries out this process for both the recommended 3-cluster size and the 8-cluster size. The most significant words and phrases are visible in the tables below. With the help of the most common words, a suitable category has been assigned. A list with more words, ordered by their score is in the linked repository.

Cluster	Mentioned Words	Labeled Topic
0	Maskenpflicht Verordnungen, geltende Regeln (Mask obligation regulations, applicable rules)	Guidelines & Measures
1	Demonstrationen, Polizei, Anzeigen, ÖVP (Demonstrations, police, reports, Austrian People's Party (ÖVP))	Police & Protests
2	FPÖ, Haimbuchner, Verfassungsgerichthof, Schule, Delta Variante (Freedom of Travel Party of Austria (FPÖ), Haimbuchner, Constitutional Court, school, Delta variant)	Opponents

Table 5.1: Words for Cluster Size: 3

Cluster	Mentioned Words	Labeled Topic
0	Veranstaltungen, Handel, Wirtschaft (Events, trade, economy)	Economy
1	Verfassungsgerichthof, Impfung, Verhältnismäß, Unverhältnismäßig (Constitutional Court, vaccination, proportionate, disproportionate)	Legal Discussion
2	Schramböck, Quarantäne, Grenzkontrollen und Reise (Schramböck, quarantine, border controls, and travel)	Freedom of Travel
3	Schule, Haimbuchner, FPÖ, Demonstration (School, Haimbuchner, FPÖ, Demonstrations)	Rebellion
4	Mückstein, ÖVP, Skifahren, Maßnahmen (Mückstein, ÖVP, skiing, measures)	Measures
5	Neuinfektionen (News Infections)	Information
6	Impfpflicht, Ungeimpfte, Geimpfte (Vaccination requirement, unvaccinated, vaccinated)	Vaccination
7	Demonstration, Polizei, Anzeigen, Festnahmen, Staatsgewalt (Demonstration, police, Reports, arrests, state authority)	Opponents

Table 5.2: Words for Cluster Size: 8

The average silhouette score for the three clusters is 0.274, while the eight clusters achieve a slightly lower score of 0.247. However, the larger cluster size led to the identification of more specific topics, as can be seen in Table 5.2. In contrast, the smaller cluster size resulted in more general article topics, which may not accurately describe each article.

Despite the benefit of more specific topics with the larger cluster size, some of the eight clusters are similar. For example, cluster three (rebellion) and cluster seven (opponents) both discuss the same events related to the requirement of wearing masks during the pandemic. As a result, there is consideration given to manually merging some of the clusters in order to reduce the cluster size. Specifically, the idea is to merge clusters containing similar words and phrases, which would reduce the eight clusters down to four. This allows for easier interpretation of the results and potentially simplifies the analysis.

The manual merging of the calculated clusters has the potential to introduce bias and distort the results of the analysis, particularly if more articles are added to the pool for clustering. Additionally, the average silhouette score of 0.247 is considered to be moderate, indicating that the clustering is not particularly well-separated (the maximum possible score is one) (Sukavanan Nanjundan, 2019). The score suggests some overlap between the clusters, but there is still enough separation to identify the eight meaningful clusters. Given this, this thesis decides not to manually merge the clusters and instead proceed with the eight clusters as they are defined by the algorithm.

5.2 Classified Articles

In their data export, *Der Standard* provides keywords and topics for each article. However, these officially released keywords are not suitable for the clustering process.

In the repository, there is a Jupyter notebook located in the *content/opinion* directory for analyzing the classification texts provided by *Der Standard*. Each article has a corresponding JSON object that contains various important topic objects in an array. These objects can describe a keyword, an event, a location, or something else that is defined under the key: *s*.

In the next step, any text values provided by *s* attribute is collected in a list to determine which words the editorial staff considers important. This is stored in a CSV file named *supervise_classified_content.csv*, located in the data folder. When examining the keywords and comparing them to the article, they are often not very accurate. The keywords appear suitable for SEO purposes but not for describing the words and phrases used in an article. Additionally, in this analysis, each text pertains to COVID-19 and the mask requirement. Thus, the subset of the same keywords is frequently identical across all articles, and many of the keywords describe the same topic. As a result, many of the keywords mentioned are not suitable for building clusters based on text expression.

Nonetheless, Table 5.3 displays the most frequent tags found for an unsupervised found cluster by the K-Mean algorithm.

The table only displays a selection of the keywords mentioned and does not accurately describe the topics discovered through unsupervised clustering for all clusters. It should be noted that these keywords include locations and persons, not just tags describing the plain text-corpus. A longer list of all merged tags is in the repository.

