Norges miljø- og
biovitenskapelige
universitet

Master´s Thesis 2020 30 ECTS
Faculty of Science and Technology

Natural Language Processing and Topic Modeling for Exploring the Vegetarian and Vegan Trends

Marius Aleksander Olavsrud
Master of Science in Data Science

Preface

2 great years at NMBU are finally done
Student life has been both stressful and fun
Humble, happy and proud to deliver
No longer any reason to elaborate and shiver.
Topic modelling and big data will rule the information of the world
the public opinion will be disclosed and people will curl
Nofima wants to understand vegetarian and vegan food
The vegetarians will probably be in a better mood

My greatest appreciation must be directed to my motivating, always present and talented supervisors Ingunn Berget at Nofima, and Oliver Tomic and Kristian Hovde Liland at NMBU. Their genuine and constructive feedback have inspired me to always do my very best and to overcome larger and smaller obstacles. Without you all, this thesis would actually not have come to an end. It is no secret that the Covid-19 pandemic, which resulted in the Norwegian society closing down on March 12th, had a great negative impact on my writing progress. As I and my whole family tested positive on Covid-19, the support from all three of you became crucial.

Further, my appreciation also goes to the kind employees at Nofima who helped me perform a manual reading of a subset of the data. Covid-19 also caused a sudden and sad stop for the great inspirational conversations during lunch breaks at Nofima. It is not to forget the amazing and nutritious food that was served at Nofima. A result I did not anticipate, was that I came to love vegetarian food.

I want to thank my co-students at NMBU as well as my friends and family who have shown interest and a great deal of involvement in my thesis. You have all given me energy and made sure that these past six months have not only been spent on this study.

Last, but not least, I want to thank my parents for your absolute amazing support and unlimited number of delicious lunches and dinners.

Enjoy your reading!

Ås, 29th June, 2020

Marius Aleksander Olavsrud

Abstract

The purpose of this thesis is to examine how topic modeling can be used as a tool to explore large sets of text data. This thesis is written on assignment from Nofima Food Research Institute. A set of about 52 000 unknown texts of various lengths were downloaded using an external web-harvesting company (Webhose.io). The texts are collected with a specific search query consisting of food related vegetarian and vegan based keywords as this is a field of interest with Nofima. Latent Dirichlet Allocation, known as LDA, is used to create and model these topics. LDA is a method that allows unobserved groups of similar data to be explained by a group of words known as a topic.

The collected texts are split into smaller subsections based on the type and lengths before being preprocessed for non-relevant information. A subset of medium length texts are used for the modeling. Further, the data is analysed with LDA, using coherence score as a metric to determine the optimal number of topics. The results are visualised using pyLDAvis. Lastly, a small subset of the same texts are manually read by a group of employees at Nofima to validate the quality of the results in order to get a better understanding of the type of data that is analysed.

The study discovered that topic modeling can be used to explore a large set of data and get some meaningful insight of parts of the content. Several topics were found to include vegetarian and vegan related words. Some of these words were found to have a high probability of existence within the topic in question.

The process revealed numerous concerns which needed to be addressed. Some examples were many non-related documents, large amounts of words that were not related to a given topic, deciding upon the optimal number of topics as well as visualisation of the topics.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The following thesis will go into detail on how to use a sub-field of text mining, known as topic modeling, to extract and explore the most important themes of a set of texts. The thesis is written on assignment from Nofima Food Research Institute. Nofima is a leading institute for applied research within the fields of fisheries, aquaculture and food research. They were interested in understanding how they can apply text analysis to improve their research. Due to the significant increase in the popularity of vegetarian and vegan food products, Nofima saw the need to learn about the core of the public opinion on vegetarian and vegan food and lifestyle. Nofima was curious as to what extent topic modeling would be an efficient tool for exploring such large amounts of relevant web data.

The study is approaching an improved possibility within the food and consumer science. This way of analyzing the market does exist [1], however, it seems to be the case that current research has not taken full advantage of these methods. As such, opportunities are revealed that researchers and businesses possibly will be able to make use of. Politics, technological development, world health, production sustainability and environment are alternative examples of fields where such tools may also be crucial in the near future.

Based upon the growing need for improved knowledge within text mining, the following research question was developed:

To what extent is it possible to explore a set of machine-readable texts, not knowing much of the content in advance, but still understanding the significant aspects afterward?

The purpose was to understand and evaluate how written web data could serve the need for structured information. Nofima's need for learning more about the core of the public opinion on vegetarian and vegan food is, in this thesis, used as an example of how text mining can be applied.

This thesis will show that it is essential to simultaneously grasp over more extensive sets of data and find hidden connections. The hypothesis was that it is possible to explore a set of indefinite amounts of unknown texts. However, the result is highly dependent upon the analysis as well as correct preprocessing. George Fuechsel, an IBM programmer and instructor, used to say: ”Garbage in, garbage out” [2]. It illustrates the importance of correct preprocessing and analysis, as insufficient preprocessing and analysis will give less reliable results, which furthermore are harder to understand and thus evaluate. Similar studies of topic modeling have been performed on known data sets with varying results. These will be evaluated within the background section 1.4.

The entire analysis was performed in Python using a range of tools and packages created for text mining and specifically topic modeling. Topic modeling is a method to explore a large amount of textual data using machine learning. The method is a type of cluster analysis that finds similarities among words within a set of documents. The goal is to detect the abstract themes that exist within the documents and to be able to get an overview of the essential parts of a data set. The topics are presented as a list of lists of words, as shown in Figure 2.1 in the theory chapter. The words are ordered from largest to smallest based upon their probability of existence within each topic. More details are found in the theory chapter.

The topics were found with Latent Dirichlet Allocation, known as LDA. LDA is often referred to as a synonym to topic modeling and is also known to be the preferred modeling technique [3]. LDA is a method that allows unobserved groups of similar data to be explained by a group of words known as a topic. Each topic is a distribution over the vocabulary, where the vocabulary, also known as the dictionary, is based upon the set of documents (the texts to be analyzed). Further, the documents are mixtures of these topics found from the words existing within each topic and each document.

In this study, a set of unknown documents were collected based upon a search query consisting of vegetarian and vegan food-related words. The documents were then preprocessed before topics were created. Lastly, the topics are visualised, and manual validation of a smaller sub-section was performed.

1.1 Background

The number of published journals, articles and research papers increases every year. So does the number of false papers and incorrect information. Both of these numbers are known to be significant concerns among researchers as the amount of unstructured data becomes even more unmanageable. Unstructured data exist in many forms, like digital images, videos, audios and texts, among others [4]. This type of data is known to be significantly harder to work with than structured data, which is much easier to handle when performing machine learning. There are currently more than 4.5 billion active internet users and this number keeps increasing [5]. With such a large number of active internet users, it can be expected that the gap between unstructured and structured data will become even larger [4]. The gap will make data management even more complicated, as the analysis process will be more time consuming and unmanageable.

Improved text mining will allow for easier handling of this unimaginable amount of unstructured data. Furthermore, it will provide easier exploration and memorization of this data. These facts enhance the importance of the research question, which was: *To what extent is it possible to explore a set of machine-readable texts, not knowing much of the content in advance, but still understanding the significant aspects afterward?*

Researchers believe that about 80 percent of all business-related data are unstructured [6, 4]. This indicates yet another reason why text mining is needed. Text mining will make it easier for businesses to explore their unstructured data and take advantage of their previously created documents. An example of this could be a consulting firm where two or more consultants lack the opportunity to compare each others previous findings and advises in similar cases. This kind of situation often occurs, according to discussions with consultants of various firms. To account for these concerns, there exist a number of models already, and some of the most prominent results come from using text mining.

It is important to define what is meant with topic modeling for the purpose of this thesis as there exist a number of definitions and explanations. Topic modeling is a sub-field of text mining. As Hotho et al. [7] described in their paper, ”A Brief Survey of Text Mining”, text mining can be divided into three different sub-fields with its own definitions. The first sub-field of text mining is defined as information extraction [7]. The goal is to extract facts from texts and is known to be a restricted form of natural language understanding [7]. One knows in advance the type of semantic information to look for and the task is to extract parts and assign specific attributes [7]. An example would be an online price checker. The product is known and the previous price is known. The new price is then updated based on a new available price. The process would decompose into a series of steps, typically including sentence segmentation, tokenization, part-of-speech tagging, and named entities such as name of organizations. Some of these steps will be described in the theory chapter of this thesis as they are important to topic modeling as well.

The second sub-field of text mining is defined as text data mining [7]. The purpose is similar to data mining where methods and algorithms from the fields of machine learning and statistics are used on texts with the goal of finding patterns [7]. Information extraction and Natural Language Processing (NLP) methods are used to extract information [7]. An example of text data mining is topic modeling. One wants to be able to extract a number of themes also known as topics. The method is used to categorize and structure text collections or extract useful information [7]. As explained further into the thesis, the main steps of topic modeling are data collection, preprocessing, and creation of topics before visualization of results.

The third sub-field of text mining is defined as knowledge discovery in databases as a process [7]. This is a process where steps from both information extraction and text data mining are used. The goal is to find connections and hidden patterns in a set of data with techniques from the first and second definition of text mining [7]. An example could be fraud detection. Bank databases are constantly analysed and small changes in the pattern of people's spending might indicate fraud. In this situation, raw data such as text files are sent into a model which preprocesses and transforms the data using methods from information extraction. The preprocessed data are then analysed with text data mining methods before being evaluated and interpreted to look for pattern changes.

While there exist three sub-fields of text mining with their own definitions, only the second sub-field will be relevant for the purpose of this thesis. However, some methods from the first sub-field will be included as they are much used with topic modeling [7]. As a result of NLP being the most important building block for any topic modeling, a section covering its history will be included. There will also be a section covering more perspectives of how topic modeling is used today and how researchers believe it will be used in the years to come. As GDPR is a major concern of interest when collecting large amount of data, a section covering the importance of this, will also be included.

In his guest lecture at the University of Edinburgh, David Blei, one of the founders of LDA, talked about how topic modeling is needed to be able to organize, visualize, summarize, search, predict and understand massive collections of machine readable documents [8]. Blei also points out that topic models can be used, and have been used, to annotate pictures, connections among people and genetic data [8]. The goal of topic models is to be able to discover the thematic structure of text while annotating the documents and using the annotations to visualize, organize and summarize the results. It would be beneficial to be able to do this with millions of documents simultaneously [8].

As explained above, the idea with topic modeling is to be able to explore a large set of unknown texts. The goal is to get some meaningful insight of the themes of the texts as well as to explore the main focuses. An example of such a focus could be to detect reasons why people decide to go vegetarian or vegan. Topic modeling is used for a large number of different applications such as marketing, recommendations, information extraction, exploration of unknown data, and summarizing large collections of text, among others. The thesis will go into depth of what topic modeling is, details on where it is used and most importantly how it can be applied for a set of unknown texts. However, the history of its origin and how statistics have adapted into machine learning models will be covered first.

1.2 History of Natural Language Processing

There are several reasons why Natural Language Processing, and text mining are increasingly important in today's society. Most people use it every day, not knowing that many of their daily interactions with any type of technology are dependent upon good Natural Language Processing. Possibly, the most common use is writing text messages. Spelling and grammar corrections are based on NLP and improved NLP techniques will improve the quality of the corrections. More and more people are also using it to translate texts into other languages and when talking to their electronic devices.

Natural Language Processing has existed for more than 70 years [9]. Researchers would argue that it started in the 1950s, but earlier work is also known to exist [9]. In early stages Natural Language Processing was only a hard-coded set of rules. The rules used simple grammar and reduction of words to their word stem (stemming) to perform NLP [9]. Word stemming is simply reducing words to their base or root form, such as ”swimming” becoming ”swim”. The methods did not work very well, and the abilities were limited. Due to the lack of computing power and performance, text mining and the use of NLP vanished for decades [9].

Then, in the late 1980s and early 1990s, a new era for Natural Language Processing was a fact [9]. The statistical revolution had begun. Instead of these hard-coded set of rules, math and statistics were used. Over the years, methods such as Latent Semantic Indexing (LSI), probabilistic Latent Semantic Indexing (pLSI), Latent Dirichlet Allocation and Hidden Markov Models (HMM), among others, were implemented to improve text mining [10]. However, it should be mentioned that some of these methods existed for other purposes prior to being implemented for text mining. As LDA is the preferred model among researchers and LDA is based upon LSI and pLSI, these will be explained in further detail in the theory chapter [3].

Over the next decades, more computer power, better integration with machine learning and improved statistical models took text mining to new levels. However, it was not until Artificial Neural Networks (ANN) were adopted to NLP that text mining gave reasonable results. Improved models opened for better translation abilities, context understanding and generating abilities among others [11]. In later years, researchers have incorporated GPUs to perform parallel computing, which has allowed for faster training of ANN's [12]. This in turn, has allowed for larger data sets to be analysed and better performance. Based on the history of Natural Language Processing and the fast pace of new and better models, no one knows where one are in 10 or 20 years. However, it is likely that new and better technology with a massive open source community will allow for even better results and most likely even more applications.

1.3 Current and future use of topic modeling

As understood from the history of Natural Language Processing, there is no doubt that topic models, which are highly dependent upon NLP, have improved over the years. An early version of what one today calls a topic model, was first proposed in the beginning of the 1990s by Deerwester et al. in the well known paper "*Indexing by latent semantic analysis*" [13]. This method will be explained in greater detail in the theory section. About a decade later, Blei et al. proposed Latent Dirichlet Allocation [14], which for many has become synonymous with topic modeling. LDA is in most situations and by most researchers the preferred method since it has proved to give the best and most accurate results [3].

One might believe that topic modeling is of interest only among researchers but there are several other areas that make use of topic modeling. Topic modeling has been used in marketing applications [15], and an example was when Amado et al. identified research trends on big data in marketing. They analysed a total of 1560 articles published between 2010 and 2015 [15]. Their study tried to distinguish what dimension, out of the five (big data, marketing, geographic location, sectors and products), that was most dominant in the articles [15]. Using topic modeling, they managed to detect which dimension that was the most important one within the documents. Further analysis even gave them detailed insight on a gap they were unaware of. The analysis showed that there exists a gap between the big data research and the benefit of marketing [15]. This indicates how topic modeling possibly can be used for other purposes as well.

Another example where topic modeling, among other text mining techniques, has been used lately are in the detection of Covid-19. When searching for "topic modeling and Covid-19" on article databases such as Google Scholar, arXiv and Science Direct, the search will find several studies that have used these methods to extract information about the spread of the disease. Some of the studies within the search have tried to analyse text data from Twitter, Facebook or pure news with varying results. Most of these studies seem to be non-peer-reviewed, which is why they will not be covered in greater detail in this thesis.

There are also companies which are using similar methods to detect epidemics and pandemics. An example is Bluedot's work on detecting Covid-19 [16]. Back in late December of 2019 a critical warning alert was detected at BlueDot in Toronto, Canada. Their AI-driven algorithm, which uses machine learning and NLP, detected news similar to the SARS epidemic [16]. The news were collected from China in the province around Wuhan. Using expert knowledge within the company, they could predict - that very same day - that a pandemic was likely to occur and they could also predict the next cities to be hit [16]. One cannot find the specific type of technology behind BlueDot, other than the fact that their software is AI-driven machine learning models using NLP [16]. BlueDot does, however, state that their software searches hundreds of thousands of sources every 15 minutes, 24 hours a day [17]. Their algorithm searches information in more than 65 languages and it can detect similarities with already existing results. The founder has also said that their company is highly dependent upon big data and their trained models [17].

Topic models also have the potential to be used to predict and suggest relevant books to people [8]. Using knowledge of what someone has read and then using topic models, one can analyze their preferred books and compare them to other available books. The same goes with movies; one can analyse what people have seen and then compare it to movie reviews and recommendations by other people. Further more, topic model recommendations can be made [8]. Topic models can also be used to learn about the texts being analysed, but possibly more important, one can use them to learn about a person [8]. An example of this, is knowing which books Charles Darwin read and by that, perform a better analysis of him as a person. Analysing the books he read gives insight in the way he learned, thought and explored new ideas [8].

Topic modeling can also be used to identify new product ideas. Analysing large amounts of product reviews might detect opinions on new and innovative products, new design ideas and new technology to be included. An example of this, is Ko et al.'s work on identifying product opportunities using social media mining [18]. They used topic modeling to identify the product topics discussed by customers [18]. Their idea was to use topic modeling to analyze large-scale reviews related to a given product, before using chance discovery theory to create new product opportunities [18]. Chance discovery theory will not be explained in greater detail, as it is not relevant for the purpose of this thesis, but more information can be found in their article "*Identifying Product Opportunities Using Social Media Mining: Application of Topic Modeling and Chance Discovery Theory*" [18]. According to Ko et al. their study can be used as an expert tool to generate practical product opportunities [18]. This will reduce product developers time-consuming tasks, as they can monitor customer interests and trends [18]. They will no longer need to study reviews one by one. One can easily imagine how a car designer can make great use of such a tool. Ko et al. managed to use topic modeling for their desired outcome and found that rare topics might identify new product opportunities, while frequent topics are important, but not for the purpose of new opportunities [18].

The above examples show how topic modeling is used not only among researchers but also among companies for several alternative uses. It shows that regular people most likely will depend even more upon the abilities of topic modeling and NLP in the near future. The overwhelming amount of unstructured data in combination with the large number of new unstructured data published every minute and ever day of the year, give reason to believe that the fast pace improvements of topic modeling will continue [4]. Companies such as BlueDot, Alibaba, Facebook, Apple and Google, among others, are working on advancement of text mining and new areas where the technology can improve people's lives [19]. Dr.

Kamran Khan, the founder of BlueDot, has said that he believes future improvements will allow for even earlier detection of new diseases [16]. He does, however, underline the fact that “We don’t use artificial intelligence to replace human intelligence, we basically use it to find the needles in the haystack and present them to our team” [17]. As Dr. Khan points out, AI will allow for improved ”needle searches”. On top of what these major companies are working with, it is no secret that the amount of non-available information in the form of unstructured data is enormous and could have been applied to better use [4].

1.4 Related work using topic models

The above description of topic modeling to business purposes does not mean that topic modeling is not highly researched and used among researchers. The following section will describe how researchers have used similar methods in their own studies and how the method has extended over the years. This section is, however, beyond the scope of this study, which is why the different methods will not be explained in further detail.

McCallum et al. proposed a probabilistic generative model that simultaneously discovers groups among the entities and topics among the corresponding texts [20]. They used a group-topic model which works slightly different than LDA, in the way that it clusters entities to groups and clusters words into topics. This differs from pure word distributions such as Latent Dirichlet Allocation [20]. It allowed the researchers to discover salient topics in social networks, not detected by solely word distribution methods [20].

Another study where topic modeling was used is Chang et al.’s work on a novel probabilistic topic model also known as Nubbi (Networks Uncovered By Bayesian Inference) [21]. Their goal was to use topic modeling to infer relationships between entities such as people, places, genes and corporations [21]. With Nubbi they managed to detect and understand relationships among the above entities hidden within plain text in network data [21].

Davison and Hong performed a study where they applied topic modeling based on Twitter data [22]. They proposed a study where they addressed the problems of using standard topic models on micro-blogging environments, such as Twitter [22]. The researchers proposed several schemes to train a standard topic model and found that the length of the documents, which is used for training, highly influences the effectiveness of the model [22]. They showed that aggregating short messages often improves the training and the quality of the final model [22]. The study also showed that topic modeling approaches can be a very useful tool for short text analysis [22].

The above examples show just a few studies where topic modeling has been used to discover groups within social networks, relationships across several different entities, as well as a tool to better understand micro-blogging environments. As both social networks and micro-blogging explode in popularity, being able to automatically understand and analyse these platforms becomes increasingly important. As all of the above discussed studies state, large chunks of the information found using the models would have been impossible to detect otherwise [20, 21, 22].