Cluster found by K-Means	Tags provided by DerStandard
0 (Economy)	Veranstaltungsbegrenzung, Gastronomie, Dienstleistung (Event restriction, gastronomy, service)
1 (Legal Discussion)	Infektionszahl, Veranstaltungsverbot, Verfassungsgerichtshof (Number of infections, event prohibition, Constitutional Court)
2 (Freedom of Travel)	Opernball, Reisewarnung, Reisen (Opera ball, travel warning, travel)
3 (Rebellion)	Impfung, Manfred Haimbuchner (Vaccination, Manfred Haimbuchner)
4 (Measures)	Mückstein, Skigebiet, Gewerkschaftsbund (Mückstein, ski resort, trade union federation)
5 (Information)	Impfstatus, Tagesüberblick, Testbeteiligung (Vaccination status, daily overview, testing participation)
6 (Vaccination)	Sperrstunde, Rudolf Anschober (Curfew, Rudolf Anschober)
7 (Opponents)	Weltverschwörung, Rechtsextrem, Protestzug (World conspiracy, far-right extremism, protest march)

Table 5.3: Classified articles

CHAPTER 6

Sentiment Calculation

In this chapter, a dictionary-based approach is used for calculating sentiment scores for both COVID-19 news articles and user comments. The method involves using a pre-constructed sentiment dictionary that assigns a polarity score to each word based on its association with positive or negative sentiment (Mayur Wankhade, 2022). This approach is advantageous because it is relatively simple and efficient to implement, and it allows for sentiment analysis at scale (Mayur Wankhade, 2022; Robert Remus, 2010).

For this project, a dictionary-based approach was used because of the efficiency in computational manner. By using a pre-defined dictionary, only a lookup is required in order to calculate the sentiment for a word (Robert Remus, 2010). In addition to that, dictionary approaches produce comparative results (Laith Abualigah, 2020) and do not need trained data (Mayur Wankhade, 2022). The sentiment calculation for the user comments took 1.5 hours on the server used for this project. Results coming from a dictionary are clear to explain and understand (Laith Abualigah, 2020). For this thesis explainability is one of the main concerns, why using a deep learning approach has not been taken. Additionally, the dictionary *SentimentWortschatz (SentiWS)* that is used here is freely available and increases the reproducibility of this project. While dictionary-based approaches may not be as accurate as machine learning-based approaches in some cases (Mayur Wankhade, 2022), they can still be effective for certain tasks. *SentiWS* has been shown to achieve high accuracy in sentiment analysis for German texts in several studies (Robert Remus, 2010).

When analyzing news articles, the sentiment score is computed by summing the polarity scores of all words within the article. On the other hand, when processing user comments, additional weighting is applied, taking into account various factors such as the number of likes and dislikes, and the existence of discussions within the comment section. These factors are integrated to consider the potential impact that popular comments or active discussions may have on the overall sentiment of the comment thread.

By applying this approach to COVID-19 news articles and user comments, this thesis aims to provide a comprehensive understanding of public sentiment towards the pandemic and to highlight the potential impact of public health policies on public opinion.

SentiWS (Robert Remus, 2010) is used as a dictionary. *SentiWS* is a German-language resource for sentiment analysis and opinion mining. It provides a list of positive and negative words, along with their part of speech tag and inflexions, if applicable. *SentiWS* includes around 1,650 positive and 1,800 negative words, totalling approximately 16,000 positive and 18,000 negative word forms (Robert Remus, 2010). It includes not only adjectives and adverbs expressing sentiment but also nouns and verbs that imply sentiment. By using a *SpaCy Extension*[1] it is now able to use the locally stored dictionary.

For user comments, we have information on whether they are predominantly rated as positive or negative by other users, which is indicated by the *VotePositive* and *VoteNegative* columns that can be either True or False. To account for comments with higher positive ratings, we apply a weighting factor of two to the calculated sentiment score. This approach ensures that comments with more positive ratings are given more weight in the sentiment analysis. Users can influence the discussion sentiment by voting for comments. When users upvote a comment, it indicates that they agree with the sentiment expressed in the comment, and this can contribute to a more positive overall sentiment for the discussion. Conversely, when users downvote a comment, it suggests that they disagree with the sentiment, which can contribute to more negative overall sentiment for the discussion. Therefore, the distribution of positive and negative votes can be used as a proxy to estimate the overall sentiment of a discussion or a comment thread.

6.1 Dealing With Sub Comments

For comment threads, a representative score for the whole subconversation must be calculated. Calculating the median sentiment score of all root comments of an article that has its own sub-comments can provide a more accurate representation of the overall sentiment of the discussion. This is because root comments are typically the main discussion points, and sub-comments often expand upon these points or provide additional insights. Additionally, calculating the median score can help to mitigate the impact of outliers, which may skew the sentiment analysis. The median score is less sensitive to these outliers compared to the mean score, as it only takes the middle value into account, and is, therefore, a more robust measure of central tendency in the sentiment distribution (Coifman, 2001).

With a recursive function, direct replies to a Posting based on a given ID are identified. In the absence of any replies, the current list of sentiment scores, starting with a list of root comments' sentiment scores, is returned. If replies are found, it iterates through them and invokes the recursive function for each reply comment, merging the current

[1] https://github.com/Liebeck/spacy-sentiws last accessed 03.05.2023

found sentiments array with the sentiments of the reply (including sub-replies). The outcome is a large array containing all the found sentiment scores in a discussion.

In this step, an additional problem is being addressed. Some user comments may be too short or contain unknown words, particularly in Austrian dialects, which can result in a sentiment score of *None* for the entire comment. To avoid this issue, it is necessary to exclude these unknown comments from the overall sentiment calculation. However, since comments are often dependent on each other, it is not possible to simply exclude them without losing the entire sub-conversation. The result of the recursive function is an array with found sentiment scores, but it may also contain NaN[2] values. Therefore, when calculating the median, only the non-NaN values in the array should be considered. If the sentiment of the entire sub-discussion is unknown, an overall score of NaN is returned. Finally, root comments with unknown overall sentiment scores are dropped. Out of 127,612 root comments, only 54,278 are used for the comparative analysis.