1.5 GDPR

As this study focuses on topic modeling, a large amount of textual data was needed. There were some concerns that needed to be addressed with regard to GDPR (General Data Protection Regulation). In 2018 new GDPR restrictions were put in place all across the European Union [23]. This caused some changes, and made people more aware of the regulations. The regulations limited the amount of time personal-data could be stored and the type of data which were allowed to be stored.

Within this study, an external company was used to perform the actual document collection, which reduced the possibility of incorrect gathering. Webhose.io was performing the data collection, and stated that they were compliant with the current GDPR. They focus on how they handle personal information with regard to their own customers, as well as how they avoid special data. Such data could be ethnic origin, political opinions and religious beliefs, among others.

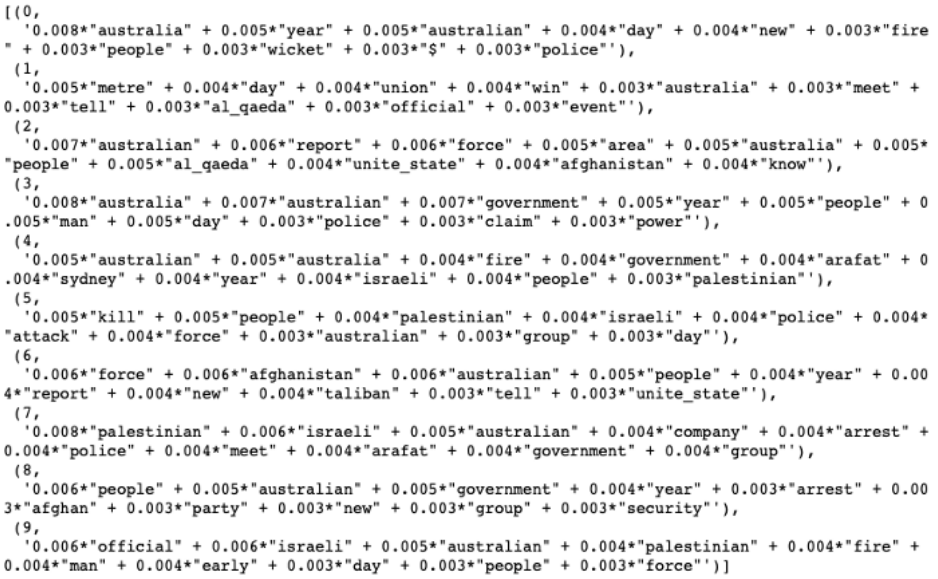
Chapter 2

Theory

The following chapter will in detail explain the theory behind topic modeling and existing tools as well as Python libraries. The methods that have been used in the current study will then be explained in the methods chapter. Here there will be more specific reasoning for the choices that have been made as well as explanations of the code that can be found in the appendix. The following chapter will be split into the stages of topic modeling performed within this thesis, hence; data collection will be found in section 2.1 on page 12, preprocessing in section 2.2 on page 16, analysis in section 2.3 on page 23, before it rounds off with visualization in section 2.4 on page 32.

As seen from the introduction and background, there exist a number of techniques and tools to perform topic modeling. However, before going into the details of how it works and some of the math behind the methods, it is important to establish a basic intuition of the discipline and understanding of what type of results it provides. As explained earlier, topic modeling is used to explore large sets of text data. The goal is to get an overview of some of the most important aspects that are needed to summarize the texts. One might expect that the results will always show full or partially full sentences, but this is not the case. Topic modeling usually creates a list of lists of words with a number next to it as shown in Figure 2.1. The actual example is created from a data set of Australian news and is not related to the topic of this thesis.

At the beginning of each inner list, the numbers 0 through 9 indicate each of the ten unique topics received from the topic modeling. Each of the numbers within the inner lists, such as 0.008*australia, indicate the weights, also known as the probability, of that specific word belonging to that topic. These lists are ordered in a decreasing manner, where the first word is the most probable within a given topic. The length of the outer list is chosen by the user, where he or she sets a number of topics. The inner lists are the number of words to be displayed per topic, which is also chosen by the user. Based on each inner list, the user can create his or her own opinion of the theme and the topic name. The words of the inner lists are all words from the vocabulary created from the corpus (all documents within a text data set) of the data set to be analysed. However, as explained above, only the ones with the highest probability are shown.



```
[(0,
  '0.008*"australia" + 0.005*"year" + 0.005*"australian" + 0.004*"day" + 0.004*"new" + 0.003*"fire
" + 0.003*"people" + 0.003*"wicket" + 0.003*"$" + 0.003*"police"'),
 (1,
  '0.005*"metre" + 0.004*"day" + 0.004*"union" + 0.004*"win" + 0.003*"australia" + 0.003*"meet" +
0.003*"tell" + 0.003*"al_qaeda" + 0.003*"official" + 0.003*"event"'),
 (2,
  '0.007*"australian" + 0.006*"report" + 0.006*"force" + 0.005*"area" + 0.005*"australia" + 0.005*
"people" + 0.005*"al_qaeda" + 0.004*"unite_state" + 0.004*"afghanistan" + 0.004*"know"'),
 (3,
  '0.008*"australia" + 0.007*"australian" + 0.007*"government" + 0.005*"year" + 0.005*"people" + 0
.005*"man" + 0.005*"day" + 0.003*"police" + 0.003*"claim" + 0.003*"power"'),
 (4,
  '0.005*"australian" + 0.005*"australia" + 0.004*"fire" + 0.004*"government" + 0.004*"arafat" + 0
.004*"sydney" + 0.004*"year" + 0.004*"israeli" + 0.004*"people" + 0.003*"palestinian"'),
 (5,
  '0.005*"kill" + 0.005*"people" + 0.004*"palestinian" + 0.004*"israeli" + 0.004*"police" + 0.004*
"attack" + 0.004*"force" + 0.003*"australian" + 0.003*"group" + 0.003*"day"'),
 (6,
  '0.006*"force" + 0.006*"afghanistan" + 0.006*"australian" + 0.005*"people" + 0.004*"year" + 0.00
4*"report" + 0.004*"new" + 0.004*"taliban" + 0.003*"tell" + 0.003*"unite_state"'),
 (7,
  '0.008*"palestinian" + 0.006*"israeli" + 0.005*"australian" + 0.004*"company" + 0.004*"arrest" +
0.004*"police" + 0.004*"meet" + 0.004*"arafat" + 0.004*"government" + 0.004*"group"'),
 (8,
  '0.006*"people" + 0.005*"australian" + 0.005*"government" + 0.004*"year" + 0.003*"arrest" + 0.00
3*"afghan" + 0.003*"party" + 0.003*"new" + 0.003*"group" + 0.003*"security"'),
 (9,
  '0.006*"official" + 0.006*"israeli" + 0.005*"australian" + 0.004*"palestinian" + 0.004*"fire" +
0.004*"man" + 0.004*"early" + 0.003*"day" + 0.003*"people" + 0.003*"force"')]
```

Figure 2.1: *Example result from a topic model. The following list of lists is an example of how a typical result looks like when performing topic modeling. The result is based upon a set of Australian news. It shows ten topics and the ten most probable words within each topic.*

When performing topic modeling with LDA, one will get two matrices, which are known as word-to-topic probability matrix and document-to-topic probability matrix [24]. These explain the probability of a word being part of a topic and the probability of a document being part of a topic.

Topic modeling is a type of cluster analysis, which identifies similarities between various words. Visualizations made using the software package pyLDAvis, explained in section 2.4, will also help with the visual aspect of why topic modeling is a cluster analysis.

There is some wording used throughout the rest of the thesis which it is important to understand correctly. A *corpus* is a collection of *documents*. These *documents* are texts made of *words* also known as tokens which are items from a vocabulary. The vocabulary is unique for each analysis, as it is based upon the documents to be analysed.

2.1 Data collection

As explained in the background, there is an increased amount of unstructured data published daily. With this increase, there is a need for better automated data collection methods, in order to be able to detect the most important information from the unstructured data. Data collection is both time consuming and often times complicated as data are stored in many different formats. Text collection, which is a type of data collection where unstructured texts are collected, also known as web-harvesting, is no exception. Different online sources store their data differently which complicates the collection process. Software packages and tools are necessary to make the text machine readable. As a result, there exists a number of different ways to collect such data. The data collection possibilities explained in this thesis are web scraping, application programming interface (API) and the use of external companies to perform web-harvesting. These three ways of collecting data work slightly different in the way they gather data and the amounts being collected. However, they all have in common that appropriate file formats and the storing techniques are required both to ensure performance speed and easy data handling [25].

To get a brief overview of the three different methods, Figure 2.2 shows how they differ. Web scraping can be seen as a spider crawling the web while API can be understood as a connection with only a key between the the source of information and the user [25]. The source of information, in all three cases, are the websites where the information is collected from. Web scraping usually goes undetected, as the user in most cases collects data directly from the HTML code not making the owner of the data aware. API, on the other hand, is dependent upon an API key given by the owner of the web page. The API key is then used to request certain information from the website. External web-harvesting companies usually do not provide any details on their collection methods; they only allow the user to request information from them, using a search query. Then the external company collects the information on behalf of the user from websites they have in their repertoire. External web-harvesting can therefore be seen as a black box.

Before establishing a deeper understanding of data collection methods it is important to be aware of the different file formats. Collected information (documents) can be stored with several different formats and choosing the optimal format for the desired purpose will allow for faster processing and also easier data handling [25]. There are several file formats to chose from, such as JavaScript Object Notation (JSON), eXtensible Markup Language (XML), Comma Separated Values (CSV) and Portable Document Format (PDF), among others. However, the two major formats that one tend to use when performing data collection of texts are JSON, and XML.

Both of these formats have existed for more than a decade and have developed much over the years as a result of better and more advanced programming [25].

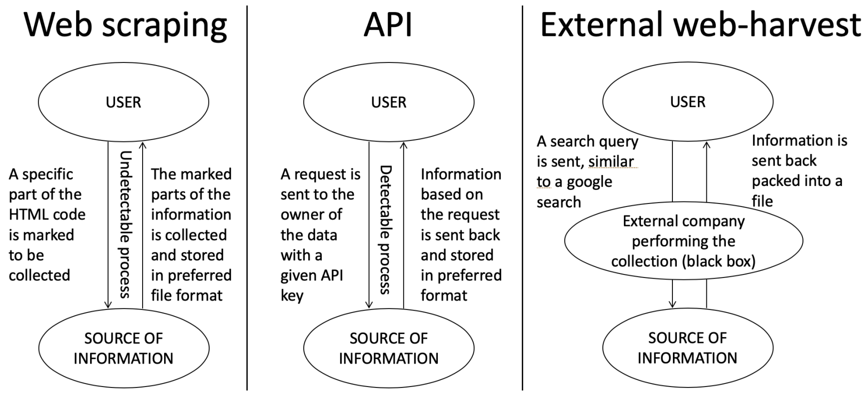

Web scraping
USER
A specific part of the HTML code is marked to be collected
Undetectable process
The marked parts of the information is collected and stored in preferred file format
SOURCE OF INFORMATION
API
USER
A request is sent to the owner of the data with a given API key
Detectable process
Information based on the request is sent back and stored in preferred format
SOURCE OF INFORMATION
External web-harvest
USER
A search query is sent, similar to a google search
Information is sent back packed into a file
External company performing the collection (black box)
SOURCE OF INFORMATION

Figure 2.2: *Graphical representation of text collection methods. The figure shows how web scraping, API and external web-harvesting companies differ in the way that data is collected. The term "Source of information" in each method represents the websites where data is collected from. More details on their differences can be found in the sub-sections below.*

XML used to be the preferred format for many years and was the go-to for most researchers. However, in recent years JSON has become vastly more popular for several reasons [25]. First, JSON generally uses fewer characters; this allows for faster parsing (analysis of the data stored) and less memory allocation [25]. Secondly, there has been a shift in web technologies allowing the JSON format to interact more smoothly with the web servers [25]. Lastly, JSON's formatting is easier to work with due to the setup of the file and its structure [25]. There exists a number of reasons to be aware of the file formats when one collects data.

2.1.1 Web Scraping

Web scraping can be performed with a number of different tools and programming languages. Python programming language is used for this thesis, and therefore the methods explained will be focused around methods implemented in various Python packages. Within Python there exist a number of packages such as BeautifulSoup, Selenium and Scrapy, among others [25]. They all serve different purposes and some are easier to work with, while others need more training. The goal of any web scraping is to receive HTML data from a domain name and then parsing the data to look for target information before storing it [25]. Web scraping is often thought of as a spider that crawls the web. The reason for this is that web scraping often happens undetected as it is a "automated process" of "stealing" information from websites via the HTML code [25]. There exist numerous books and articles written about the process of web scraping by itself, but there are few research articles discussing its use as a first step in topic modeling.

Web scraping has several benefits over both API and external web-harvesting companies. The user can access and collect any information available on the web. There are no limitations to the information one can ”acquire” as it is taken undetected and there are no rules to abide nor any API keys to pay or sign up for [26]. Web scraping has no limitations to the specific number of queries performed per day. This means that the user can perform repeated queries unlimitedly.

When this is said, it is important to include that there exist disadvantages, too. Collecting data with web scraping requires large server capacity from the website itself. As a result, some users have experienced to get banned due to the amount of server capacity used [26]. This happens when websites detect abnormal activity from the same IP address. In addition, web scraping is not optimal when one wants to scrape from several different websites. Websites are formatted differently and as a result one needs to write a custom encoding script per site that one wants to scrape. Websites also tend to change their structures which means that crawlers (scrap setup) need maintenance [26].

Web scraping can be implemented either manually or by using one of the existing packages, such as BeautifulSoup, Selenium and Scrapy, among others. As the following thesis does not use this as a methodology to collect data, the setup of neither the manual nor the packages will be explained. However, the book *Web Scraping with Python: Collecting More Data from the Modern Web* by Ryan Mitchell explains all there is to know very well [25].

2.1.2 API

For the purpose of this thesis, application programming interface does the same as web scraping, but differs in the way that data is collected and also the amount of available information. As the following description of API is based upon the book *Web scraping with Python: Collecting more data from the modern web*, there will not be any specific referencing within this section [25]. API is a standardized syntax allowing for communication between computers, even if the software differs. API can also be used in other contexts, such as communication between two programs within the same computer, but this definition is not used for data collection in this thesis.

Differently from web scraping, API requires that the website has an option for application programming interface, and this is not always the case. The collectable data depends strictly upon the owner of the website. The reason for this, is that the data available is restricted by the owner of the website, and the only way of accessing the information is through an API key. An API key is generated by the website owner and includes restriction on the type of information that can be collected. If a key is accepted, the requested information is sent back to the user in a preferred file format. As such, not all information can be collected.

Similar to web scraping, there exist benefits and disadvantages with API, which are often the opposite of web scraping. API is useful when there exists an application programming interface for a given purpose, but cannot be used when it does not exist. Differently from web scraping, API tends to have restrictions on the number of times data can be requested per day. This might, for example, result in poorer predictions in cases where there is a need of continuous data, such as temperatures throughout a day, a month or a year. In situations where there exists an API, one often sees that the responses are well-formatted, which results in less work for the user. Some APIs also need less attention to maintenance, as this is updated by the owner of the website. When this is said, APIs often include lots of unnecessary information, which needs to be temporarily understood and stored. This means that an API collects an entire set of information sent by the owner of the website. It further indicates that one has to decide whether the time commitment to split and understand the data is worth it, as well as whether the extra memory allocation is justifiable over web scraping.

2.1.3 External web-harvesting companies

As for the two methods above, there exists a third fully automated possibility where the user pays a third party to perform web-harvesting. Instead of setting up a manual script or using packages built into Python, one uses an external company or software to perform the entire process. The processes of performing either web scraping or API tend to be extremely time consuming [25], and the user also needs to know where to look for information. External web-harvesting companies, on the other hand, do this entire process based upon search queries. They find places with desired information and hereby attempt to reduce the time commitment for the user. For a person or a company who has a goal of being time effective, these external web-harvesting companies are an attractive option. However, one should be aware of some major drawbacks.

First of all, the number of such external web-harvesting companies grows rapidly as more and more companies from all types of businesses have understood the importance of text data. This results in an increased competition among the providers, but also an increased number of poor web-harvesting companies being established. Secondly, in most situations the user does not know anything about the providers of information. This may indicate a lack of valuable source criticism. In other words, the user can no longer make sure that the collected data comes from credible sources. External web-harvesting also limits the possibility for the user to extend their search query to other information providers. As the user has no idea about where the information is collected from, he or she won't have the option to add alternative sources of information. Nor will he or she be able to filter and remove data from specific websites.

Thirdly, the use of external web-harvesting companies can be very costly. Depending on the purpose of the analysis, it is of significant value to decide on a budget for data collection. Lastly, it is important to take into account that the collected information often includes many non-relevant piles of words. As a result, one needs to evaluate and understand the cost vs value of this extra collected information. This is mostly a concern when performing continuous data collection on a daily basis. The buyer of such gathered information also needs to split up and make memory allocation to store it. Having said this, the decision on whether or not to make use of an external web-harvesting company should be based upon a thorough investigation of cost vs value for the user.

There is a large number of different external web-harvesting companies specializing on various aspects of data collection. While some focus on data collection for news papers, others focus on business related data. There are also companies who collect data for research purposes. As the following study was not intended to compare different web-harvesting companies, only the use of Webhose.io will be explained. This was the company used for data collection within this thesis.

2.2 Preprocessing

To be able to perform any analysis on a set of data, there is a need to perform some preprocessing ahead of time. Preprocessing is like a ”cleaning” process of the data where the idea is to get rid of unnecessary disruptions, such as stop-words, punctuation and unnecessary symbols, among other. There exist numerous possibilities to clean data and as a result it is of great value to be focused on the correct purpose. As computers cannot understand human readable text, they need the text to be converted into numbers [24, p. 259]. The purpose for this thesis was to train a topic model and check how well a set of unknown texts could be explored. Therefore, this section will cover some of the most common preprocessing steps within topic modeling. However, not all preprocessing steps explained in this section have been used on the collected documents within this thesis. This section is written to identify suitable preprocessing methods in order to assure that decisions made had a theoretical backing. The selected methods will be explained in greater detail in the method chapter.

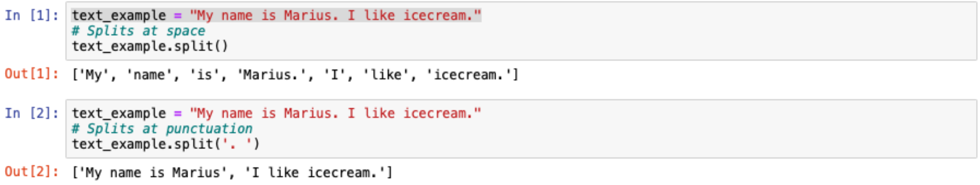
As introduced in the background section, there are many different types of texts of different lengths. There is also a large variation in the number of documents available about a specific topic. For example, there are millions of documents about politics, while only a limited number of texts about Shakespeare’s plays are available[27]. This complicates the use of topic models. As Tang et al. described in their study short documents, or too few documents about a specific topic, will limit the quality of the results when performing topic modeling [27]. Entire books are also known to be harder to analyse, as they consist of too many topics [27]. Topic models are also affected by misspelled words [28], abbreviations and variation in languages within a corpus, as they often tend to make topics harder to understand and evaluate. There are, however, ways of dealing with these issues such as term frequency-inverse document frequency (tf-idf), or reduction of vocabulary by some simple restrictions of occurrences within each document and across the corpus [29].

It is important to note that the order of the following theory section relates to the order of the preprocessing steps within this thesis. However, studies on topic modeling do not always agree on the order of the preprocessing steps, which is why this thesis tries to reason the order that was used within the method section.

2.2.1 Tokenization

Tokenization is often the first step of preprocessing when performing most text mining. This simply means to split an entire corpus into documents, which are further split into into separate words, terms or symbols, also known as tokens [30]. Tokens are usually separated by white space or full stop. This means that all continuous strings of letters in the alphabet will be identified as one individual token. When performing tokenization there are several ways of how to improve the quality of the tokens. It is possible to restrict the length of a token, remove unnecessary tokens such as punctuation, and lowercase all tokens, among others. These are performed to reduce the complexity of the corpus and increase the chances of getting more valuable results [30]. For topic modeling, it is important to remove all tokens existing of only punctuation as it otherwise will affect the resulting topics. An example is by including punctuation characters within the inner lists of topics. Also, lower-casing all words is important due to the fact that none-lower-casing will make the computer differentiate two of the same words. For example, ”cat”, ”Cat” or ”CAT” will otherwise be identified as three different words [30].

Possibly the simplest way of tokenization, is to use a built-in function in Python such as *.split()*. This method splits words based on the spaces in a given document or alternatively a predefined splitting parameter. This is shown in Figure 2.3. As seen in the figure, the top example is two sentences split based on white space, while the lower example is a split based on punctuation.

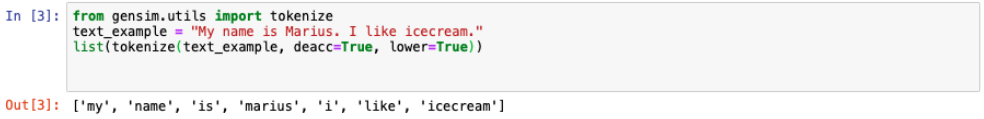


```
In [1]: text_example = "My name is Marius. I like icecream."
        # Splits at space
        text_example.split()

Out[1]: ['My', 'name', 'is', 'Marius.', 'I', 'like', 'icecream.']

In [2]: text_example = "My name is Marius. I like icecream."
        # Splits at punctuation
        text_example.split('. ')

Out[2]: ['My name is Marius', 'I like icecream.']
```

Figure 2.3: *Simple tokenization. The following figure shows how simple tokenization is performed with a Python built-in function.*

An alternative way of performing tokenization is using a package built in Python. Here there is a number to choose from, such as **NLTK**, **spacy** and **Gensim**. Figure 2.4 shows how Gensim is applied with respect to the same example as above.

However, it also shows how easy it is to remove punctuation and lower-case all words in the same operation.



```
In [3]: from gensim.utils import tokenize
        text_example = "My name is Marius. I like icecream."
        list(tokenize(text_example, deacc=True, lower=True))

Out[3]: ['my', 'name', 'is', 'marius', 'i', 'like', 'icecream']
```

Figure 2.4: *Gensim tokenization. The following figure shows how Gensim tokenization is performed with Python. It also shows how easy removing punctuation is, while lower-casing all words at the same time.*

2.2.2 Stemming or lemmatization

The next step of prepossessing, in most cases, is stemming or lemmatization. Both methods serve somewhat the same purpose. Stemming is a process where words (tokens) are reduced to a common stem form while lemmatization reduces words to their base or dictionary form [31]. Stemming removes inflections and derivational suffixes to reduce words into their stems. This allows for the vocabulary to be reduced as multiple forms of the same word is no longer included. However, there is a risk of under- or over-stemming. This means that two words of the same conceptual group is kept differently, or the other way around, that two words of the same conceptual group is merged into one word [31]. An example of under-stemming could be the words ”running” and ”ran”, where both ”run” and ”ran” will be kept within the same conceptual group. However, they mean the same. An example of over-stemming, on the other hand, could be ”new” and ”news”, where both become ”new”. Here the two conceptual groups are diminished to one. Stemming may be performed with many different algorithms, where most of them are based on simple rules which remove endings of words.

Lemmatization uses an external vocabulary (there exist numerous different options) and performs a morphological analysis of the words within the corpus. A morphological analysis, in this case, means that the method identifies a relationship to other words. Using this relationship, the words are reduced to their dictionary form [31]. Lemmatization also has the benefit of being able to check the part-of-speech category per word to identify the correct dictionary form. Part-of-speech categories are, for example, nouns, pronouns, verbs, adjectives and adverbs. Some of the predefined methods also have checks for synonyms of the same word [31].

Similar to tokenization, there are packages built for Python to perform both stemming and lemmatization, but neither of them will be explained in greater detail within this thesis. There will, however, be an explanation as to how it is performed for the purpose of this thesis within section 3.2.

2.2.3 Stop-words

There is also a need to remove stop-words. Examples of stop-words are; *"I"*, *"and"*, *"the"* and *"are"*, among others. Removing stop words is essential to the quality of the topics created within analysis [32]. The reason for this is that these words tend to be over-represented in the topics, as they are very common in any English sentence [32]. Stop-words are, for the most case, used as fillings and structural benefits for the reader. However, it should be checked which words are removed as some removal might accidentally get rid of important data. Removing some stop-words might completely turn around the meaning of a sentence. An example is how *"not like"* becomes *"like"* after stop-words have been removed, as *"not"* is a stop-word.

Two frequently used Python packages for stop-word removal are spacy and NLTK. Both aim at removing stop-words based upon a predefined list of such words. The two methods have their own list of words which are greatly overlapping, some words may be unique to the package.

2.2.4 Converting text to numbers

The next step of preprocessing, in most applications within text mining, is to convert the words or tokens into numbers which can be used for computations by a computer. As stated in the book *Python Machine Learning* by Raschka and Mirjalili, text or words need to be converted into a numerical form before they can be passed on to a machine learning algorithm [24, p. 259]. As a result, there are several different ways of doing this. The ones explained in this thesis are bag-of-words, tf-idf and word embedding with word2vec. The focus will, however, be on the bag-of-words method as it is the numerical feature vector needed to perform LDA [24, p. 274].

Bag-of-words can be implemented simply by creating a vocabulary of unique tokens (words) from the entire corpus [24, p. 259]. Further, one constructs a feature vector per document that exists within the corpus. The vectors contain information on the number of times each word occurs within the specific document [24, p. 259]. These feature vectors are known to be sparse vectors, as they mostly contain 0's. This is a result of the size of the vocabulary of the corpus which in most cases are much larger than the vocabulary of a given document.

In order to go into more detail on how this actually happens, an example is helpful. Imagine that one has a corpus consisting of 100 documents with a total vocabulary of 1000 unique words. These 1000 words are then the vocabulary of the corpus [24, p. 259]. Next, the words are put into a table where the frequency of each word is counted within the entire corpus [24, p. 259]. Further, binary bag-of-words vectors or simply bag-of-words vectors are created. These differ slightly in the way that binary bag-of-words vectors (explained as one-hot encoding by Bengfort et al. [33, p. 59]) do not account for the frequency of each word within a given document; it only indicates if the word exists or not. Bag-of-words, on the other hand, accounts for the occurrences of each word within each document [24, p. 259]. Both binary- and bag-of-words then create vectors where each number within a given vector represents the existence (value set to 1) or frequency of a given word from the vocabulary. The values are set to 0 if the word does not exist within the given document [24, p. 259]. As understood from this, the vectors are sparse vectors with mostly 0's and just a few 1's (or frequency counts) for words existing within the given document.

Due to the sparsity of bag-of-words, the method is known to take up very much memory allocation if the corpus is large [24, p. 259]. For some computers this create problems as they cannot handle the amount of data. Further, bag-of-words vectors are known to exclude the importance of certain words as they all have the same weighting [24]. Term frequency-inverse document frequency could then be used to account for this, by allowing certain words to get a higher weight or completely exclude words that are not important. Tf-idf is explained in greater detail within sub-section 2.2.5.

As introduced in the beginning of this section, there is an alternative to bag-of-words and tf-idf. This is known as word embedding. In word embedding each word is assigned a unique vector representation where the goal is to keep word similarities by assigning a similar vector to related words. One of the most prominent word embedding methods is word2vec which is an algorithm introduced by Google in 2013 as an alternative to the bag-of-words method [24, p. 274]. This is an unsupervised algorithm based on neural networks attempting to learn relationships between words [24, p. 274]. Words with similar meanings are gathered into similar clusters via vector-spacing using simple vector math [34]. Word2vec preserves the relationship between words. An example would be that a "cat" and a "tiger" most likely would get a more similar vector representation in comparison to the vector representation of a "cat" and a "computer".

Word2vec exists with two different architectures where both serve for somewhat the same purpose [34]. They both aim at creating word vectors, however, their ways of training the model to create these vectors differ. The first model is known as continuous bag-of-words (CBOW) model as the order of the words do not influence the creation of the vectors [34]. CBOW uses a neural network with hidden layers to predict the new word. More information can be found in *Efficient Estimation of Word Representations in Vector Space* by Mikolov et al., who are the inventors of this method [34].

Even though the math will not be explained, there exists a simplified representation of the model as shown in Figure 2.5. The figure shows both CBOW and skip-gram and how they differ. CBOW uses the surrounding words as input. By surrounding words, it is meant words appearing directly in front and after the word to be predicted. According to Mikolov et al. CBOW uses four prior and four future words of the word to be predicted [34]. Skip-gram, on the other hand, does the opposite. It uses an existing word as input, and predicts the surrounding words. Neither of the models show the hidden neural network layers of the actual model, which are an essential part of both models.

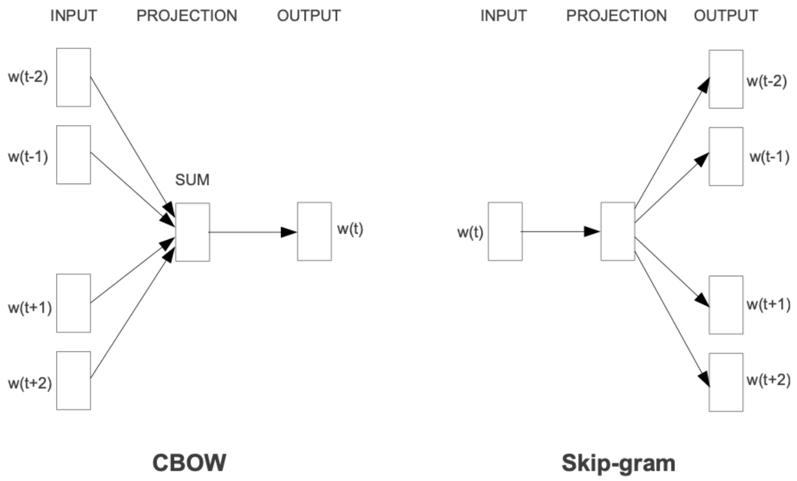


Figure 2.5: *Graphical representation of the CBOW and skip-gram models [34]. The following representation is taken from Mikolov et al.'s work on Efficient Estimation of Word Representations in Vector Space. It shows how CBOW predict the new word based upon the existing words. It uses both prior and future words of the word that is being predicted, while skip-gram uses the existing word to predict the surrounding words.*

2.2.5 Term Frequency-Inverse Document Frequency

Term frequency-inverse document frequency (tf-idf) is used as a method to optimize the importance of words in documents within a corpus [24, p. 261]. The method adjusts the weights of the words according to its frequency within a document and across the corpus [24, p. 261]. The importance increases proportionally with the number of times it occurs within a document, while being offset by the number of documents it occurs within. In other words, words occurring frequently across the entire corpus get low weights, while frequent words in few documents get higher weights. Tf-idf is therefore a product of the term frequency and the inverse document frequency [24, p. 261] as seen in equation 2.1, where *tf* is the number of times a term *t* occurs in a document *d* [24, p. 261].

$$tf\text{-}idf(t,d) = tf(t,d) * idf(t,d) \tag{2.1}$$

In the above equation *tf(t,d)* is the term frequency within a document calculated as explained in the bag-of-words model (2.2.4), while *idf(t,d)* is the inverse document frequency which is the number of documents the word occurs within. *Idf(t,d)* can be calculated as seen in equation 2.2, where *nd* is the total number of documents in the corpus, while *df(d,t)* is the number of documents *d* that contain the word *t*.

$$idf(t,d) = log\frac{n_d}{1 + df(d,t)} \tag{2.2}$$

In equation 2.2 adding the constant is optional and simply serves to create non-zero values in situations where words occur in all documents [24]. Log is used to make sure that low document frequency does not get a too large weight [24]. If this is performed on the same set of documents as used for the regular bag-of-words model, one will find that words occurring multiple times in most texts will acquire less importance, in the form of a lower weight [24]. One will also find that words occurring only a few times within a smaller sub-section of the data will get higher importance. These weights will be numbers between 0 and 1.

There exists a similar way of making sure that only words with higher importance are kept in the vocabulary. This is performed by creating restrictions to the vocabulary by only allowing words that occur at least a certain number of times per document, while they at the same time only occur within a given percentage of documents [29]. Deciding on the size of the vocabulary is essential to the speed of the method. Reducing the size of the vocabulary reduces the information that can be presented but often removal is necessary to improve the speed of the model.

2.2.6 Bi- and tri-grams

Yet another step of preprocessing is the option to create bi- and tri-grams. These are situations where words occur often enough next to each other, indicating that the word is supposed to be identified as one word [35]. An example of such is *"New York"*, even though *"New"* and *"York"* have their separate meanings; the meaning is different when the words are placed next to each other. These "double or triple words" are often very important to detect, keep together and include into the vocabulary as it will change the interpretation of a result such as a topic [35].

To create bi- and tri-grams marginal and conditional word counts are determined. Marginal word count is the total number of times a given word *i* has occurred within the corpus [35]. Conditional word count, on the other hand, is the number of times a word *i* has followed a word *j*. Using these numbers calculations can be performed with marginal and conditional probabilities, as well as marginal frequency estimator [35]. Further details on the mathematics can be found in *Topic modeling: beyond bag-of-words* by Hanna Wallach.

2.3 Topic modeling

The next stage is to analyse and create topics. The preprocessed and cleaned data will be analyzed. Several different methods can be used to create such topics, and the ones explained here are Latent Semantic Indexing (LSI) and LDA, as these are some of the most used models among researchers as well as the industry. The most fundamental ideas and some math will be discussed. However, as LDA is the preferred and most used model among the two [3], this will be the main focus.

The goal of any machine learning model is to find the best fit of the data. This is also the case of a topic model. Here the idea is to explore and give an overview of the aspects of what is being discussed in the documents within the corpus. As such, the point is to get the most information possible and it is therefore important to be able to cover all the content of the corpus. An automatic measure of the amount of information received and the human-interpretability of topics would be optimal, but as this does not exist, researchers are working on finding the best method [36]. Perplexity or held-out likelihood are two well known and tested methods to measure the fit of a topic model. They are, however, shown not to be correlated with human judgements of topic quality [37]. As a result the methods are not used in this thesis and therefore out of scope of this thesis. More information on the methods can be found in the article ”*Reading tea leaves: How humans interpret topic models*” by Chang et al. [37].

There are some alternatives to measure interpretability of topic models, such as word intrusion and coherence. Word intrusion is a method where a human searches for non-related words. Word intrusion is very labor intensive and needs expert knowledge within the field of research of a given study, as they need to identify the words which do not belong in a given topic. Coherence, on the other hand, is an automated process where the method identifies semantic similarities among the words across topics. Both methods will be explained in greater detail after the sub-sections of LSI and LDA.

2.3.1 Latent Semantic Indexing

Latent Semantic Analysis was invented in the early 1990’s to improve detection of relevant documents based on simple search queries [13]. The method aimed at finding underlying latent semantic structures using statistical techniques. Deerwester et al. used singular-value decomposition (SVD) on a matrix of terms and documents, such as bag-of-words, to construct a ”semantic” space where terms and documents associated to one another were placed closely. SVD allowed for ignoring less important influences while at the same time reflect major associative patterns, such as word similarities. Further, it resulted in the detection of less important terms to a given document, to actually appear close to that same document after all. This was a result of the major pattern among the rest of the data. These new positions in space were then used to identify topics within the data set

[13].

To get a better understanding of how LSI works, a slightly more mathematical explanation is needed. The method starts with a matrix of documents by terms (words) which is the input matrix. To create such a matrix bag-of-words method can be used [13]. Here the frequency of each word in the vocabulary is counted per document. Alternatively, binary bag-of-words can be used where the existence of a word is identified as 1, while 0 means that the word does not exist within the document.

Further, this matrix is analyzed with SVD to identify the latent semantic structure and reduce the dimensionality of the input data [13]. Equation 2.3 shows how SVD is defined. Matrix *A* is the input matrix of m documents and n terms. Matrix *U* is the left singular vectors of m documents by *r* topics. Matrix Σ is the diagonal singular values of $r \times r$, where *r* is the rank of matrix A. This matrix identifies the strength of each topic. A diagonal matrix means that all values are zero except the diagonal. In this situation the diagonal values are also sorted from largest to smallest. Matrix V^T, is the right singular vectors of *r* topics by n terms [13]. It is important to be aware of the fact that U and V are orthogonal to one another.

$$A_{[m*n]} = U_{[m*r]} \Sigma_{[r*r]} (V_{[n*r]})^T \tag{2.3}$$

Truncated SVD is further used. This means that only the vectors of U related to the largest values of Σ are kept [13]. The rest are discarded. A new simplified version of equation 2.3 becomes 2.4. Here $\hat{A}$ is an approximate decomposition which is close to the actual A. The approximate $\hat{A}$ is of rank *t* topics, replacing each of the above *r's* with *t's* as the new rank is *t*. In other words, setting the number of topics truncates the SVD and gives an approximate decomposition of the input matrix *A*. As explained above, setting the proper number of topics is somewhat complicated as one want a topic number large enough to fit all real data, but at the same time small enough to exclude unimportant details.

$$A \approx \hat{A} = U_t \hat{\Sigma}_t (V_t)^T \tag{2.4}$$

The actual matrix calculations and more details on the interpretation techniques are beyond the scope of this thesis. However, it is important to understand that the decomposition can be used to further investigate the topics. The *U* matrix identifies the relationship between each document and the topics, while the V^T identifies the relationship between the words (vocabulary) and the topics.

In 1999 Thomas Hofmann proposed a novel approach to LSI which he called Probabilistic Latent Semantic Indexing (pLSI) [38]. According to Hofmann, his method had an improved statistical foundation, as it was based on the likelihood principal and a proper generative model. Neither of these will be explained as they were not used within this thesis. He explicitly focused on the fact that his model allowed to understand polysemous by being able to distinguish between meanings and different word usage. Polysemous means that the same word, phrase or symbol can have several different meanings, such as *"get"* can mean *"become"*, *"procure"* or *"understand"*. More details about pLSI are given in, *Probabilistic Latent Semantic Indexing* [38].

2.3.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation is "a generative probabilistic model for collections of discrete data such as text corpora" [14]. The method was proposed in 2003 and builds on some of the ground foundations of pLSI which was proposed just a few years earlier by Hofmann [14]. LDA is a cluster method that tries to group words occurring frequently together across a corpus [24, p. 275] and can be compared to fuzzy K-means clustering. K-means is a vector quantization method where the goal is to find *k* clusters in which each observation only belongs to a cluster based on the nearest mean. Fuzzy K-means works similarly in the way that it discovers clusters, however, any given observation can belong to several clusters. LDA is called a mixed membership model which is why it is similar to fuzzy K-means and not only K-means.

Natively, LDA is an *unsupervised* machine learning method where the algorithm infers patterns without reference to known outcomes or labels. A *supervised* machine learning method, on the other hand, is one where the model has been trained on known labels. The LDA method can be used to detect underlying structure of the data. This fits topic modeling, as the idea is to be able to explore an unknown data set while getting some meaningful insight of patterns and structures. There does, however, exist supervised models such as Labeled-LDA [39] and supervised-LDA [40], but these will not be explained in further detail.