[2]https://pandas.pydata.org/docs/user_guide/missing_data.html last accessed 03.05.2023

CHAPTER 7

Comparison Analysis

In this section, the gained knowledge is used to compare the sentiment change in the COVID-19 mask requirement per article group. They will take advantage of the found topic clusters and the sentiment scores calculated by a dictionary approach to see changes in the way of reporting and in the user discussion.

7.1 Exploration

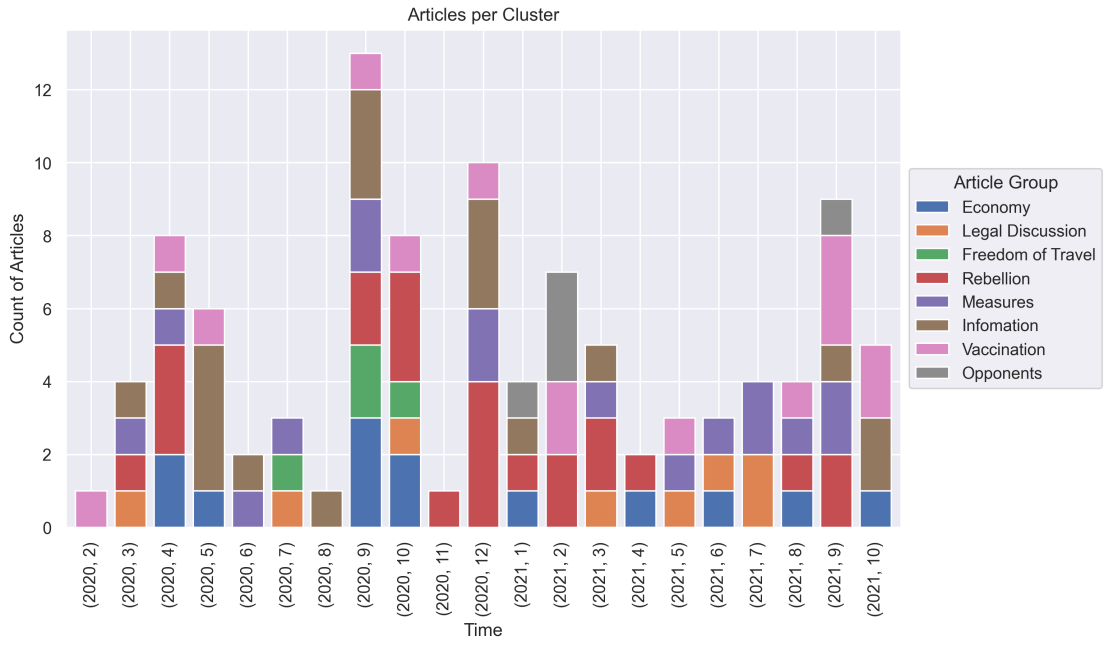

Figure 7.1: Article Count per Cluster

In Figure 7.1, the number of articles published in a group is shown on a time axis. The Figure highlights that a significant number of articles were published in September 2020, whereas the preceding month only had one informational article in the dataset. Notably, articles discussing the Rebellion and the *FPÖ* politician Manfred Haimbuchner remained relevant from the end of 2020 until March 2021. In December 2020, *Der Standard* published four articles in that niche. Additionally, it can be observed that articles related to vaccination gained more popularity over time, while those addressing quarantine and travel restrictions were only popular for a brief period.