Before diving deeper into the math behind LDA, it is important to have a basic understanding of the method. As explained in the introduction, it is a method that allows unobserved groups of similar data to be explained by a group of words known as a topic. This means that an LDA model takes documents as input. This set of documents is known as a corpus. The corpus is preprocessed before it is sent into the LDA model represented as a bag-of-words. The model then outputs a list of lists of words which are the actual topics as seen in Figure 2.1.

Figure 2.6 shows a well known illustration of how topics are generated. The illustration is created by David Blei, one of the founders of LDA for topic modeling [8]. As shown on the figure (just the text document in the middle), each document exhibits multiple topics shown with different colors. The colors identify separate "topics". Imagine that all words are colored and stop-words are removed. Then without having read the actual document it should be easy to detect the overall themes (these are the topics) just by reading a few words of each color. LDA takes this intuition and creates a probabilistic model of text [8].

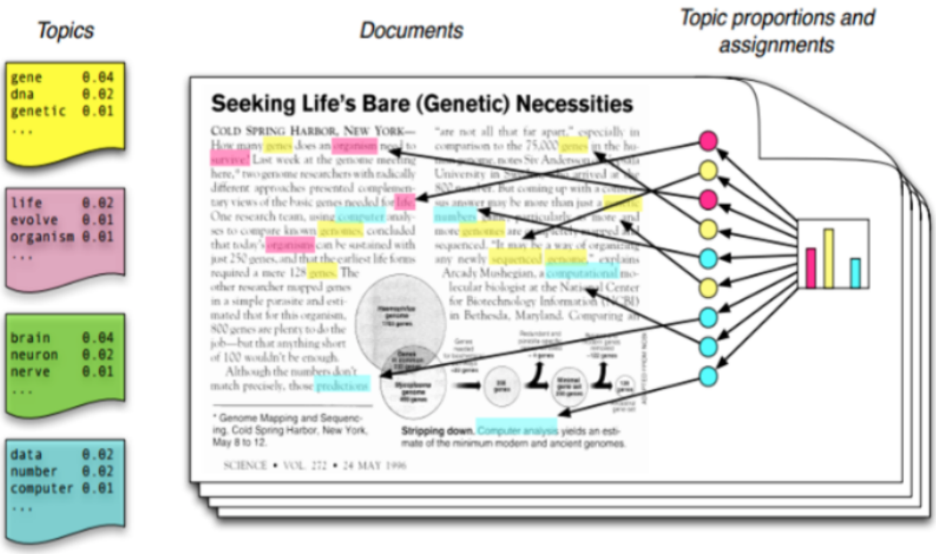


Figure 2.6: Illustrative LDA document generation process. The figure shows a well known illustrative example of how topics are found by using LDA. The left most boxes are the actual topics, while the text is an example of a document within the corpus. The colored dots and the histogram is the distribution of topics used for each iteration of training. The figure is used in numerous articles and research papers and, among them, Blei's lecture [8]

Now, looking at the entire figure, it is easier to understand the sequence of LDA. The topics to the left are distributions of words based on the vocabulary created from the corpus. The number of topics is fixed and set by the user. Each number next to the words within the topics indicates the probability of that word to the given topic.

LDA further assumes that each document arises as follows. First, the model chooses a random distribution over topics. This is shown as the histogram to the far right. Then for each word within the document, a color is chosen which identifies the closest interpretation of the word (topic to be placed within). This is repeated for every word within the document. It creates the document itself. The entire process is then repeated multiple times, where each word within a given document is assigned to a new distribution over topics. Throughout the process, the number of topics stay the same, while the amount that document exhibits to a given topic changes. This type of model is called a mixed membership model which is why it is similar to fuzzy K-means and not only K-means. A given topic is used to build several documents and a word is used to build several topics.

Latent Dirichlet Allocation can further be explained more mathematically. LDA finds a collection of words which represent topics, assuming that a corpus consist of documents of different words. The method uses bag-of-words (2.2.4 on page 19) as the vector representation of the corpus and outputs a document-to-topic matrix as well as a word-to-topic matrix [24, p. 275]. Bag-of-words neglects the order of words in a document, known as the assumption of exchangeability [14]. The same assumption is present across documents within the corpus; the order of documents can be neglected [14]. These assumptions build the foundation of the LDA model described in this thesis; ”mixture models that capture the exchangeability of both words and documents” [14].

To easier dive into the math, a graphical model helps. Figure 2.7 shows the different parts of the model. The nodes are random variables where the shaded one is observed and the unshaded ones are hidden [8]. The edges indicate dependencies, while the plates indicate replicated variables [8]. Replicated variables are repeated entities [8]. The outer plate represents the documents while the inner plate represents the words and their position within the documents. To be more specific and make a connection to the illustration in Figure 2.6, the *Dirichlet parameter* and the *Per-document topic proportions* are shown as the histogram. Each word within the text is placed within any of the topics per iteration of the training. The *Per-word topic assignment* is shown as the colored buttons, which means that this is where each word is assigned the best fit topic. The *Observed word* is the actual text document, which has been preprocessed. The combination of *Per-topic word distributions* and the *Topic parameter* is the list of topics shown to the far left. These are the ones that are used per iteration when a distribution of topics are picked out.

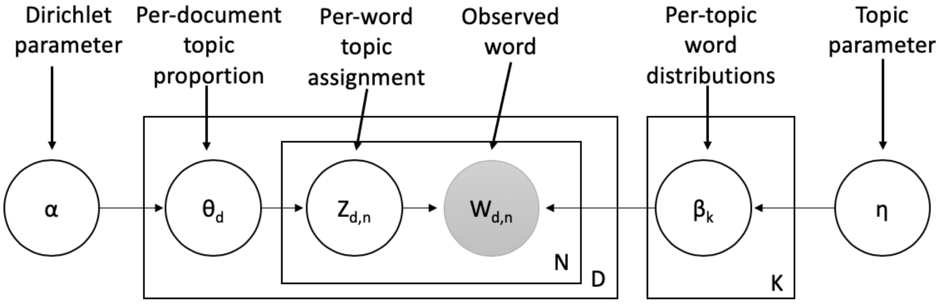

Dirichlet parameter
Per-document topic proportion
Per-word topic assignment
Observed word
Per-topic word distributions
Topic parameter
α
θ_d
Z_d,n
W_d,n
N
D
β_k
K
η

Figure 2.7: *Graphical model of LDA. The following model is a replica of the graphical model presented by David Blei in his guest lecture at the University of Edinburgh [8]. The model shows the parts that go into creating topics using LDA. The shaded node is observed, the unshaded nodes are hidden, the arrows indicate dependencies and the boxes indicate replicated variables.*

The Greek and Roman letters all indicate different parts of the model. The below list explains their reason to be part of the model [8].

- *D* is the total number of documents in the corpus.
- *N* is the number of words within a given document.
- *K* is the number of topics and this number is set by the user.
- α is the Dirichlet parameter vector which affects the distribution over topics chosen for each iteration. A large value results in an even distribution while a small value results in a distribution favoring certain topics.
- η is the Dirichlet topic parameter vector. This is the Dirichlet distribution which affects the distribution of words within each topic. A large value results in an even distribution while a small value results in a distribution favoring certain words.
- $W_{d,n}$ is the observed words within the documents. More specifically it is the *n*th word in the sequence within the *d*th document, where n goes from 1 to N and d from 1 to D.
- $Z_{d,n}$ is per-word topic assignment of each of the *n*th word within the *d*th document.
- $\boldsymbol{\beta_k}$ are vectors representing each of the *k*th topic where k goes from 1 to K. In other words, each vector $\boldsymbol{\beta_k}$ is the distribution over the full vocabulary and this sums up to 1. These vectors are put together into a matrix which is

what makes the word-to-topic probability matrix. This is one of the outputs from an LDA model

- $\boldsymbol{\theta_d}$ are vectors representing the per-document topic proportions for each of the *d*th document, where d goes from 1 to D. In other words, how probable a given topic is for each document in the corpus. The combination of these vectors creates the document-to-topic probability matrix. This is a second output from an LDA model.

Based on the graphical model one aim at estimating the hidden variables, in other words, trying to compute their distributions based on the given documents. The goal is to estimate the probability of topics, proportions and assignments given the words shown as $P(topics, proportions, assignments|words)$. This is performed by first choosing a distribution $\boldsymbol{\theta_d}$ based on the Dirichlet distribution α. The Dirichlet distribution is a vector that always sums up to 1 and is used to explain the probability distribution of the θ probability distribution [14]. Dirichlet distribution is beyond the scope of this thesis. However, the effect of different α values are important. A small value means that the document will have a few outstanding topics while a high α value means that the documents are a mixture of most topics [14].

Further, a topic $z_{d,n}$ is assigned from the $\boldsymbol{\theta_d}$ for each of N words within the given document. This is a latent variable. Finally, words $w_{d,n}$ are sampled and conditioned on the $z_{d,n}$ topic. In other words, this means that $\boldsymbol{\theta_d}$ explains the probability to which a topic *i* explains a given document *d*. The math is beyond the scope of this thesis.

β represents the topics and is a matrix created from the $\boldsymbol{\beta_k}$ vectors where each row is an unique $\boldsymbol{\beta_k}$ vector. Within the β matrix each row is a topic and each column is a word within the topic, as shown in Table 2.1 [41]. The values within the matrix represent the probability of a topic *i* containing the word *j*. Each $\boldsymbol{\beta_k}$ vector is the distribution over the full vocabulary and this sums up to 1. Larger values indicate higher probability of the specific word to be included in a given topic, while a zero value indicate that the word is not likely to have importance within the topic. The words are usually evenly distributed, but this can be changed to favor certain words of a particular topic [14]. This distribution is based on the η which is a Dirichlet distribution parameter similar to α. The only difference is that this does affect the word distribution instead of the topic distribution and the same technique can be used here [14]. A low value will favor just a few words per topic, while a high value will result in an even distribution. Both the η and the α values, can be set manually. It is, however, only needed when there exist previous knowledge about the word and/or topic distributions respectively.

Based on the above one can see that, given a set of documents one find a vocabulary. Then LDA is used to create an θ document-to-topic probability matrix which is a distribution of topics per document and also a β word-to-topic probability ma-

Table 2.1: *β matrix for LDA. This is a visualization of the β matrix which is the result when performing LDA. Each of the $\boldsymbol{\beta_k}$ vectors put together creates this matrix. The result is usually visualized as a list of lists instead, similar to Figure 2.1 on page 11 in the beginning of this theory chapter.*

Word probabilities for each topic							
	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7
Topic 1	0	0.1	0.5	0	0.3	0.05	0.05
Topic 2	0.7	0.05	0.05	0.05	0.05	0	0.1
Topic 3	0.3	0.3	0.2	0.1	0.03	0.07	0

trix which is a distribution of words per topic. These two can then be analysed to explore the content of the given documents, both each document specifically and also across all documents. There is, however, one issue with the results, and that is how one identifies the quality of the topics found as well as to what degree the topics cover the themes addressed within the documents. This comes down to finding the perfect number of topics.

2.3.3 Validation of topics

As briefly explained in the beginning of this theory section, there exist numerous of ways to identify the quality of topics found, to be able to decide the proper number of topics. The goal is to find human-interpretability topics that covers most of the content within a given corpus. When this is said, there is no golden standard strategy for the optimal approach and different research disagree. The one and only thing most researchers seem agree upon is that expert knowledge work [42] is needed, but it is tedious.

Word intrusion is a possible way of checking the semantic coherence of a topic known as the. This means to check if there is a relationship among the words within the topic. When doing this, taking a topic and add a word from a different topic which is not probable for the topic where the word is added [37]. Then if it is easy to pick out miss placed word (the intruder) it indicates that the topics have a high semantic coherence. On the other hand, if the words cannot be easily spotted it indicates the opposite. The method is very tedious as one need humans to make an understanding of words that fit and words that do not fit [37]. The actual word is picked out by taking the five most probable words from a topic, and then picking a word among the ones with low probability within the same topic. It is beneficial if the word with low probability from within the topic has a high probability within a different topic [37].

Topic intrusion is an alternative where the aim is to identify if the mixture of topics for a given document agrees with human judgment [37]. When doing this, the person is shown the title and a small section of text from the document, along with four topics where each is represented with the eight most probable words. Three of these topics are the topics with the highest probability within the given document while the last topic is randomly chosen among the low probability topics. Once again the person is supposed to pick out the non-relevant topic among the four [37].

The only concern with both of the above methods is that they are time-consuming and often experts are needed to be able to identify word intruders or topic intruders. Imagine doing this if the result consist of 100 topics. If humans are to work through hundreds of documents, the work becomes increasingly unsustainable. As a result, there exist an alternative which is an automated process of finding the topic coherence.

Topic coherence can be calculated with several different algorithms such as pairwise pointwise mutual information (PMI), log conditional probability, WordNet hierarchy and distributional similarity, among others. However the method explained in this thesis is based on *Exploring the Space of Topic Coherence Measures* by Roder et al. This is the methodology implemented with Gensim and also the method that gave the highest correlation with human-interpretability [36].

Their method is based on a composition of parts from already existing methods. They labeled the new method C_V. The dimensions of their model are [36]:

- The first dimension is the segmentation of words into word pairs.
- The second dimension is the word pair probabilities
- The third dimension is how strongly word sets support one another.
- The last dimension is the aggregation functions used to aggregate the scalar values from above into an overall coherence score.

The segmentation is found by setting $W = W_1, ..., W_N$ where W is a set of top N most probable words within a given topic. Each word $W' \in W$ is paired with all other words $W^* \in W$ giving S_i, while S then is the set of all individual S_i's [43]. An example is, given $W = w_1, w_2, w_3$, then a pair is given as $S_i = (W' = w_1), (W^* = w_1, w_2, w_3)$. The same goes for every word within every topic.

The second dimension, measuring the probabilities of word pairs can be found with Boolean document calculation [43]. This is found by taking the number of documents where a single word (w_i) or a word pair (w_i, w_j) occurs, and dividing it by the total number of documents. As this method ignores frequencies and distances among words, Roder et al. implemented Boolean sliding window [36]. Their method creates a new virtual document when sliding over the documents at a rate of one token per step. Each window is of size *s* such that a document d with words *w* results in a virtual document $d'_1 = w_1, ..., w_s$ and $d'_2 = w_1, ..., w_{s+1}$

and so on. The new probabilities are then calculated from the virtual documents to capture the word proximity [36].

The third dimension, measuring how strongly words support one another is based on similarity of the words. This means how similar W' and W^* are in comparison to the rest of the words. The similarities are found by setting W' and W^* as context vectors [43]. An example of such is shown in equation 2.5, where NPMI is the normalised pointwise mutual information calculation, which will not be explained in greater detail [43]. γ is used to give more weight to higher NPMI values. Such vectors are created for all word pairs giving a set of $\vec{v}(W')$ and $\vec{v}(W^*)$.

$$\vec{v}(W') = \left\{ \sum_{w_i \in W'} NPMI(w_i, w_j)^\gamma \right\}_{j=1,\ldots,|W|} \tag{2.5}$$

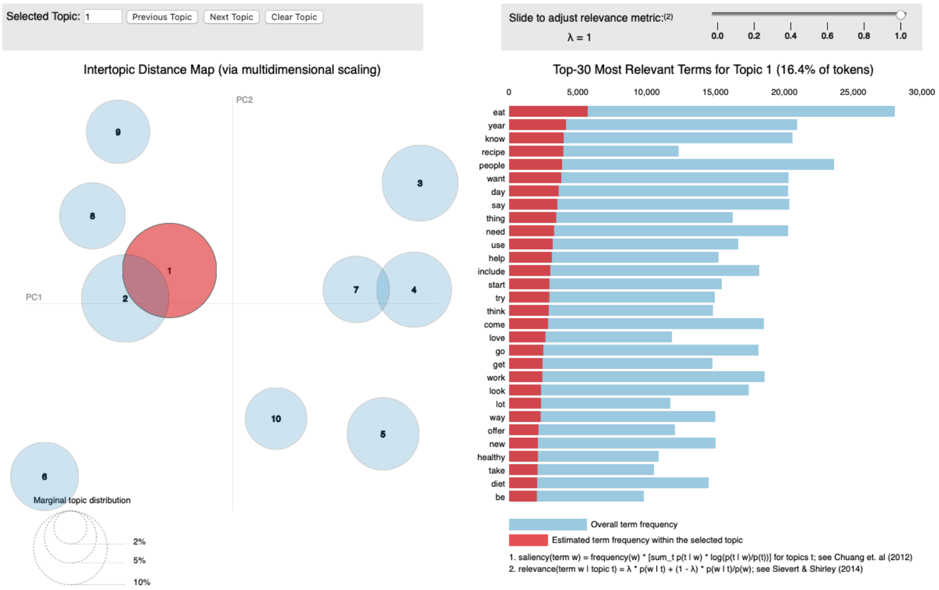
Further, the actual similarity is found by calculating the cosine vector similarity among all of the above vectors. This is shown in equation 2.6, where $\vec{v}(W') \in \vec{u}$ and $\vec{v}(W^*) \in \vec{w}$ [43].

$$\phi_{S_i}(\vec{u}, \vec{w}) = \frac{\sum_{i=1}^{|W|} u_i * w_i}{||\vec{u}||_2 * ||\vec{w}||_2} \tag{2.6}$$

The last dimension is calculating the final coherence score. This is found by calculating the arithmetic mean of all of the above similarity values of ϕ. By arithmetic mean, all values are added and then divided by the total number of values.

2.4 Visualisation for interpretation

The very last section of the automatic part of a topic model is visualization of the results. As shown in the beginning of this theory chapter in Figure 2.1 on page 11, the list of lists are not easily interpretable as they include all weights and words next to one another. To the benefit of the end user, LDAvis was created by Sievert and Shirley in 2014. They proposed a tool (R package) to create interactive visualizations of topics found from LDA [44]. The package allowed users to get a global view of the results, while at the same time allowing for deeper inspections of terms with high probability (weights). Further this package has been implemented into Python as pyLDAvis. An example screenshot from an early result is shown in Figure 2.8. The figure is used as an illustration to better explain the different options within the visualization package.



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 2.8: *pyLDAvis example visualization. This is a screenshot from the interactive pyLDAvis package. It shows an example result, where the words and topics are not relevant for the purpose of this explanation. The left panel shows the global view of the topics, while the right panel shows the 30 most probable words within the the selected topic (topic 1 in the illustration).*

As seen in Figure 2.8, the left panel is the global view of the topics. The centers of each topic is determined by the distance between each topic before multidimensional scaling is used to project the distances onto two dimensions [44]. The distances indicates the similarity among the topics. As seen on the plot, the topics are projected onto two dimensions. These are known as the principal components, *PC1* and *PC2*. Principal component analysis (PCA) is used as it allows for dimensionality reduction. It identifies patterns based on correlations between features. PCA aims at finding orthogonal axes (principal components) in directions of maximum variance in high dimensionality and project it onto a new subspace of equal or fewer dimensions, compared to the original one [24, p. 142]. These new principal components can be used to visualise the original high-dimension data in a 2D space.

Each of the circle sizes are explaining the prevalence of the topics compared to one another. In other words, the sizes represent the percent of documents which are explained by a given topic. The right panel, on the other hand, gives a list of the 30 most probable words (words with the highest weights) within the selected topic [44]. In other words, it represents the most useful words for interpreting a given topic. The two colored bars, give the user both the overall frequency of the word within the corpus (blue part) as well as the frequency of the word within the documents (red part) which are based upon the selected topic [44].

The two panels are linked interactively, which means that the user can pick a topic to the left and the most probable words will show on the right. In addition, the user can pick a word to the right and reveal the conditional distribution of topics to the left [44].

The visualization has one extra feature that can be adjusted interactively by the user. This is the λ value, which according to Sievert and Shirley is the relevance metric [44]. By this they mean that the user has the option to decide on the relevance of a word to a given topic. This metric is not directly related to LDA, this is calculations performed automatically by pyLDAvis using the LDA model. The idea is that the user have a way of not only showing the most frequent words but also decide on the exclusivity of a word to a given topic. Increasing the exclusivity allows the user to flexibly rank words for usefulness of interpreting topics [44]. The actual calculation of what happens behind the scenes when adjusting the λ value is beyond the scope of this thesis. However, their study shows that a λ value of 0.6 is optimal and gives the best results when humans interprets topics.

Chapter 3

Methods

The methodology used within this study consists of five major phases; data collection, preprocessing, topic modeling, visualisation and validation. The methods are based directly upon theory described in chapter two, but while chapter two consisted of four major sections, this chapter consists of five. The reason for this, is that on top of what the theory focused on, one have performed a manual reading of a small sub-section of the collected data to validate the results. Thus, it will give a better understanding of the type of data that was collected and analysed.

To get a brief overview of the stages that have been performed within this thesis, a graphical representation of the analysis pipeline is shown in Figure 3.1. The blue boxes show the main stages of an average topic model where one starts with data collection (section 3.1). Within this blue box there is a number of steps to go through. The specific steps in this pipeline are modified with respect to topic modeling and this thesis. Data collection in general, however, is essential for any analysis, as data is needed to work with. In this study an unique data set have been collected using an external web-harvesting company.

When performing data analysis, the next stage will for most, if not all, be preprocessing (section 3.2). Within this stage there exist numerous of different steps and these will differ largely depending on the analysis. The pipeline in Figure 3.1 shows the most important steps for this topic model. The stage of preprocessing, as explained in the beginning, is critical and should be paid large attention to.

After preprocessing the actual topic modeling is performed (section 3.3). The quality of the results within this stage is highly dependent upon correct preprocessing, which is why the user often tends to go back and update the steps of the preprocessing multiple times. This is indicated with a loop between *Preprocessing* and *Topic modeling*. For the purpose of this thesis, LDA was chosen as the preferred method.

The fourth blue box within this pipeline is visualisation (section 3.4). As explained earlier, topic models do not give topics that are easy to interpret, but they rather provide list of lists with words and their weights (see Figure 2.1 on page 11). Therefore, visualisation techniques should be used to make them easier to comprehend.

The last blue box in Figure 3.1 is validation (section 3.5). This stage is somewhat unique as it usually tends to be within the visualisation stage. However, as mentioned above, one has manually read a smaller sub-section of the documents to better understand the type of data that was collected and also what data the results are based upon.

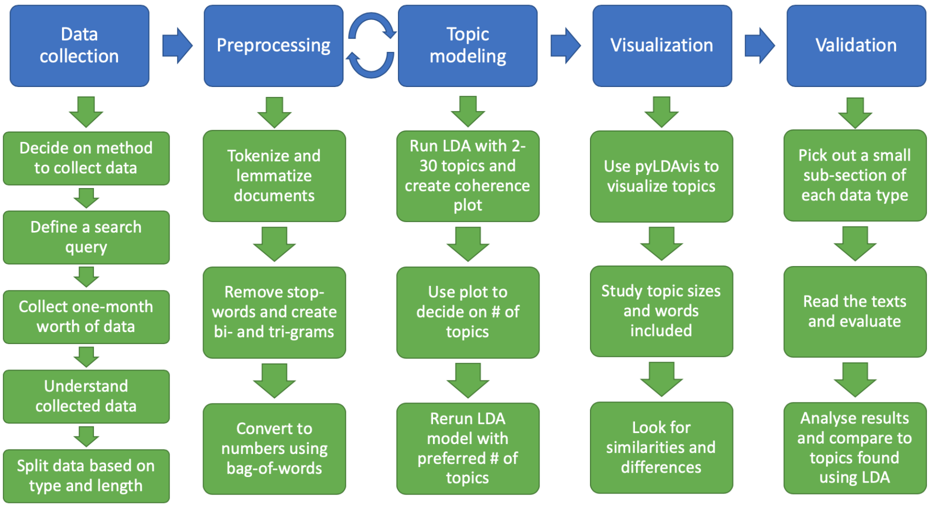

Data collection
Preprocessing
Topic modeling
Visualization
Validation
Decide on method to collect data
Define a search query
Collect one-month worth of data
Understand collected data
Split data based on type and length
Tokenize and lemmatize documents
Remove stop-words and create bi- and tri-grams
Convert to numbers using bag-of-words
Run LDA with 2-30 topics and create coherence plot
Use plot to decide on # of topics
Rerun LDA model with preferred # of topics
Use pyLDAvis to visualize topics
Study topic sizes and words included
Look for similarities and differences
Pick out a small sub-section of each data type
Read the texts and evaluate
Analyse results and compare to topics found using LDA

Figure 3.1: *Pipeline of the entire methodology from collected texts to validated topics. The blue boxes shows the main stages, while the green boxes shows the steps within each stage.*

Before diving deeper into the stages of the methodology used in this thesis, it is important to include that there exists a number of packages and tools built for the purpose of topic modeling in Python. These packages have built-in-functions for most of the steps explained in the theory chapter. Often there are several packages for the same stages of the pipeline. However, they tend to differ in speed and performance, and based on these factors and some testing one have decided that using Gensim and Spacy best suited. Both packages are known to be among the fastest and with the most functionality. The packages will be explained in greater detail within this chapter. The description will focus on how they have been applied and also what kind of modifications that have been made.

The methodology will take the reader through the actual process where some code snippets will be shown. These can be found in the appendix. This chapter will also explain how certain decisions were made, and how parameter settings were decided as well as make connections to the result chapter.

3.1 Data collection

The first stage of the process was to collect a large amount of text data. This was early in the process found to be very time-consuming and thereby taking much more time than first anticipated. From the theory chapter it is known that there are three well known ways of collecting data. Based upon this, with the research question in mind, an external web-harvesting company was decided to be used. This gave the broadest possibilities of text collection and also allowed for no information as to where the text data was gathered from. With an ocean of companies offering the ”same” service, there was a need to make a decision on who to trust. After some testing, reading multiple reviews and research [45], comparing prices as well as possibilities, Webhose.io was the chosen web-harvesting company. Their prices seemed fair, they offered free trial, and their setup within Python seemed to be reasonable. Reviewers also liked their possibilities and according to Webhose.io they already performed media monitoring, AI and machine learning, all with respect to text mining and NLP.

The web-harvesting company did not provide information on the data collection sources. This created some concern with respect to source criticism and lead to a situation where further sources for data collection could not be added. However, with the question of the thesis in mind, such source criticism was not as important and also allowed for less previous knowledge about the web sources.

The next step of the data collection process was to identify a search query before implementing the search into Python. The search query was based upon research within the field of vegetarian and vegan food as well as consultations with researchers within the field of study. In order to cover a larger set of documents about vegetarian and vegan food habits, as many different words as possible were included. Testing was also needed, as slight differences in the wording of the search query gave large variations in the number of documents available for collection. After several attempts the search query became; *food AND (vegetar* OR vegan* OR meatfree OR ”meat-free” OR ”meat free” OR plantbased OR ”plant-based” OR ”plant based” OR ”meat re*” OR flexitarian OR meatless OR ”meat less” OR ”meat-less” OR ”meat subst*”).*

As shown, the documents that were collected needed to include food and either of the vegetarian/vegan words. The reason several of the words were written as many as three times was to account for most of the incorrect spelling. The asterisks indicated that any letter combination after the star will be accepted. A screenshot of the actual Python implementation can be found within the appendix 6.1. The numbers shown in the code snippet are critical for proper functioning. The token is similar to an API key, and is uniquely created by Webhose.io for each of their customers; it indicates the allowances given to each customer. The ”ts” number is used to decide the number of days back in time that the the documents were included from. The search query also included a requirement that the texts that were collected needed to be written in English.

The collection process with Webhose.io gave more than 52 thousand documents of various lengths. The documents were collected over a time-span of 30 days. More precisely, as the search was performed on February 5th 2020, it included documents from January 7th 2020 through February 5th 2020. The time-span (ts-number) was set by the user as shown in Figure 6.1. Webhose.io collected a large number of different types of information where a major part was not needed for the purpose of this thesis. This is easier understood from the example result shown in Figure 4.3 in the result section.

Figure 4.3 also shows how the information is stored in a JSON format, which was an option to be set by the user when performing the search. As explained in the theory chapter, JSON is the preferred file format among most people due to its benefits over, for example, XML. As a result of the extra information collected there was a need to dive deeper into the different parts to check for other valuable documentation. However, most of it was URLs, information extraction performed on each document to identify social media ”likes” and ”shares”, dates and highlights, among others. As most of this data was not valuable for the purpose of this thesis, only a few sections of each document were kept and inserted into a pandas data-frame.

On top of the text itself, the ”sub-category” tag which was either; blogs, news or discussions were important to keep. Based upon the ”sub-category” tag, the texts were split into the distribution shown in Figure 4.1 within the result section.

Further, as there were large variation of lengths among the collected documents, they were split into three data sets within each sub-category. The lengths were decided based upon research of blogs, news and discussions. One have studied a handful of all three sub-categories and decided that a good split seemed to be texts up to one paragraph (short texts), texts longer than a paragraph and up to one page (medium texts), and texts longer than a page (long texts). The corpus was then split based upon these various text lengths; with short texts being less than 650 characters (symbols including white space), medium texts being longer than 650 characters but less than 4550 characters and long texts being longer than 4550 characters. These numbers were found from average character lengths per word and average number of words per page within the studied sub-categories. Given this, the final distribution of documents ended up looking like Figure 4.2 in the result section.

Out of the three length distributions, medium texts were decided to keep working with. First of all, literature indicate that topic models do not necessarily perform as well on too short or too long texts [27]. Secondly, the distribution among the three sub-categories were most similar to each other within the medium length data set. Lastly, manual reading of long texts seemed very time consuming and short texts often did not include enough content.

3.2 Preprocessing

This next stage of the pipeline is possibly the most important stage to increase the chances of valuable results. As described in the introduction, ”Garbage in, garbage out” [2], is a very good description of this. With poor preprocessing, the best model ever created can be used and still receive indecipherable results. Due to this, the stage of preprocessing often tends to be repeated multiple times with smaller and larger improvements every time. As shown in the pipeline in Figure 3.1 this stage included multiple green boxes, with equally important steps. The degree of repetition and updates, is indicated with an arrow making the stage of preprocessing an infinite loop with topic modeling.

In the beginning of this chapter it was introduced that there exist multiple similar packages which are created for somewhat the same purpose with respect to topic modeling. Further, describing how the two packages used within this study were Gensim and Spacy. As this was the first step where both packages were used, an introduction is needed. Gensim was introduced in 2008 as a collection of various Python scripts [46]. It was proposed by Radim Řehůřek, and is today a vary popular package among a large number of researchers within the field of text mining. The package include options for both preprocessing, analysis and visualisations. However, it will mostly be used with the preprocessing and analysis stages. Gensim is built to handle large amounts of data simultaneously, due to the behind the scenes calculations.

Spacy was created by the software company Explosion back in 2015 with the main developer being Matthew Honnibal and Ines Montani [47]. According to explosion.ai they both have several years of experience within the field of NLP. Spacy is today a free open-source library for advanced NLP in Python [47]. The benefit of the package being open-source, is that everyone can contribute to further improvements. The library is built to process and extract information from large volumes of texts [47]. The package does specifically well at preprocessing, which is what it is used for within this thesis.

Tokenization was the first step of preprocessing performed on the collected data after it was split based upon the above decisions. To perform the tokenization Gensim’s builtin function *gensim.utils.simple_preprocess* was used. The function uses a tokenization option to split all documents within the corpus to single tokens, where each token is a separate word. At the same time, punctuation is removed, all words are lower-cased and the token length is set to minimum two characters and maximum 15 characters. This allowed for removing one-character words as well as very long words.

Further, Spacy was used to lemmatize the data. Lemmatization was chosen to avoid under- and over-stemming, and also the option to exclude certain word types. Within the Spacy implementation, as shown in the appendix 6.2 on page vi, there was an option to set the allowed part-of-speech categories. As seen, only nouns, adjectives, verbs and adverbs were kept. Spacy performs the part-of-speech tagging using trained builtin languages based upon huge databases. Yet another benefit with Spacy is their large number of languages available. This allows for even Norwegian text mining with their packages.

The next step of the preprocessing was to remove stop-words. These words do not serve any good purpose for topic models. They only pollute the list of words (results) with non-relevant terms. A list of stop-words were created based upon Spacy's and Natural Language Toolkit's (NLTK) predefined stop-word lists. As neither of them seemed more preferred, one decided to keep them all. No stop-words where removed, due to lack of time. This could potentially have improved the result as explained in the theory section.

Now that the corpus was tokenized, lemmatized and stop-words were removed, bi- and tri-grams were created using Gensim's builtin *Phrases* function. As shown in Figure 6.3 on page vi in the appendix, the function has an option to set the minimum count. This means that a given word needs to have an overall frequency across the corpus which is higher than the value. Further, the user also has the option to set the threshold. This is a value that indicates the strictness of the model and is related to the probabilities as explained in the theory section. Based upon testing, and validation of some of the words, both min_count and threshold was set to 5. This seemed to include a great number of bi- and tri-grams, but at the same time, not convert every single word into either of the two.

Further the list of words were converted into numbers to prepare for the topic modeling. This step is also known as the creation of the vocabulary. LDA needed bag-of-words to work properly. As a result, one converted the preprocessed documents into numbers as shown in appendix 6.4. As shown there, Gensim was used to perform the conversion as it creates the optimal bag-of-words to be used with LDA within the next stage.

First, documents were converted with the *corpra.Dictionary* function, to create a vocabulary. The vocabulary was further reduced with $filter\_extremes$. This is an alternative to tf-idf as both are meant to reduce the vocabulary and by that hopefully receive improved results. $filter\_extremes$ was set to remove all words occurring in less than 3 documents and all words occurring in more than 70 percent of the corpus. These limits were based upon multiple attempts of trial and error. The size of the vocabulary and the amount of words that were reduced were identified. The last step of the conversion was to convert the words, included in the final vocabulary, into bag-of-words format. This was done with Gensim's doc2bow.

3.3 Topic modeling

The next stage of the pipeline shown in Figure 3.1, is the topic modeling. This was where the "magic" happened and topics were created based on the training. The stage of topic modeling includes the step of testing different topic values with LDA and measure the associated coherence scores, as well as training the model with the preferred topic number in the end.

As stated and described throughout the thesis, LDA was used to create the topics, as it is known to be the preferred method among researchers [3]. As such, an LDA model needed to be implemented. Figure 6.5 within the appendix, shows how this was done within Python. As shown in the figure, Gensim's LdaMulticore was used as it allows for faster training, due to the possibilities of multiprocessing. Within the model, the user needed to include a corpus, id2word, num_topics, $random\_state$, chunksize, passes and workers. The corpus is the final bag-of-words vectors, while id2word is the vocabulary. num_topics is the user defined number of topics, $random\_state$ is the seed used for the random number generator, while the chunksize is the number of documents used within each training iteration. Passes are the number of times the entire corpus is covered within the training session, while workers are the number of cores to be used.

As explained above, the corpus, also known as the bag-of-words vectors are based upon the vocabulary (id2word). This means that reducing or expanding the vocabulary was affecting this step directly. The the optimal topic number was based upon the coherence graph 4.4, as shown in the result chapter. Coherence was chosen as the validation measure over word intrusion, as word intrusion was found to be tedious and expert knowledge were needed.

The coherence scores were found with Gensim's CoherenceModel as shown in Figure 6.6. Within the model a set of topic values between 2 and 30 were tested. Higher numbers were also tested, but these gave lower coherence values.

It is worth mentioning that both the η and the α values, which are Dirichlet parameters, could be set manually. This is only needed when there exist previous knowledge about the word and/or topic distributions respectively.

Analysing the coherence graphs, the optimal number of topics were decided per sub-category. The numbers were picked based upon the first drop within each of the graphs. Using these topic numbers and the same LDA multicore model, as shown in the code snippet 6.5, LDA was retrained to output the optimal topics. The topics were further visualised within the next stage.

3.4 Visualisation

The last step of most topic models are to present and analyze the results, which in this case are the topics. This was also the case within this thesis, however, as mentioned earlier a manual reading was also performed to make a better understanding of the results as well as a deeper intuition of the type of data that was collected.

pyLDAvis was used for visualisation purposes. Instead of presenting the topics as a list of lists, similar to Figure 2.1, pyLDAvis created interactive visualisations similar to Figure 2.8. The visualisation tool has an option to set a λ value, which is a relevance metric. When interpreting the topics, this value was set 0.6 based upon Sievert and Shirley's study [44].

3.5 Validation

The very last part of this thesis was a manual reading of a smaller subset of the corpus. This is usually not performed within other topic modeling studies, but it was done to make a better understanding of the type of data that were analysed. The manual reading was performed by a group of employees at Nofima, where each person was given a section of the subset. With the manual reading, they were also given a questionnaire which included the questions shown in Table 3.1. As seen, most questions were given as multiple choice to reduce the span of answers. The left column is the table headings within the excel sheet of the questionnaire, while the right column is an explanation of each question.

The number of texts manually read, were picked based upon the optimal topic number found within the previous stage. 10 documents per topic where chosen based upon the document-to-topic probability. The 10 documents with the highest probability to each of the topics were extracted and manually read. While reading, the questionnaire were answered. A total of 100 documents from the discussion sub-category, and 120 from each of the blogs and news sub-categories were read.

Table 3.1: *Questionnaire manual reading. These are the questions that were given with the manual reading. Most of the questions where multiple choice where the answer options are given within parenthesis. The left column is the heading of the excel document, while the right column is the actual questions.*

text number	Fill in the number found in the far left column of the text document
text id	Fill in the number found in the text id column (second from left) of the text document
Medium	At the top of the dataset it says if the medium is news, discussions or blogs (News/blogs/discussion)
Readers name	Fill in your own name
Relevant	Do you find the text relevant for meat reduction (or talking about meat in any way)?
Relevant	(Yes/No) (If the answer is no there is no need to answer the rest of the questions for that text)
Reduction of meat	Do you find the text positive, neutral or negative towards reduction of meat? (Positive/Neutral/Negative)
Objective or subjective	Do you find the text objective (fact based) or subjective (personal opinion based)? (Objective/Subjective)
Need for meat reduction	Do you find that the text focuses on a need to reduce meat consumption? (Yes/No/I don't know)
Emotional words	Does the writer use emotional words to get the message out? (Yes/No/I don't know)
Animal welfare	Does the writer focuses on animal welfare as reasons to reduce or increase meat consumption? (Yes/No/I don't know)
Climate change	Does the writer focuses on climate change as reasons to reduce or increase meat consumption? (Yes/No/I don't know)
Nutrition	Does the writer focuses on nutrition as reasons to reduce or increase meat consumption? (Yes/No/I don't know)
Keywords	Please write at least 5, top 10 keywords that explains the main message of the text

The reading also allowed the user to get an understanding of the average quality of documents as well as discovery of major concerns. These concerns will be shown in the result chapter 4.4 and discussed within the discussion 5.5.

Chapter 4

Results

The results chapter will cover some basic information about the number and type of texts that were collected. Further, it will present the findings of the optimal topic numbers as well as visualisations of some of the topics. The visualisations were made using pyLDAvis, which created interactive HTML files. As interactive files cannot be included within a PDF, the results will be presented as screenshots instead. They can also be found by downloading the HTML files from here. To preview and interact with the visualisations, the files need to be downloaded locally and then opened in a browser. Lastly, the result chapter will present some of the findings that came from the manual reading.