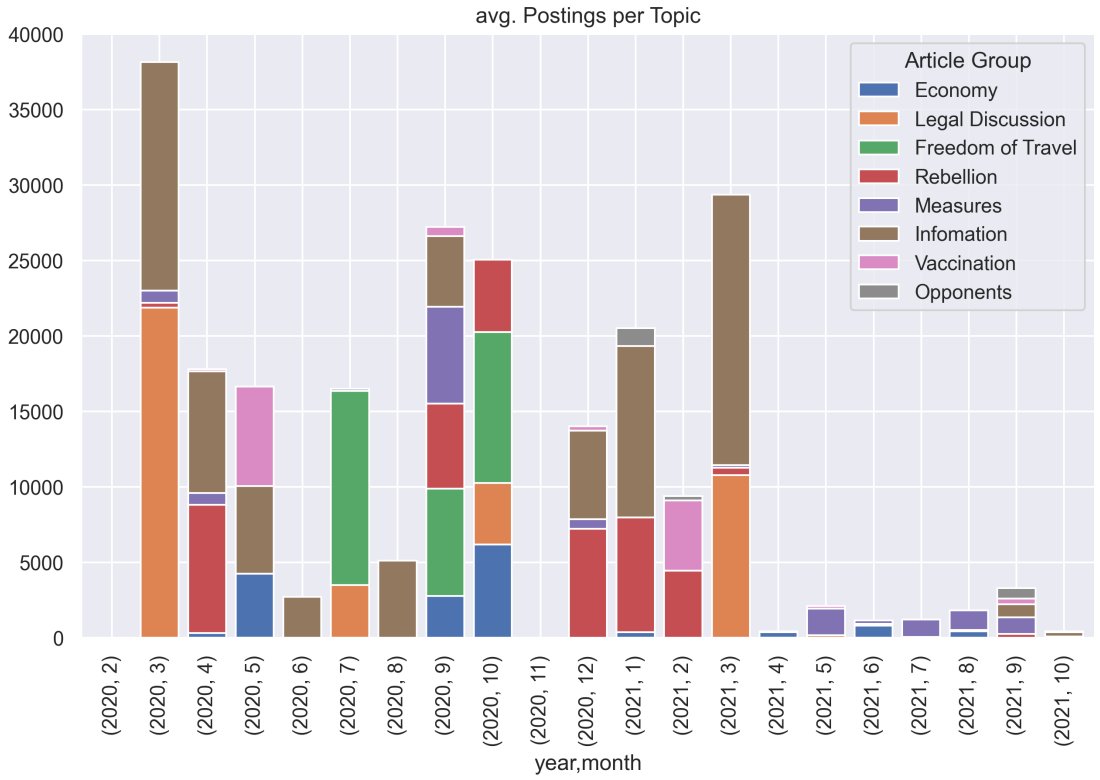

Figure 7.2: Postings Count per Cluster

In Figure 7.2, the number of postings for each topic is plotted on a time axis. It is evident that articles related to legal discussions received an exceptionally high number of comments. For instance, in March 2020, a single article in that group generated over 20,000 comments. Similarly, the small group of articles discussing quarantine garnered significant user traffic in a short period. Additionally, it is interesting to note that the discussions on COVID-19 counterpart articles declined in 2021. Overall, the data traffic plateaued in 2021.

Table 7.1 shows the total number of articles and user comments per cluster.

Figure 7.3 displays that the sentiment scores of COVID-related articles are predominantly negative. This is primarily because topics such as restrictions, diseases, and economic

Cluster	Labeled Topic	Articles	Postings
0	Economy	13	27,775
1	Legal Discussion	8	40,671
2	Freedom of Travel	4	37,055
3	Rebellion	23	98,536
4	Measures	16	23,812
5	Information	19	116,582
6	Vaccination	15	18,405
7	Opponents	5	2,716

Table 7.1: Cluster Size

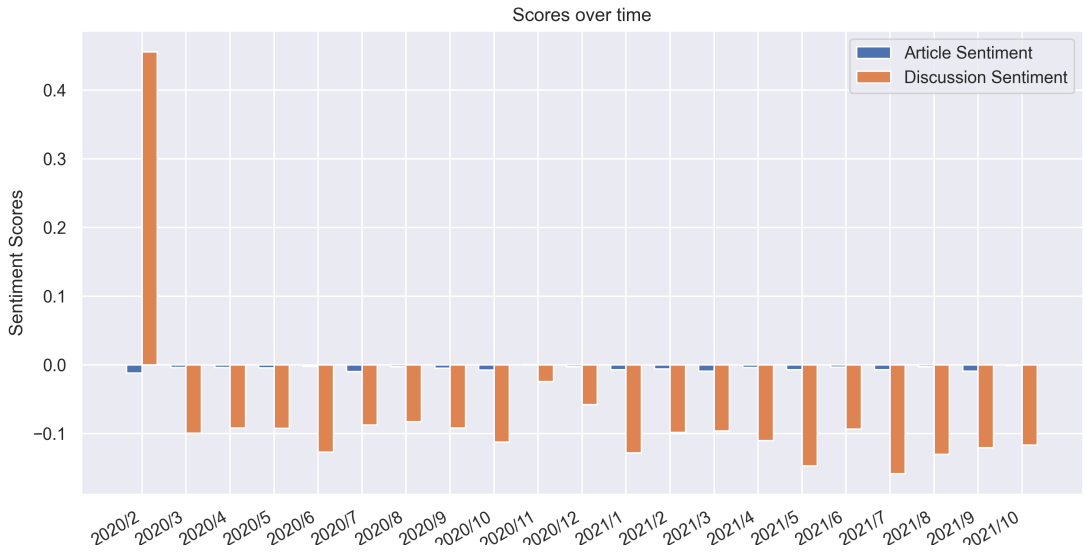

Figure 7.3: Overal Sentiment Evolution

problems are associated with negative word scores when performing dictionary-based sentiment calculations. Focusing on the blue bars, the sentiment of the news articles can be analyzed.

The outliers in Figure 7.2 can be attributed to the absence of article and discussion data. For example, in February and November 2020, only one article was released, and the received comments were too few to be displayed in the figure. Hence, the data in these instances may not be very significant. In general, it can be observed that the discussion sentiment declined in 2021 compared to 2020 (Figure 7.3), but this could also be due to a decrease of postings as shown in Figure 7.2.

Figure 7.4 presents the findings on sentiment patterns within the clusters. Notably, the x and y axes are uniformly scaled across all subplots, enabling a clear visualization of the discussion timing and sentiment performance in comparison to other clusters.