4.1 Data collection results

The use of Webhose.io provided a set of about 52 thousand texts. These were split into three sub-categories; blogs, news and discussions. The categorization was taken from Webhose.io's label where their decision seemed to be made upon predefined allocations of the texts. This means that if a text was collected from a news site, it was categorized as news, even if the text was a blog post within the news site. Figure 4.1 shows the distribution of each type of medium that was collected.

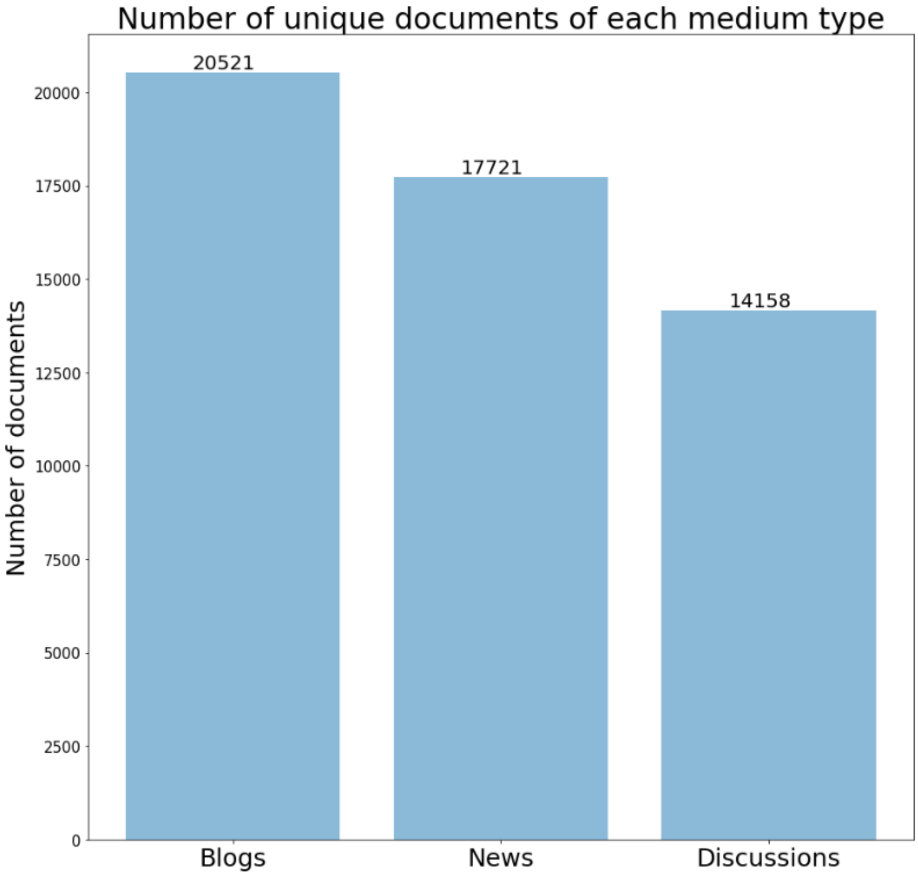


Figure 4.1: *Number of documents within each category. The following plot shows the distribution of the three different medium sub-categories that were collected.*

Each type of medium were further split into new sub categories based on the length differences. The new categories are shown in Figure 4.2. Information about the split and the length differences can be found in the method section 3.1. The figure shows how most of the discussion documents were found to be of short and medium lengths while news and blogs were more evenly distributed among medium and long document lengths.

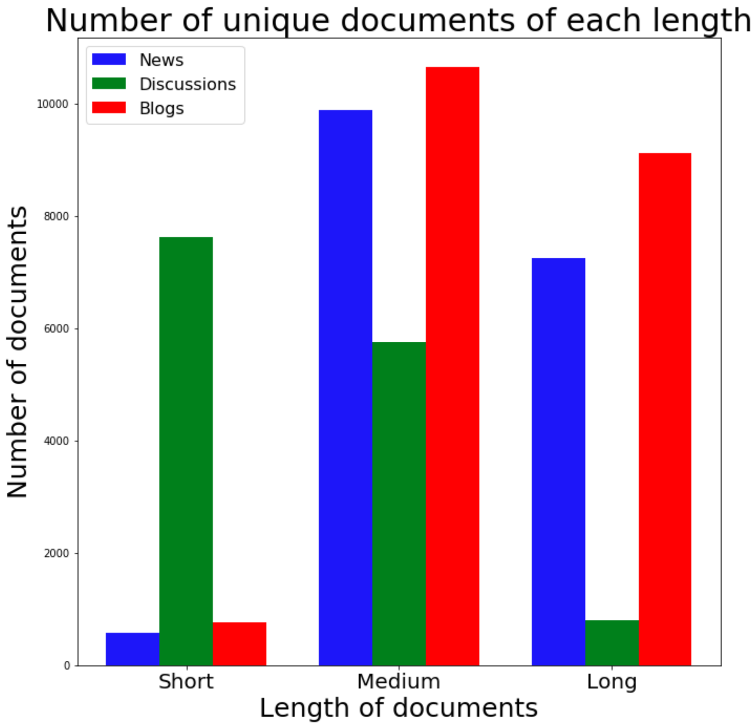
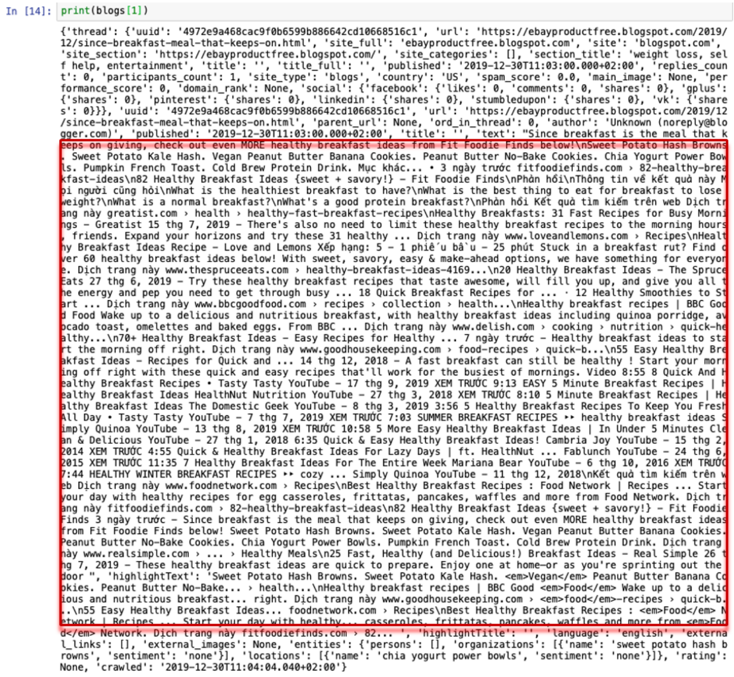


Figure 4.2: *Number of documents within each category after length split. The following plot shows the distribution of the three different medium sub-categories that were collected after they were split based upon length.*

Each document that was collected included lots of information that was not needed for the purpose of this thesis. Figure 4.3 shows an example of a text that was collected, where the red box indicates the information used within this study. The content of the text, is very small and not supposed to be readable, it is rather used as an illustration.



```
In [14]: print(blogs[1])
```

```
{'thread': {'uuid': '4972e9a468cac9f0b6599b886642cd10668516c1', 'url': 'https://ebayproductfree.blogspot.com/2019/
12/since-breakfast-meal-that-keeps-on.html', 'site_full': 'ebayproductfree.blogspot.com', 'site': 'blogspot.com',
'site_section': 'https://ebayproductfree.blogspot.com/', 'site_categories': [], 'section_title': 'weight loss, sel
f help, entertainment', 'title': '', 'title_full': '', 'published': '2019-12-30T11:03:00.000+02:00', 'replies_coun
t': 0, 'participants_count': 1, 'site_type': 'blogs', 'country': 'US', 'spam_score': 0.0, 'main_image': None, 'per
formance_score': 0, 'domain_rank': None, 'social': {'facebook': {'likes': 0, 'comments': 0, 'shares': 0}, 'gplus':
{'shares': 0}, 'pinterest': {'shares': 0}, 'linkedin': {'shares': 0}, 'stumbledupon': {'shares': 0}, 'vk': {'share
s': 0}}}, 'uuid': '4972e9a468cac9f0b6599b886642cd10668516c1', 'url': 'https://ebayproductfree.blogspot.com/2019/12
/since-breakfast-meal-that-keeps-on.html', 'parent_url': None, 'ord_in_thread': 0, 'author': 'Unknown (noreply@blo
gger.com)', 'published': '2019-12-30T11:03:00.000+02:00', 'title': '', 'text': "Since breakfast is the meal that k
eeps on giving, check out even MORE healthy breakfast ideas from Fit Foodie Finds below!\nSweet Potato Hash Browns
. Sweet Potato Kale Hash. Vegan Peanut Butter Banana Cookies. Peanut Butter No-Bake Cookies. Chia Yogurt Power Bow
ls. Pumpkin French Toast. Cold Brew Protein Drink. Mục khác... • 3 ngày trước fitfoodiefinds.com › 82-healthy-brea
kfast-ideas\n82 Healthy Breakfast Ideas {sweet + savory!} – Fit Foodie Finds\nPhản hồi\nThông tin về kết quả này M
oi người cũng hỏi\nWhat is the healthiest breakfast to have?\nWhat is the best thing to eat for breakfast to lose
weight?\nWhat is a normal breakfast?\nWhat's a good protein breakfast?\nPhản hồi Kết quả tìm kiếm trên web Dịch tr
ang này greatist.com › health › healthy-fast-breakfast-recipes\nHealthy Breakfasts: 31 Fast Recipes for Busy Morni
ngs – Greatist 15 thg 7, 2019 – There's also no need to limit these healthy breakfast recipes to the morning hours
, friends. Expand your horizons and try these 31 healthy ... Dịch trang này www.loveandlemons.com › Recipes\nHealt
hy Breakfast Ideas Recipe – Love and Lemons Xếp hạng: 5 – 1 phiếu bầu – 25 phút Stuck in a breakfast rut? Find o
ver 60 healthy breakfast ideas below! With sweet, savory, easy & make-ahead options, we have something for everyon
e. Dịch trang này www.thespruceeats.com › healthy-breakfast-ideas-4169...\n20 Healthy Breakfast Ideas – The Spruce
Eats 27 thg 6, 2019 – Try these healthy breakfast recipes that taste awesome, will fill you up, and give you all t
he energy and pep you need to get through busy ... 18 Quick Breakfast Recipes for ... · 12 Healthy Smoothies to St
art ... Dịch trang này www.bbcgoodfood.com › recipes › collection › health...\nHealthy breakfast recipes | BBC Goo
d Food Wake up to a delicious and nutritious breakfast, with healthy breakfast ideas including quinoa porridge, av
ocado toast, omelettes and baked eggs. From BBC ... Dịch trang này www.delish.com › cooking › nutrition › quick-he
althy...\n70+ Healthy Breakfast Ideas – Easy Recipes for Healthy ... 7 ngày trước – Healthy breakfast ideas to sta
rt the morning off right. Dịch trang này www.goodhousekeeping.com › food-recipes › quick-b...\n55 Easy Healthy Bre
akfast Ideas – Recipes for Quick and ... 14 thg 12, 2018 – A fast breakfast can still be healthy ! Start your morn
ing off right with these quick and easy recipes that'll work for the busiest of mornings. Video 8:55 8 Quick And H
ealthy Breakfast Recipes • Tasty Tasty YouTube – 17 thg 9, 2019 XEM TRƯỚC 9:13 EASY 5 Minute Breakfast Recipes | H
ealthy Breakfast Ideas HealthNut Nutrition YouTube – 27 thg 3, 2018 XEM TRƯỚC 8:10 5 Minute Breakfast Recipes | He
althy Breakfast Ideas The Domestic Geek YouTube – 8 thg 3, 2019 3:56 5 Healthy Breakfast Recipes To Keep You Fresh
All Day • Tasty Tasty YouTube – 7 thg 7, 2019 XEM TRƯỚC 7:03 SUMMER BREAKFAST RECIPES ▸▸ healthy breakfast ideas S
imply Quinoa YouTube – 13 thg 8, 2019 XEM TRƯỚC 10:58 5 More Easy Healthy Breakfast Ideas | In Under 5 Minutes Cle
an & Delicious YouTube – 27 thg 1, 2018 6:35 Quick & Easy Healthy Breakfast Ideas! Cambria Joy YouTube – 15 thg 2,
2014 XEM TRƯỚC 4:55 Quick & Healthy Breakfast Ideas For Lazy Days | ft. HealthNut ... Fablunch YouTube – 24 thg 6,
2015 XEM TRƯỚC 11:35 7 Healthy Breakfast Ideas For The Entire Week Mariana Bear YouTube – 6 thg 10, 2016 XEM TRƯỚC
7:44 HEALTHY WINTER BREAKFAST RECIPES ▸▸ cozy ... Simply Quinoa YouTube – 11 thg 12, 2018\nKết quả tìm kiếm trên w
eb Dịch trang này www.foodnetwork.com › Recipes\nBest Healthy Breakfast Recipes : Food Network | Recipes ... Start
your day with healthy recipes for egg casseroles, frittatas, pancakes, waffles and more from Food Network. Dịch tr
ang này fitfoodiefinds.com › 82-healthy-breakfast-ideas\n82 Healthy Breakfast Ideas {sweet + savory!} – Fit Foodie
Finds 3 ngày trước – Since breakfast is the meal that keeps on giving, check out even MORE healthy breakfast ideas
from Fit Foodie Finds below! Sweet Potato Hash Browns. Sweet Potato Kale Hash. Vegan Peanut Butter Banana Cookies.
Peanut Butter No-Bake Cookies. Chia Yogurt Power Bowls. Pumpkin French Toast. Cold Brew Protein Drink. Dịch trang
này www.realsimple.com › ... › Healthy Meals\n25 Fast, Healthy (and Delicious!) Breakfast Ideas – Real Simple 26 t
hg 7, 2019 – These healthy breakfast ideas are quick to prepare. Enjoy one at home—or as you're sprinting out the
door ", 'highlightText': 'Sweet Potato Hash Browns. Sweet Potato Kale Hash. <em>Vegan</em> Peanut Butter Banana Co
okies. Peanut Butter No-Bake... › health...\nHealthy breakfast recipes | BBC Good <em>Food</em> Wake up to a delic
ious and nutritious breakfast... right. Dịch trang này www.goodhousekeeping.com › <em>food</em>-recipes › quick-b.
..\n55 Easy Healthy Breakfast Ideas... foodnetwork.com › Recipes\nBest Healthy Breakfast Recipes : <em>Food</em> N
etwork | Recipes ... Start your day with healthy... casseroles, frittatas, pancakes, waffles and more from <em>Foo
d</em> Network. Dịch trang này fitfoodiefinds.com › 82... ', 'highlightTitle': '', 'language': 'english', 'externa
l_links': [], 'external_images': None, 'entities': {'persons': [], 'organizations': [{'name': 'sweet potato hash b
rowns', 'sentiment': 'none'}], 'locations': [{'name': 'chia yogurt power bowls', 'sentiment': 'none'}]}, 'rating':
None, 'crawled': '2019-12-30T11:04:04.040+02:00'}
```

Figure 4.3: *Example of collected text document. The figure shows an example of a document collected with Webhose.io. It shows the amount of unnecessary information that was included, where the red box shows the information that was used. The text on the picture is not supposed to be readable, it is just used as an illustration to identify amount of unnecessary information.*

4.2 Topic modeling

Within the section of topic modeling, the most important findings were the optimal number of topics and the actual topics themselves within each of the three sub-categories. To find the optimal number of topics the preprocessed data was trained with number of topics ranging from 2 to 30 with intervals of 2. This resulted in three coherence graphs, which further were used to identify the optimal number of topics per sub-category. As shown in Figure 4.4, the optimal number of topics is identified for each of the three data sets with black circles. These numbers are chosen as they are at the highest probability before it starts decreasing slightly. The training was also performed on topic numbers ranging from 10 to 210, with intervals of 20, but as the plot did not show any evidence of higher topic numbers to be correct or more valuable, the results were discarded.

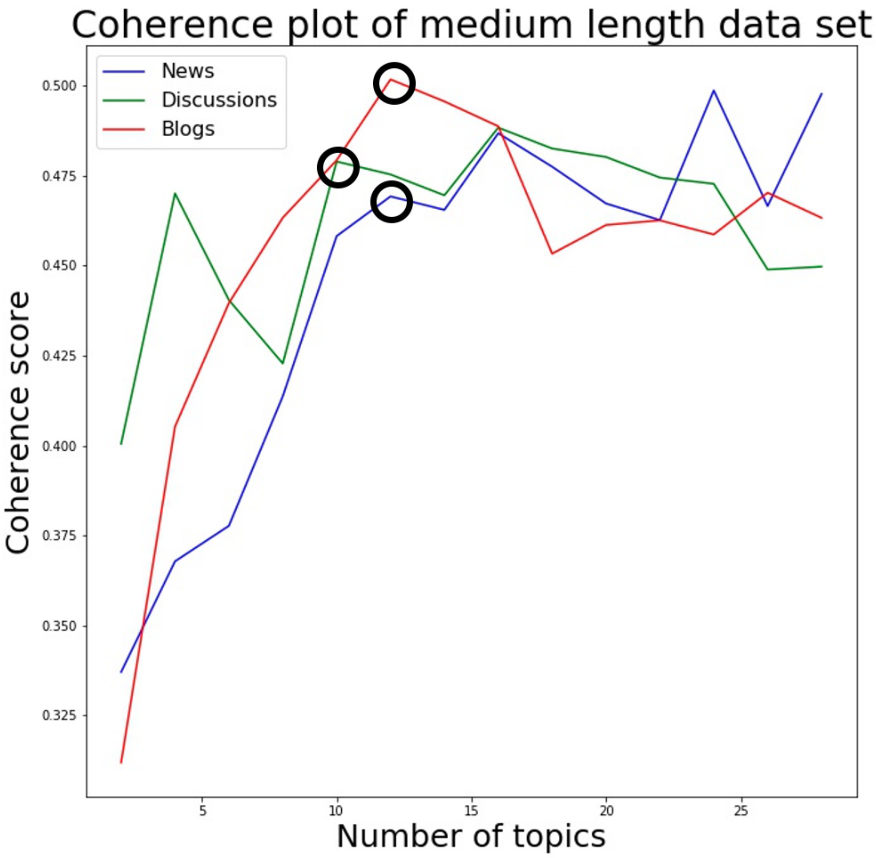


Figure 4.4: *Coherence plot of the coherence graphs for each of the subcategories. As shown, 12 topics were found to be the optimal number for both news and blogs, while discussions was found to have 10 as it's optimal number of topics. These values are shown as black circles.*