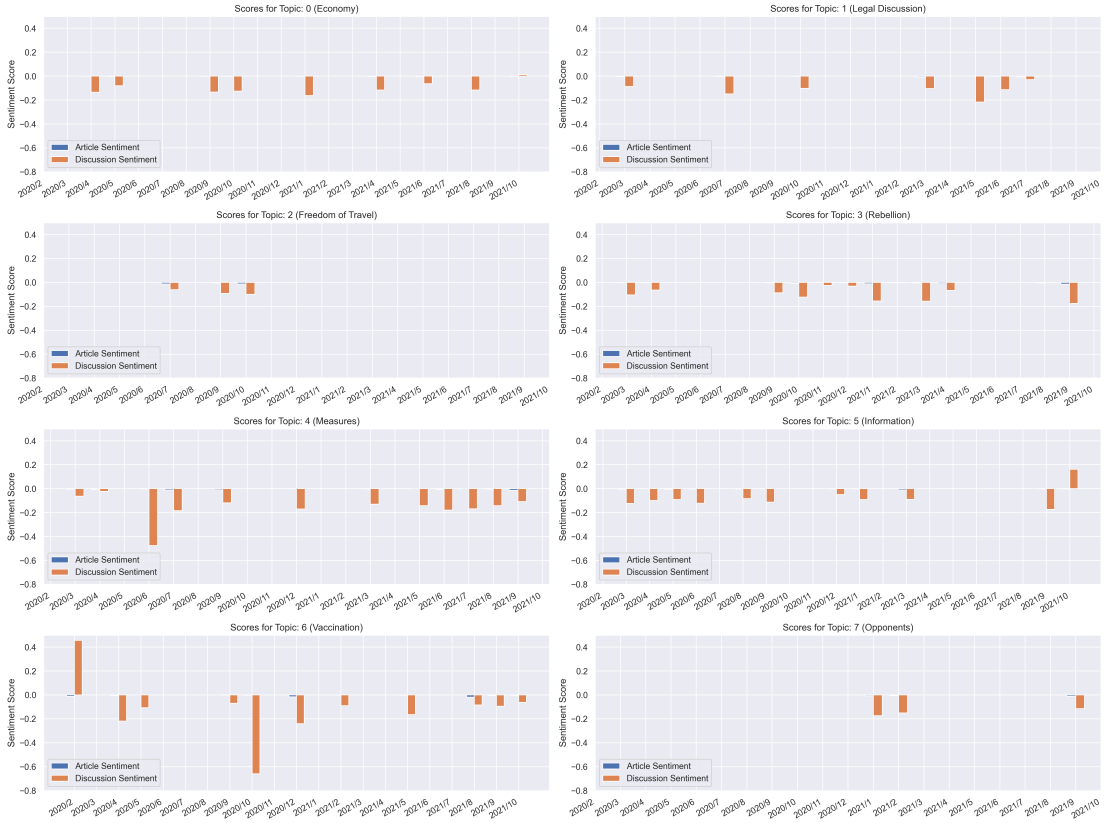

Figure 7.4: Time Sentiment Evolution per Cluster

To interpret Figure 7.1 and 7.2, one must consider whether there was sufficient data at a specific time point. An interesting insight can be derived from the measures article cluster. From mid-2021 to the end of 2021, this cluster received a significant amount of discussion data, and the sentiment remained consistently negative throughout the period, while the sentiment score of the articles stayed close to zero. This indicates a pattern of rejection towards policy. In larger discussions, opposing sentiments can cancel each other out, and it can also be observed that a decrease in discussion data often results in a more extreme sentiment score. As the group of discussants becomes smaller, the sentiment tends to become more polarized.

Figure 7.5 presents a summary of the analysis findings, depicting the clusters on the x-axis and the article/community sentiment scores in the grouped bars. Notably, the *opponents* cluster, which discusses demonstrations and police operations, has the lowest overall sentiment score. However, it is worth noting that this cluster only comprises five articles and received 2,716 comments, which may contribute to the high level of polarization among the participating users.

Another interesting finding is related to the measures topic cluster, which received

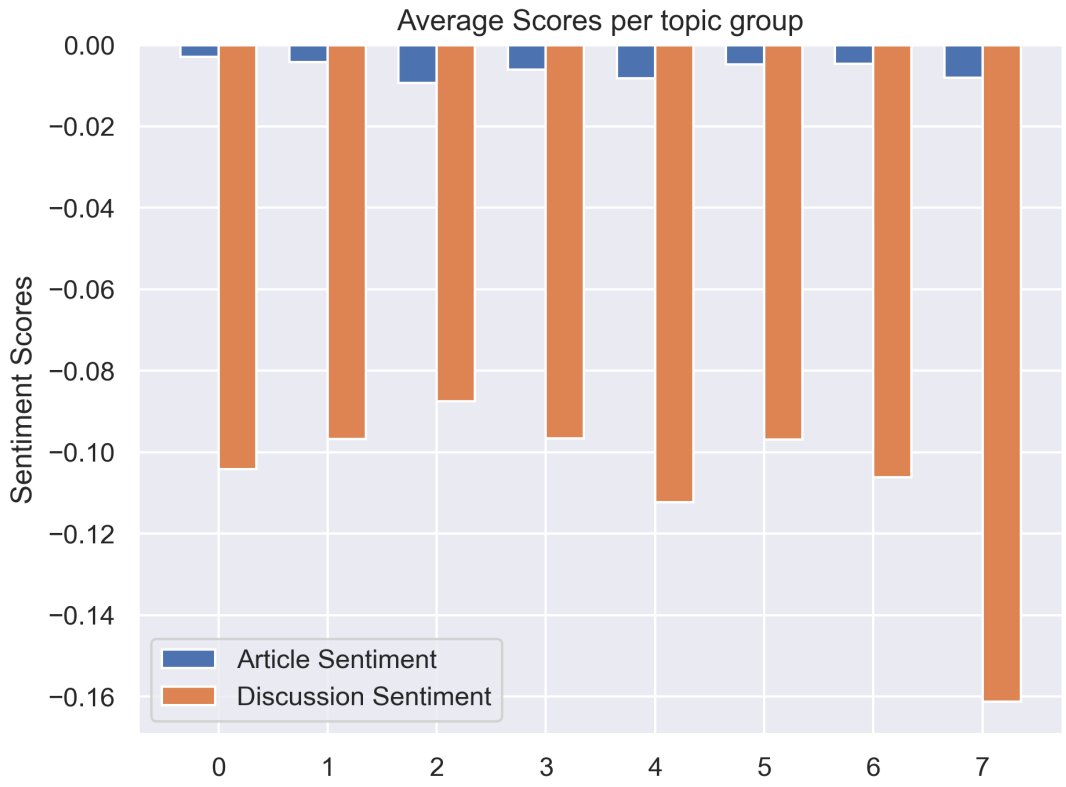

Figure 7.5: Sentiment Evolution per Cluster

a significant number of postings (23,812). The data size was sufficient to generate a comprehensive sentiment analysis. However, the overall sentiment score remained extremely low. Additionally, the *economy* cluster received poor sentiment scores despite having a large number of postings (27,775), indicating that the participating users were dissatisfied with the economic situation.

The vaccination topic was also popular in Austria in the early months of 2021 due to discussions around the vaccination plan. However, the sentiment score for this topic was notably lower than for other topics.

7.2 Conclusion

This research employed NLP techniques such as word embedding and TF-IDF scores to cluster and evaluate sentiment in text data related to COVID-19 in Austria.

In this study, we presented a comprehensive preprocessing pipeline that involved techniques such as stemming, n-gram creation, spell-checking, and emoticon handling to obtain clean text corpora for extracting numeric features. We also discussed the extrac-

tion of numeric features from the text for subsequent clustering model fitting. Moreover, we provided a detailed explanation of the chosen Word2Vec word embedding technique and how vectorization is integrated into our pipeline. Utilizing cluster analysis, we identified the optimal cluster size for K-Means clustering of COVID-19-related articles. By incorporating preprocessing techniques such as PCA, we achieved a improved silhouette score, resulting in the selection of a cluster size of eight. Additionally, we incorporated sentiment analysis by calculating sentiment scores for user comments using *SentiWS* (Robert Remus, 2010). We also elucidated how we weighted the scores with user ratings and tackled the issue of sub-comments in a discussion. We can thus answer the research questions:

- **RQ1:** Which preprocessing techniques should be considered when clustering text data and performing sentiment analysis?

  In NLP, several techniques can be used to preprocess textual data before analysis. Among these techniques, tokenization and stopwords removal are essential steps. However, to enhance the quality of the analysis, additional preprocessing steps can be taken. In our case, we applied stemming to reduce each word to its root form, which allowed us to build n-grams and identify common phrases. These phrases are then useful for word embedding approaches.

  When analyzing user comments, we also had to deal with miswritten comments and dialect words. To address this, we implemented a spellchecker to correct any misspelled comments and removed any dialect words that could have skewed our analysis. Additionally, emoticons used in the comments can express emotions that are relevant to sentiment analysis. Therefore, we extracted the written emotions from the emoticons and used them in our sentiment analysis. Overall, these preprocessing techniques allowed us to obtain cleaner and more informative data for analysis.

- **RQ2:** How can news articles be grouped into distinct topics based on the textual content of each article?

  Numerical features were extracted using Word2Vec. To obtain independent features, we addressed multicollinearity and performed dimensional reduction. The columns were then scaled and transformed for clustering the articles into groups using the K-Means algorithm. The grouping was based on features extracted directly from the words used in the articles.

- **RQ3:** How can sentiment scores be computed for user comments while taking into account both user ratings and sub-comments?

  A dictionary-based method was used, which involved using a pre-constructed sentiment dictionary that assigns a polarity score to each word based on its association with positive or negative sentiment. The method was simple and efficient to implement and allowed sentiment analysis at scale. Sentiment scores were calculated by summing the polarity scores of all words within the article or comment. For user

comments, additional weighting was applied, taking into account various factors such as the number of likes and dislikes, and the existence of discussions within the comment section. The sentiment dictionary used is called *SentiWS*, which is a German-language resource for sentiment analysis and opinion mining. It provided a list of positive and negative words, along with their part of speech tag and inflections, if applicable. The distribution of positive and negative votes could be used as a proxy to estimate the overall sentiment of a discussion or a comment thread. The median score was calculated to provide a more accurate representation of the overall sentiment of the discussion. A recursive function was used to identify direct replies to a Posting based on a given ID, and to merge the current found sentiments array with the sentiments of the reply, including sub-replies. Comments containing unknown words or too short are excluded from the overall sentiment calculation.