4.3 Visualisation

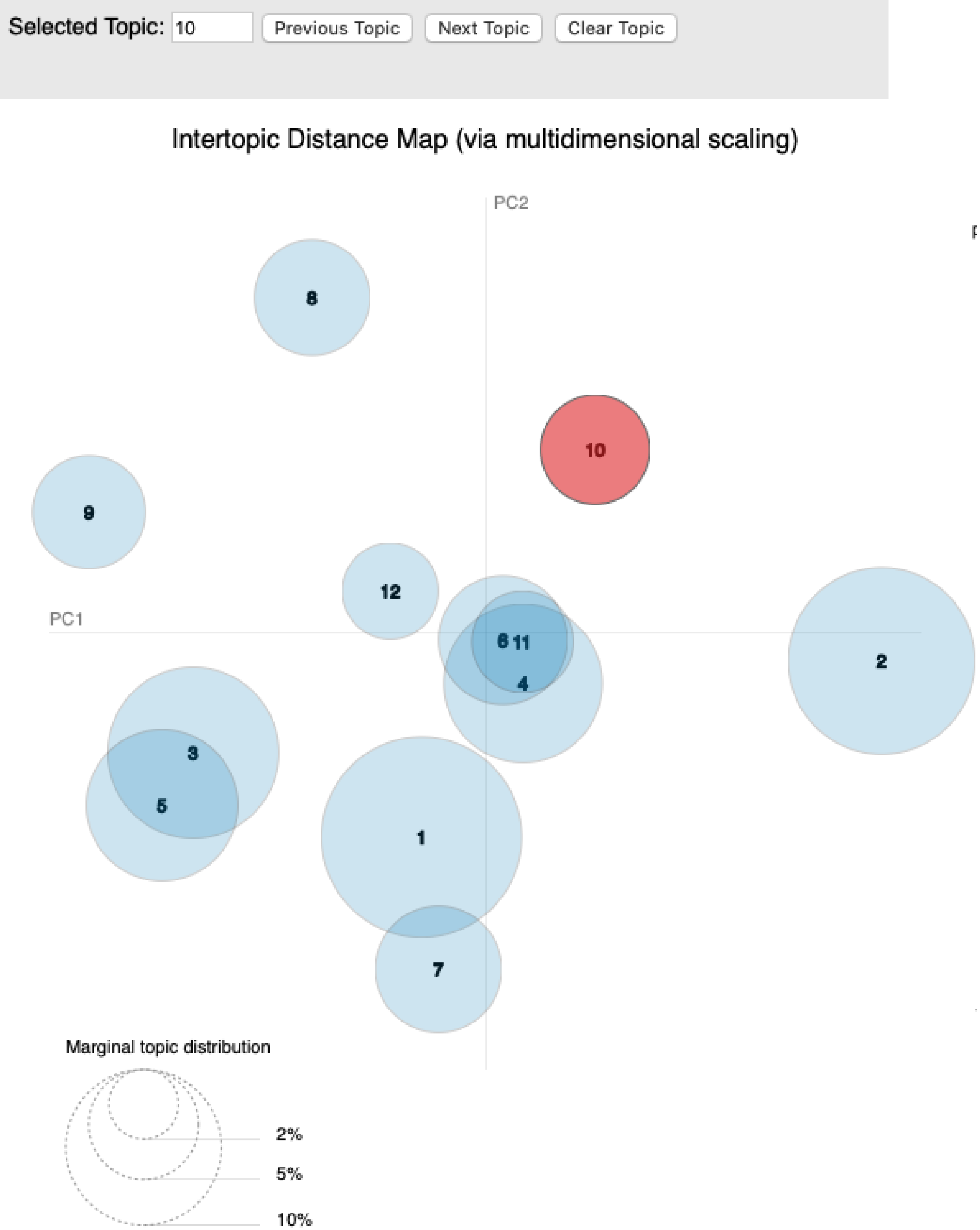
The visualisations showed in Figures 4.5, 4.6 and 4.7 are screenshots of interactive models created with pyLDAvis. As shown, the figures are similar to the example shown in Figure 2.8, but split in two for readability improvements. Figures 4.5 and 4.6 show a topic found within the blog data set considered to be relevant as it included words such as *plant-base, eat, diet, recipe, healthy, meat, dish* and *vegetarian*. Figure 4.7 (also found within the blog data set), on the other hand, do not show as much evidence of being related to vegetarian and vegan food as only a few words are related. The left side of this figure is identical to Figure 4.5, however, topic 2 is colored in red instead.

The remaining topics will be presented within the Tables 4.1 - 4.6. The results are presented this way, as the screenshots take too many pages. The tables will only include words relevant to vegetarian and vegan food, while the remaining words are discarded within the result section. However, such relevancy is dependent upon the person who analyses the topics. The remaining topics can also be visualised from here.

The left panel shows the overall topic distribution, while the right panels (Figures 4.6 and 4.7) show the word distribution within the selected topic [44]. As explained within the theory chapter 2.4, the topics were projected onto two dimensions using PCA. The distances between the clusters indicate the similarity among the topics. It should therefore be possible to detect larger differences when clusters are further apart. However, detecting these differences need expert knowledge within the field of vegetarian and vegan food. The sizes of the clusters are dependent upon the number of documents explained by the given topic. In other words, a larger cluster means that the topic is more important in explaining the themes within the corpus. All topics were interpreted with a λ value of 0.6 based upon Sievert and Shirley's study [44].

4.3.1 Topic results presented with pyLDAvis

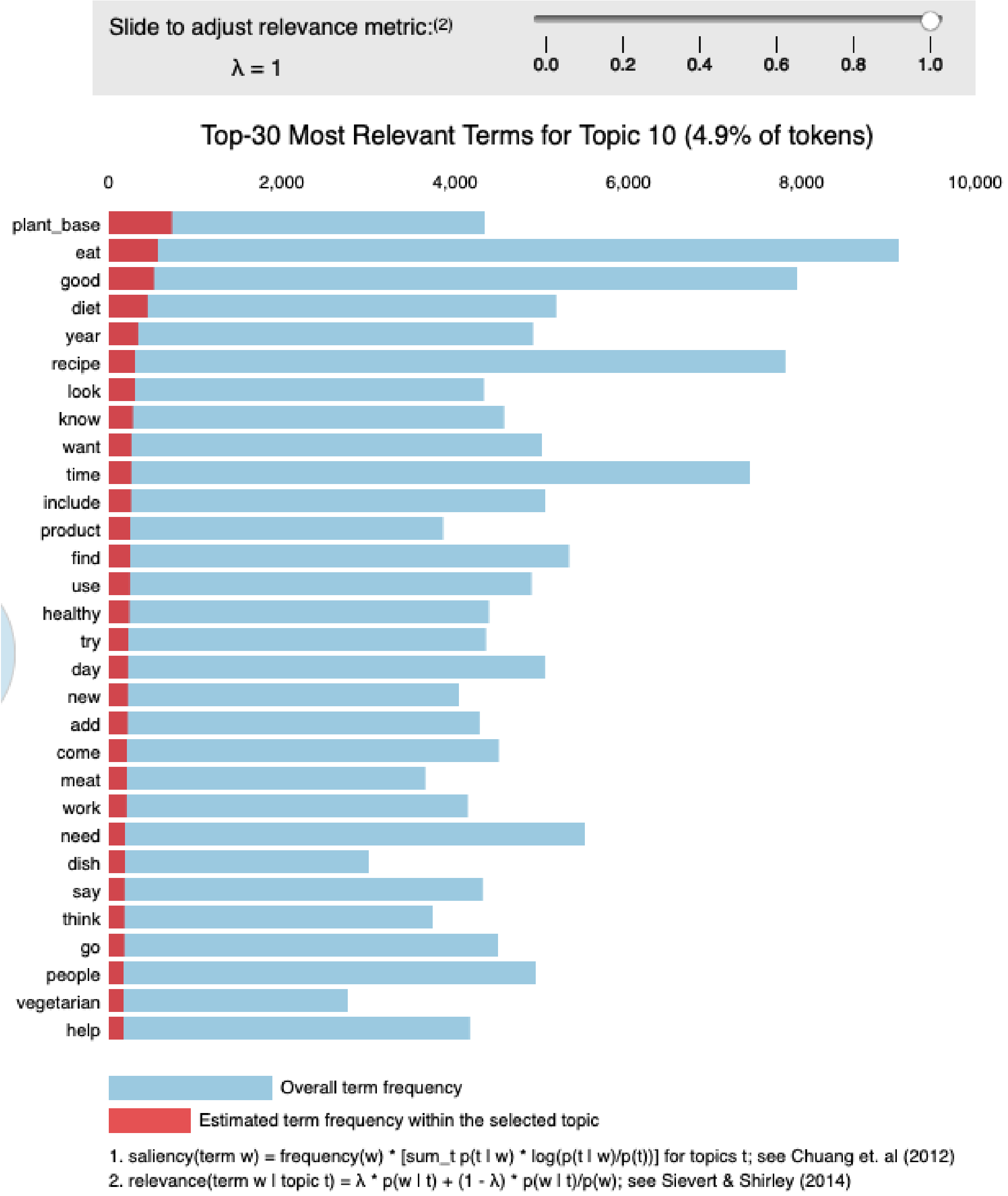
Based upon the visualisations found using pyLDAvis, it is important to mention that the right panel, which shows the words, shows both the overall word frequency and the word frequency within a selected topic. In other words, the red part shows the word frequency of a given word within the selected topic. This indicates how important a given word is when interpreting the topic. The overall word frequency, shown in blue, indicates the importance of the same word to the overall corpus. The sizes of the clusters indicate the importance of each topic with respect to the number of documents explained per topic. Further, the PC1 and PC2 indicate the orthogonal largest variance and is used to explain the most information possible within a 2D plot.



Selected Topic: 10

Previous Topic
Next Topic
Clear Topic
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
8
10
9
12
6 11
4
2
3
5
1
7
Marginal topic distribution
2%
5%
10%

Figure 4.5: *pyLDAvis visualisation of a topic from the* ***blogs*** *data set. The following figure is a screenshot of topic 10 found within the blogs data set. This topic was found to include the most words related to vegetarian and vegan food among the topics created from the* ***blogs*** *data set. The figure shows the left side of the pyLDAvis visualization, which is the overview. The exact same figure is created for the bad topic shown below, however, topic 2 is colored red instead.*



Slide to adjust relevance metric:(2)

λ = 1

Top-30 Most Relevant Terms for Topic 10 (4.9% of tokens)

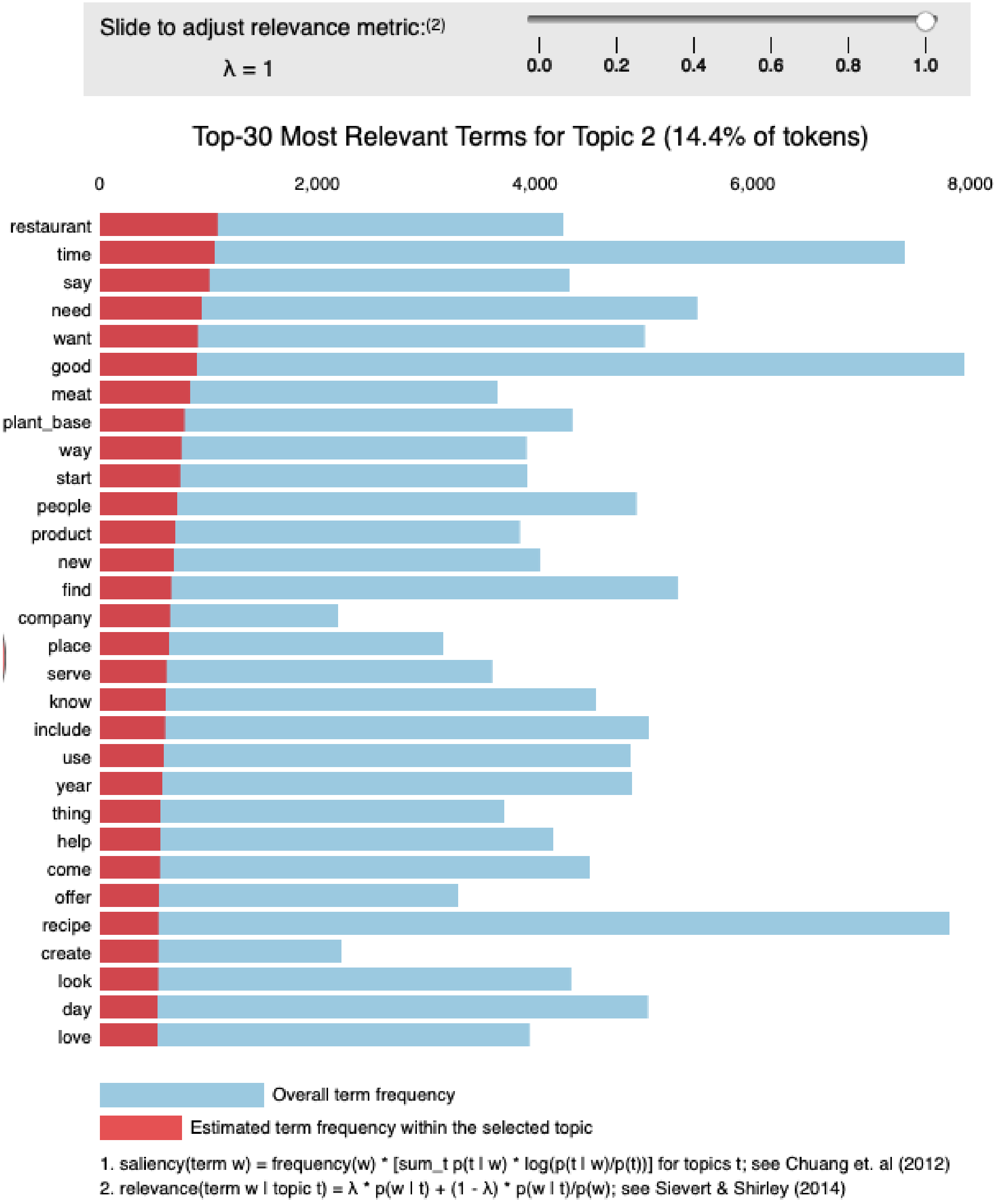
Overall term frequency

Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)

2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 4.6: *pyLDAvis visualisation of a topic from the* ***blogs*** *data set. This figure is a screenshot of topic 10 found within the blogs data set. This topic was found to include the most words related to vegetarian and vegan food among the topics created from the* ***blogs*** *data set. The figure shows the right side of the visualisation, which is the word distribution within the selected topic.*



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 4.7: *pyLDAvis visualisation of a less relevant* ***blogs*** *topic. The following figure is a screenshot of topic 2 found within the* ***blogs*** *data set. This topic was found to include very few words related to the vegetarian and vegan food. The figure shows the right side of the visualisation, which is the word distribution within the selected topic.*

4.3.2 Topic results presented within tables

The overall interpretation of Tables 4.1 and 4.2, which are based upon the blogs data set, indicates that the main focuses are on healthy diets and recipes. Discussions, shown in Tables 4.3 and 4.4, on the other hand, focuses on animals and vegetarian/vegan topics. Lastly, news, found within Tables 4.5 and 4.6, focuses on the launching of plant based menus in restaurants.

Table 4.1: *Keywords from blogs data set topic 1-6. These are the vegetarian and vegan related words from the visualisations of the* ***blogs*** *data set made with pyLDAvis. The entire list of words can be found from the interactive HTML files found here.*

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**	**Topic 6**
Eat	Restaurant	Recipe	Restaurant	Eat	Eat
Recipe	Meat	Eat	Meal	Recipe	Diet
Healthy	Plant base	Diet	Eat	Meal	Meat
Diet	New	Meal	Recipe	Dish	Healthy
Meat		Protein	Cook	Cook	Recipe
Plant base		Cook	Ingredient	Oil	New
Meal		Plant base			Dish
		Healthy			

Table 4.2: *Keywords from blogs data set topic 7-12. These are the vegetarian and vegan related words from the visualisations of the* ***blogs*** *data set made with pyLDAvis. The entire list of words can be found from the interactive HTML files found here.*

Topic 7	**Topic 8**	**Topic 9**	**Topic 10**	**Topic 11**	**Topic 12**
Eat	Diet	Recipe	Plant base	Recipe	Eat
Meal	Eat	Healthy	Eat	Eat	Recipe
Healthy	Recipe	Meal	Diet	Healthy	Vegetarian
Diet	Vegetarian	Cook	Recipe	Change	Diet
Ingredient	Restaurant	Eat	Healthy	Dish	Taste
	Water	Diet	Meat	Burger	
	Healthy	Flavor	Dish	Meat	
		Ingredient	Vegetarian	Restaurant	
			Flavor	Body	

Table 4.3: *Keywords from discussion data set topic 1-5. These are the vegetarian and vegan related words from the visualisations of the* ***discussion*** *data set made with pyLDAvis. The entire list of words can be found from the interactive HTML files found here.*

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**
Diet	Eat	Meat	Eat	Eat
Eat	Meat	Eat	Animal	Protein
Vegetarian	Vegan	Vegetarian	Healthy	Diet
Meat	Animal	Meal	Diet	Healthy
Animal	Plant base	Restaurant	Meat	Vegetarian
Vegan	Eat meat	Diet	Cook	Meat
Water	Healthy	Cook	Fat	
Recipe	Meal	Animal	Body	
	Diet	Taste		
		Dish		

Table 4.4: *Keywords from discussion data set topic 6-10. These are the vegetarian and vegan related words from the visualisations of the* ***discussion*** *data set made with pyLDAvis. The entire list of words can be found from the interactive HTML files found here.*

Topic 6	**Topic 7**	**Topic 8**	**Topic 9**	**Topic 10**
Eat	Eat	Eat	Meat	Eat
Meat	Change	Cheese	Plant base	Diet
Diet	Animal	Diet	Protein	plant
Vegetarian	Meat	Easy	Diet	Healthy
Meal	Vegetarian	Start	Source	Meat
Plant base	Cook		Vegetarian	Vegetarian
Live	Meal		Vegan	Animal
Restaurant			Animal	Restaurant

Table 4.5: *Keywords from news data set topic 1-6. These are the vegetarian and vegan related words from the visualisations of the* ***news*** *data set made with pyLDAvis. The entire list of words can be found from the interactive HTML files found here.*

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**	**Topic 6**
Food	Food	Restaurant	Launch	Food	Food
Eat	Plant base	Food	Food	Plant base	Product
Plant base	Meat	Menu	Plant base	Recipe	New
Meat	Eat	Eat	Customer	Meat	Include
Diet	Meal	Include	Industry	Diet	Plant base
Ingredient	Diet	New	Eat	Ingredient	Consumer
	Healthy	Dish	Dish	Meal	Protein
	Recipe	Meat	Business	Eat	Restaurant
	Restaurant	Plant base		Cook	Burger
	Serve	Ingredient		Healthy	Customer
				Dish	Drink
					Eat

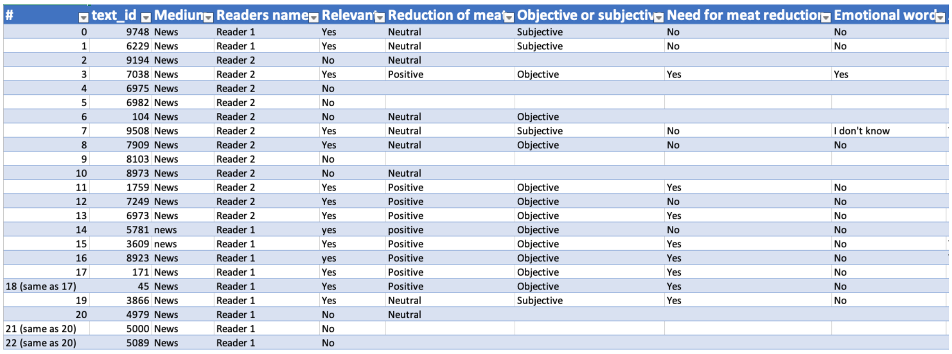
Table 4.6: *Keywords from news data set topic 7-12. These are the vegetarian and vegan related words from the visualisations of the* ***news*** *data set made with pyLDAvis. The entire list of words can be found from the interactive HTML files found here.*

Topic 7	**Topic 8**	**Topic 9**	**Topic 10**	**Topic 11**	**Topic 12**
Food	Food	Food	Eat	Plant base	Food
Restaurant	Product	Restaurant	Food	Food	Restaurant
Consumer	Meat	Plant base	Plant base	Milk	Diet
Product	Plant base	Meat	Meat	Restaurant	Product
Taste	Restaurant	Recipe	Vegetarian	Oat milk	Eat
Food eat way	Protein	Healthy	Ingredient	Flavor	Ingredient
Plant base	Consumer		Change	Meat	
Serve	Eat		Launch	Meal	
Launch	Trend			Healthy	

4.4 Validation

The validation was performed to establish a general knowledge of the colored text data as well as a comparison to the actual topics found with the model. Table 4.7 shows a screenshot of a part of the questionnaire which was answered with the manual reading. The figure shows how the readers needed to fill out certain information about each document.