- **RQ4:** What trends and patterns in sentiment can be investigated by comparing different groups of articles over a specific time frame?

  The findings indicated that the COVID-19 discussion peaked early in 2020, with a high level of discussion on the legal basis for mask requirements. During this time, informational articles received a significant amount of engagement. The topic of Rebellion remained popular from late 2020 to early 2021, after which its popularity declined. Since mid-2021, measures topics have become more dominant, with a decrease in the demand for informational articles.

  Regarding sentiment evaluation, the predominance of negative sentiment scores can be attributed to the association of negative words with topics such as restrictions, diseases, and economic problems. The findings suggested that data size is an essential factor to consider when interpreting the sentiment scores of discussions. The measures article cluster displayed a consistent negative sentiment towards policy, whereas the opponents' cluster, comprising only five articles, exhibited a high level of polarization among participants. The economic cluster also had poor sentiment scores, indicating dissatisfaction among the users.

Overall, the findings of this study provide valuable insights into the sentiment of news articles and user comments related to COVID-19 in Austria. The study could inform policymakers, media outlets, and researchers to improve their communication strategies and address concerns raised by the public. However, further research is necessary to identify the underlying causes of the negative sentiment observed and explore potential solutions.

7.3 Discussion and Future Work

The article group analyzed a notable shift in sentiment towards the end of 2021 in the political domain. Austria faced significant issues with changing secretaries in various

departments during the COVID-19 pandemic, resulting in political instability that could potentially explain the negative sentiment among the population during that period.

Discussions and plans for vaccinations in Austria were met with protests and large demonstrations, which could be a potential explanation for the negative sentiment surrounding articles on vaccination andopponents.

However, it should be noted that this thesis is solely based on a subset of COVID-related news data. To thoroughly analyze the reasons behind the sentiment change, a larger dataset must be employed. Additionally, in order to properly examine the sentiment shift, connections to political events must be established, and psychological motives should also be taken into account.

Furthermore, it would be intriguing to investigate if the sentiment of discussions is correlated with the sentiment of the articles. Although this thesis solely focuses on objectively written articles, the pattern may differ when analyzing discussions on opinion essays.

An intriguing avenue for future exploration involves retaining the research problem while exploring various methods and strategies to extract more insightful information. One possible direction is to compare different clustering methods for grouping articles and explore diverse word embedding techniques. Furthermore, it would be worthwhile to compare the outcomes of sentiment analysis using a dictionary-based scoring system with those obtained from a neural network-based approach. This comparative analysis would offer valuable insights and shed light on the effectiveness of different methodologies.

7.4 Repository

The code described can be found under this *GitHub* repository[1]. A corresponding *Readme* file explains the recommended order in which the notebooks should be executed to obtain the desired results. Intermediate results are stored in the *data/feature* directory. The clustered articles can be found in the *data/feature/clusters* subfolder, which includes the document IDs and their content. In addition to the data, the Python code for analyzing news articles is located in the *content* directory, and the code for analyzing user-generated content is in the *comments* directory. The repository also contains the plots used for comparing the results.

[1] https://github.com/lukasthekid/corona-mask-requirement-nlp-analysis last accessed 20.06.2023

List of Figures

4.1 CBOW Architecture, taken from Efficient Estimation of Word Representation in Vector Space 15
4.2 Correlation within the Document's Features 16
4.3 Explained Variance in PCA 17

5.1 Silhouette Scores (with PCA) 21
5.2 Silhouette Scores (without PCA) 22
5.3 Elbow Method Plot 23
5.4 Silhouette Plot 24
5.5 Raw TF-IDF Scores Table (8 Clusters) 24

7.1 Article Count per Cluster 33
7.2 Postings Count per Cluster 34
7.3 Overal Sentiment Evolution 35
7.4 Time Sentiment Evolution per Cluster 36
7.5 Sentiment Evolution per Cluster 37

List of Tables

5.1 Words for Cluster Size: 3 25
5.2 Words for Cluster Size: 8 25
5.3 Classified articles 27

7.1 Cluster Size 35

List of Algorithms

2.1 Retrieve Top 10 Articles Related to COVID-19 and Mask Mandates . . 6

Bibliography

Aizawa, A. (2002). An information-theoretic perspective of tf–idf measures.

Andrea Ceron, F. N. (2015). Public policy and social media: How sentiment analysis can support policy-makers across the policy cycle.
doi: 10.1483/81600

Austria. (2020). Bundesgesetzblatt für die rebublik Österreich. 463. verordnung: Covid-19-schutzmaßnahmenverordnung – covid-19-schumav.

Barhoumi, L. H. B., Amira. (2017). Document embeddings for arabic sentiment analysis.

Barrus, T. (2023). pyspellchecker documentation, release 0.7.1.

Castillo, D. (2021). *How to cluster documents using word2vec and k-means.* Zugriff am 2023-04-04 auf `https://dylancastillo.co/nlp-snippets-cluster-documents-using-word2vec/`

Coifman, B. (2001). Estimating median velocity instead of mean velocity at single loop detectors.

CR, A. (2020). *Topic modeling using gensim-lda in python.* Zugriff am 2023-04-04 auf `https://medium.com/analytics-vidhya/topic-modeling-using-gensim-lda-in-python-48eaa2344920`

Cucinotta D, V. M. (2020, March). Who declares covid-19 a pandemic.
doi: doi:10.23750/abm.v91i1.9397

Dandannavar, P. S. (2019). Emoticons and their effects on sentiment analysis of twitter data.

David Johnson, V. M. (2006). More effective web search using bigrams and trigrams.

Diksha Khurana, A. K. (2022). Natural language processing: state of the art, current trends and challenges.

Dubey, A. D. (2021). Twitter sentiment analysis during covid-19 outbreak.
doi: 10.2139/ssrn.3572023

Hartigan, J. A. (1979). Algorithm as 136: A k-means clustering algorithm. , 100–108.
doi: 10.2307/2346830

Honnibal, M. (2014). spacy documentation, release 0.1.

Huang Chaolin, L. X., Wang Yeming. (2020). Clinical features of patients infected with 2019 novel coronavirus in wuhan, china.

Humaira, H. (2020). Determining the appropiate cluster number using elbow method for k-means algorithm.
doi: 10.4108/eai.24-1-2018.2292388

James Kavanagh, K. G. (2023). Assessing the effects of lemmatisation and spell checking on sentiment analysis of online reviews.

Janiesch, C. (2021). Machine learning and deep learning.

Khafaie Morteza Abdullatif, R. F. (2020). Cross-country comparison of case fatality rates of covid-19/sars-cov-2.

Laith Abualigah, M. S., Hamza Essam Alfar. (2020). Sentiment analysis in healthcare: A brief review.

Marwa Naili, A. H. C. (2017). Comparative study of word embedding methods in topic segmentation.

Mayur Wankhade, A. C. S. R. (2022). A survey on sentiment analysis methods, applications, and challenges.

Meyer, D. (2016). How exactly does word2vec work?

Mikolov, K. C. G. C., Tomas. (2013). Efficient estimation of word representations in vector space.

Min Shi, J. L. (2017). We-lda: A word embeddings augmented lda model for web services clustering.

Muhr, R. (2008). The pragmatics of a pluricentric language: A comparison between austrian german and german german. *PRAGMATICS AND BEYOND NEW SERIES*, *178*, 211.

Pennington, R. S., Jeffrey. (2014). Glove: Global vectors for word representation.

Ponmani, K. D. (2022). Clustering based sentiment analysis on twitter data for covid-19 vaccines in india.

Rehurek, R. (2020). gensim documentation, release 0.8.6.

Richardson, L. (2019). Beautiful soup documentation, release 4.4.0.

Rio Pramana, A. A. S. G. (2022). Systematic literature review of stemming and lemmatization performance for sentence similarity.
doi: 10.1109/ICITDA55840.2022.9971451

Robert Remus, G. H., Uwe Quasthoff. (2010). Sentiws – a publicly available german-language resource for sentiment analysis.

Rodionova, O. (2021). Efficient tools for principal component analysis of complex data—a tutorial.

Sara Elshobaky, N. A. (2018). A platform for sentiment analysis on arabic social media.

Shoaib, A. M. (2021). Detecting sentiment dynamics and clusters of twitter users for trending topics in covid-19 pandemic.
doi: 10.1371/journal.pone.0253300

Shrestha, N. (2020). Detecting multicollinearity in regression analysis.

Sinaga, K. P. (2020). Unsupervised k-means clustering algorithm.
doi: 10.1109/ACCESS.2020.2988796

Singh, S. K. (2019). Sentiverb system: classification of social media text using sentiment analysis.

Sukavanan Nanjundan, S. S. (2019). Identifying the number of clusters for k-means: A hypersphere density based approach.

Sven Martin, J. L. (1997). Algorithms for bigram and trigram word clustering.

Tiffani, I. E. (2020). Optimization of naïve bayes classifier by implemented unigram, bigram, trigram for sentiment analysis of hotel review.

Tomas Mikolov, K. C. (2013). Efficient estimation of word representations in vector space.

Tripti Agrawal, A. S. (2019). An effective knowledge-based pre-processing system with emojis and emoticons handling on twitter and google+.

Vimala Balakrishnan, E. L.-Y. (2014). Stemming and lemmatization: A comparison of retrieval performances.

Wartena, C. (2019). A probabilistic morphology model for german lemmatization.

Wenchuan Mu, K. H. L. (2022). A clustering-based topic model using word networks and word embeddings.

WHO, W. H. O. (2020, Jannuary). Preliminary investigations conducted by the chinese authorities have found no clear evidence of human-to-human transmission of the novel coronavirus (2019-ncov) identified in wuhan, china.

Zhang, S. W. B. L., Lei. (2018). Deep learning for sentiment analysis: A survey.

Zhao WX, W. J. H. J. L. E. Y. H., Jiang J. (2011, April). Comparing twitter and traditional media using topic models. ineuropean conference on information retrieval. doi: 10.1371/journal.pone.0253300