Table 4.7: *Manual reading illustration. The table is used to illustrate how the readers were questioned about each of the texts they read.*



#	text_id	Mediun	Readers name	Relevant	Reduction of meat	Objective or subjectiv	Need for meat reductior	Emotional word
0	9748	News	Reader 1	Yes	Neutral	Subjective	No	No
1	6229	News	Reader 1	Yes	Neutral	Subjective	No	No
2	9194	News	Reader 2	No	Neutral			
3	7038	News	Reader 2	Yes	Positive	Objective	Yes	Yes
4	6975	News	Reader 2	No				
5	6982	News	Reader 2	No				
6	104	News	Reader 2	No	Neutral	Objective		
7	9508	News	Reader 2	Yes	Neutral	Subjective	No	I don't know
8	7909	News	Reader 2	Yes	Neutral	Objective	No	No
9	8103	News	Reader 2	No				
10	8973	News	Reader 2	No	Neutral			
11	1759	News	Reader 2	Yes	Positive	Objective	Yes	No
12	7249	News	Reader 2	Yes	Positive	Objective	No	No
13	6973	News	Reader 2	Yes	Positive	Objective	Yes	No
14	5781	news	Reader 1	yes	positive	Objective	No	No
15	3609	news	Reader 1	Yes	Positive	Objective	Yes	No
16	8923	News	Reader 1	yes	Positive	Objective	Yes	No
17	171	News	Reader 1	Yes	Positive	Objective	Yes	No
18 (same as 17)	45	News	Reader 1	Yes	Positive	Objective	Yes	No
19	3866	News	Reader 1	Yes	Neutral	Subjective	Yes	No
20	4979	News	Reader 1	No	Neutral			
21 (same as 20)	5000	News	Reader 1	No				
22 (same as 20)	5089	News	Reader 1	No				

The manual reading of 100 discussions, 120 blogs and 120 news documents identified several important findings. First of all, it identified that 40 % of the news, 53 % of the discussions, and 72 % of the blogs were found to be non-relevant texts with regard to vegetarian and vegan food. Secondly, it identified that 17 % of the news, 6 % of the discussions and 1 % of the blogs were identical texts. In other words, these texts occurred two or more times.

Thirdly, the findings showed that news mostly had neutral opinionated texts with regard to meat reduction, discussions mostly had positive opinionated texts, while blogs were evenly distributed among neutral and positive opinionated texts. The findings also identified that news texts were reasoning their vegetarian and vegan based documents mostly around nutrition and climate change. On the other hand, discussions focused on nutrition while blogs focused on nutrition and animal welfare.

Lastly, the manual reading gave numerous keywords related to vegetarian and vegan food. Some examples were ”fake meat”, ”vegetarian sandwich”, ”beyond meat”, ”killing animals” and ”anti-meat tribe”, among others.

Chapter 5

Discussion

The main purpose of this study has been to analyse how text mining, more specifically topic modeling, can be used to explore and identify the main topics among a large amount of unknown text data. A sub-purpose of the thesis has been to find and test alternative methods of gathering and analysing information within various fields. Examples would be sustainable production, politics and health issues, among others, where the purpose is for research and development.

Vegetarian and vegan food related documents were used as a valuable example as to how topic modeling can be used to identify the major themes within a given corpus. The thesis intends to investigate a new possible manner in which Nofima becomes able to analyse large amounts of unknown web based texts. More specifically, the core of the public opinion with regard to vegetarian and vegan food was studied.

In this chapter, the intention is to answer the research question by discussing each of the stages within the process of analysis. There are five different stages, among which the first three are of greatest interest and value in answering the research question. The idea is to discuss whether the theory matches or derogate from the findings within the method and result sections.

The manual reading allowed for a better starting point as a ground for comparison with the performed study, with regard to the background, theory, methods, and results. The manual reading of a smaller sub-section of the documents discovered that there existed several major concerns and issues. It was discovered that many of the documents were identical, which seemed to be a result of Webhose.io's collection method. When data is collected it typically seems like their methodology only requires the words from the search query to be included and does not have any way of comparing collected texts to those already collected. This means that, for example, if a news article is published on several different sub-divisions of the publisher, the same text will be collected multiple times. A specific example of this is shown in Figure 5.1, where the same article was published in both *Stamford Advocate* and *The Hour*, which both are part of the Hearst Media Services.

5.1 Data collection

Throughout the stage of data collection, the goal was to gather as much relevant information as possible, in order to have a better foundation for the topic modeling. As such, it is believed that it was correct to use an external company for web-harvesting, due to the fact that it had the possibilities to collect large amounts of data in a short period of time.

5.1.1 Identical documents

As seen from the results, and especially the manual reading, 40 % of the news, 53 % of the discussions, and 72 % of the blogs within the collected corpus was not believed to be relevant to vegetarian and vegan food. The manual reading also revealed how several documents were collected multiple times. As a result, there is no doubt that a more careful evaluation of the collected texts would have proved that there were several identical documents within the corpus. These could easily have been removed by comparing the percentage of similarity between the documents ahead of the preprocessing.

Interestingly enough, the study shows that *news* were found to have the most occurrences of several identical documents. Typically, large news corporations own multiple news papers, which in turn publish each others news articles. The result is misleading frequency of information. Not accounting for this, might affect the final topics as some information is counted multiple times.

5.1.2 Lack of representativeness

The collection methodology does, however, not allow for easy detection of representativeness. Within most research and market analysis it is important to be aware of the representativeness. In other words, it is important to be able to reach the entire population, by analysing a smaller sample of the population, which represents the population [48]. This smaller sample group should be picked by a probability sampling method. This creates a random selection, which is needed to withdraw conclusions within the empirical data [48]. Using this method it is possible to generalize the results, however, a margin of error should be taken into consideration. According to Gisle Andersen, a sampling group of about 1000 people, will result in a margin of error of about plus or minus three percent [48]. Sampling groups are often picked based upon sex, age and socioeconomic characteristics. More information about selection process can be found within *Valg av informanter* by Gisle

Andersen [48].

This methodology is harder to follow when collecting texts from the web. In numerous cases, there is seldom any information about the writer's sex, age or socioeconomic status. On top of this, GDPR regulations make it even harder to collect such information. As a result, the methodology used within this study is a non-probability sampling method [48]. In this situation one can no longer generalize the result. This means that there is a chance that the margin of error increases and further the results are less reliable and no longer statistical representative [48].

When this is said, a non-probability sampling group, which is what web-based text analysis is, can be used as a first step in new fields of research to establish some general ideas. This can further be used to create a hypothesis about unknown research or user groups. [49]

Web-harvesting allows for a much larger group of people to be analysed simultaneously. It should give a broader detection of opinions. This said, the people who express their opinions online would usually be the ones that have either strong positive or negative opinions towards a given topic. With respect to this thesis, the texts collected about vegetarian and vegan food, are therefore most likely optioned either positively or negatively. This correlates somewhat with the manual reading performed within this study, where the results show that news were mostly objectively opinionated, while blogs were distributed equally between objectively and subjectively opinions and discussions were subjectively opinionated. As the manual reading method requires that the researchers are objectively opinionated, there is yet another margin of error in this, as humans are naturally subjective in their opinions [50].

There is one more concern with web-harvesting which was identified with the manual reading. As seen from the result, numerous texts were found to be identical which might affect the final topics. This may also skew the view of the opinions found within the corpus.

All in all, the benefit of being able to analyse a large amount of texts simultaneously, gives great possibilities of creating hypothesises for further and deeper analysis. The type of methodology can be used within varying fields of study. It can also be applied on the detection of new ideas regarding further product development.

5.1.3 Search query

Defining a perfect search query for the collection of texts is essential with respect to the number of documents that one is able to collect, as well as, the content of the documents. The study found that small changes in the query gave large variations in the amount of data available. Less strict search queries opens for a collection of millions of documents. The strictness of the search query is dependent upon the number of ”and” used within the search, as well as, perfectly spelled words.

This is why the words within the search query only included one ”and”, which was used to make sure that the word ”food” and at least one vegetarian/vegan word was included. However, as shown from the results of the manual reading, a large number of the documents were non-related to vegetarian and vegan food. As such, there is reason to believe that a more strict query with more uses of ”and” possibly would have given less non-relevant texts. On the other hand, an even broader search query, with less focus on vegetarian and vegan food words, could possibly have given better foundation of comparison to non-vegetarian/vegan food.

It is also important to take incorrect spelling into account. A word within a query needs to be 100 percent identical to the word found within a document to be collected. This means that if the writer of a relevant or non-relevant text has written a word incorrectly, it no longer matches with the query, and thus will not be collected. This is why several of the vegetarian/vegan words within the query were written with multiple spellings and open endings. In other words, the strictness of the query will influence the number of documents collected.

5.1.4 Texts to be applied for further analysis

The findings showed that groups of documents with varying lengths and types of medium, such as blogs, news and discussions, were found. The types of medium were identified by the external company, while the lengths were decided based upon literature. Some literature indicated that short texts (tweets) and long texts (books) were found to have less reliable results [27]. However, there are studies which disagree with the findings regarding short texts. Hall et al. [51] and Tong and Zhang [52] are both examples of this.

Based upon both the literature and multiple trials within this thesis, a middle ground was decided as the optimal text lengths. These have been named ”medium lengths” throughout the thesis. The trials identified the time-commitment and computer capacity limitations with longer texts. It also identified the lack of human interpretability of topics regarding short texts. The literature has shown that medium length texts tend to give more reliable results [27].

5.1.5 Summary data collection

In summary, the main findings within data collection shows that large amounts of data can be collected. However, it is crucial to be aware of the likelihood of identical texts. Instead of regarding the lack of representativeness as a drawback, the findings suggest that the focus should be on the possibility to use the methodology to create hypotheses for further development. This stage also identified the importance of correct defined search queries as well as most appropriate document length.

5.2 Preprocessing

The steps within the preprocessing aimed at preparing the documents for the analysis process. As such, the documents needed to be converted into machine readable numbers. Computers cannot analyse text directly, which is the reason why preprocessing is necessary.

The study found that the steps of preprocessing were highly affecting the quality of the results. Including or removing different preprocessing steps, changed the word distribution in each topic and the topic distribution within each document. In other words, the order of the words within the results and the topics describing the different documents were highly affected by the steps of preprocessing. It was also identified that the order of some of the steps affected the results. With the large variation of preprocessing options to chose from, it is important to be aware of the desired final result. Different preprocessing techniques will typically give varying results.

The study also identified that the preprocessing stage was a never ending story. This is why a circular loop is shown on the graphical representation of the pipeline in Figure 3.1. Throughout the entire process, the steps were updated multiple times, allowing for small improvements within the resulting topics. The topics became easier to interpret with a larger degree of valuable words related to vegetarian and vegan food.

5.2.1 Tokenization and lemmatization

Tokenization is essential for the purpose of most text mining, as words within documents need to be split into separate words also known as tokens. By doing this, the user becomes able to perform further preprocessing and also detect symbols, numbers and punctuation. These are suggested to be removed, as they would otherwise affect the resulting topics negatively. Such information is, in most cases, not needed to interpret topics.

The next step was to perform lemmatization or stemming to reduce words to their dictionary- or root form. Lemmatization was chosen over stemming for several reasons. First of all, stemming only removes inflections and derivational suffixes to reduce words into their stems, while lemmatization uses a pre-trained model to reduce the words into a dictionary form [31]. As a result, the vocabulary would have looked differently and probably been harder to interpret as word stems not necessarily would be correctly spelled. Lemmatization, on the other hand, compares the words within the documents to a pre-trained model of varying language, such that the words are reduced to their dictionary form. These pre-trained models also enable for some spelling errors to be corrected within the documents that are to be analysed. Misspelled words would necessarily not be considered as unique words. Lemmatization, however, allows for misspelled words to be corrected into their dictionary form. This would allow for a more condensed vocabulary, with less misspelled words and thereby give better and more accurate final topics.

Secondly, stemming often results in over- and under-stemming where conceptual groups are assigned incorrectly[31]. This would possibly have increased the size of the vocabulary as more words are kept in varying forms. Lemmatization, on the other hand, is not to be affected by this as it assigns words based upon a pre-trained model. Lastly, the process of lemmatization allows for a part-of-speech tagging per word, which further can be used to reduce the vocabulary [31]. Here words are assigned a unique part-of-speech tag and thereby the vocabulary can be restricted to only certain tags. Stemming does not have this as an option and, as such, the vocabulary would be increased and less condensed.

5.2.2 Removal of stop-words

Stop-words were removed with a list of stop-words, created from Spacy and NLTK. It should, however, be taken into consideration that the list of stop-words could have been extended with more words which had a high frequency count within the corpus. If a word is repeated among most topics, it does not give any value to the interpretation of that specific topic. This would have allowed for a more condensed vocabulary and possibly better interpretation.

There is one more concern with removing stop-words that should be taken into consideration. Some stop-words might completely turn around the meaning of a phrase within a sentence. As explained within the theory section 2.2.3, some stop-words should be considered removed from the list of stop-words in order to keep the meaning of the phrase. An alternative would be to create bi- and tri-grams prior to the removal of stop-words, as it would allow for some of these phrases to be detected and included.

5.2.3 Bi- and tri-grams

The fourth step of the preprocessing was to create bi- and tri-grams. Such word-combinations are not necessary for proper functionality of a topic model, which is why it is optional to include. Within this study it was included as it allowed for a better understanding of words occurring next to each other. It creates a more condensed vocabulary where words have the possibility to be interpreted differently. Instead of, for example, ”salt”, ”lake” and ”city” to be interpreted as salt and a lake and a city, the user perceives the words as a word combination.

Interestingly, these word combinations did not show within the visualisations created with pyLDAvis until after the λ value was set to about 0.3 or below. This indicates that such word combinations are, in most cases, exclusive to only a few topics, as smaller λ values increase the exclusivity of a given word to a selected topic [44].

5.2.4 Vocabulary

In the last step of preprocessing, documents were converted into numbers and a vocabulary was created. The vocabulary was then reduced, as explained in the method section. This was further used to create bag-of-words vectors. These vectors excluded the importance of certain words, as all words were weighted equally [24, p. 259].

Optimally, tf-idf should have been considered at an earlier stage in order to account for the equal weights within bag-of-words and receive alternative results for evaluation [24, p. 261]. Comparing the results of this methodology to the pipeline used within this study, would have been intriguing.

5.2.5 Summary preprocessing

As seen from the above discussions, there exist a large number of possibilities to improve the steps of preprocessing. It seems like the first focus in improving the steps should be on optimizing the vocabulary, both with respect to the size and also the importance of words occurring in texts. Implementing tf-idf has shown great results in other research and should therefore be considered first.

5.3 Topic modeling

Within this step, the hardest decision was to find the optimal number of topics. As explained in the methods section, the coherence measure was used as it is an automated process and thus requires less human knowledge and time commitment. With the research question in mind, this also makes sense as the goal was to validate the possibilities of exploring machine readable texts with topic modeling. However, researchers or businesses would most likely have revealed even higher quality topics and covered a larger degree of the corpus by applying word intrusion with experts. The time commitment to perform word intrusion within an explorative stage does not seem to be reasonable.

5.3.1 Coherence

Coherence measure is greatly studied among researchers within the field of topic modeling. Most of them indicate that they have received relevant information on the optimal number of topics [43, 53]. As shown within the result of this thesis, the optimal number of topics was found to be between 9-11, 11-13 and 11-13 for discussions, news, and blogs, respectively. The values are given as ranges due to the fact that a step size of 2 was used. However, the analysis of the topics revealed that several of the topics within each sub-category did not have any relation to the expected field of study (vegetarian and vegan food). They did not include many vegetarian or vegan based words and thus seemed to be irrelevant. An example of such is shown in Figure 4.7 within the result section. It is, however, important to remember that a person with a higher degree of expert knowledge within the field of vegetarian and vegan food might find the topic within Figure 4.7, very useful.

With the optimal topic number given as a range, it is important to be aware of a certain condition with LDA. If a user trains a topic model with 11 topics, these cannot be compared directly to a 12 topics model, as the user will not obtain the same 11+1 topics. This is due to the probabilistic nature of LDA [14]. The 12 topics would rather be completely new, but still hopefully remind of the content from the 11 topics model.

A question that arose was, do the topics cover the entire content of the corpus? Answering this seems impossible, but as indicated within the manual reading validation, a large degree of the content appears to be covered within the given topics.

5.3.2 Hyper-parameter tuning

Alternative ways of improving the result, would be to investigate the use of Dirichlet parameters, η and α. By applying these with a grid search there is the possibility of improving the quality of the final topics. Such hyper-parameters are known to affect the results according to Tang et al. [27]. Low values of either parameters would indicate that the user believes that the word and topic distributions are sparse [14]. In other words, the number of words within each topic with high probability is fewer than if the α value was set to a higher number. The same goes for the number of topics related to each document; if the value is low, the number of topics is sparse per document. On the other hand, if both values were set at a high value, it would indicate that the word and topic distributions would be evenly distributed [14]. Low values are usually only applied if the user has some previous knowledge about the corpus [14].

5.3.3 Size of vocabulary

A reduction or expansion of the vocabulary would greatly change the result. As explained within section 5.2, a reduction or expansion could be performed by the creation of bi- and tri-grams, the strictness of stop-words removal as well as alternative libraries used for lemmatization. This change would be visible within the resulting topics and may cause the interpretation of the topics to be easier. A correctly created vocabulary most often will be very valuable to the user, as he or she would be able to get a better understanding of the content within the corpus. In other words, fewer disturbing words would increase the chance of a correct interpretation by the user.

5.4 Visualisation

The visualisations were made with pyLDAvis, where the λ value was set to 0.6 based upon literature [44]. As seen within the results in tables 4.1 - 4.6, many vegetarian and vegan words were identified. As such, there is no doubt that the topic models can detect several important aspects to be taken into consideration for further investigation. Experts within the field of study are needed to be able to detect the underlying meaning, as well as the correlation to the other words within each topic.

The visualisation also identified that a lower value of λ was needed to include more of the bi- and tri-grams created within the preprocessing steps. It is important to be aware of that the λ value does not change the LDA model, it only affects the visualisations. The lower λ value shows that many of these word combinations have lower occurrences across the corpus, while they seem to occur more frequently within a smaller range of documents. As shown from the validation, several documents were found to be identical, which indicates that one should not rely too much on these bi- and tri-grams. The creation of such word-combinations are namely dependent upon the existence within a sufficient number of the documents across the corpus to be created. In other words, if a word combination only occurs within a unique document, it will not be added to the vocabulary. However, if this same identical document is repeated 100 times in the corpus, the word has a possibility of being added to the vocabulary. If the concern of identical texts is solved, these lower λ values can be used as a tool to get more accurate interpretations of the topics. It would possibly allow someone with expert knowledge within the field of study to detect the themes more accurately.

As explained within section 5.3.1, the optimal topic numbers were given as ranges. The same section also explains how the resulting topics change every time the topic number is updated. This means that an identification of a perfect topic number is extremely hard to perform, and sometimes seems impossible. It might not be necessary to cover all content with a model, as long as the major aspects are covered. It is often times enough for an expert to be able to get some meaningful insight and thus be able to create an hypothesis. Then, for further stages of a longer development process, word intrusion can be used for more accurate topic validation and thereby allow for even better hypothesis creation.

All in all, the findings from the visualisations from within this thesis show evidence that blogs seem to mostly focus on healthy diets and recipes, which relates to the findings from the manual reading. The manual reading indicated that blogs focused on nutrition and animal welfare. Nutrition is most likely similar to healthy diets. However, the topic model did not show many evidences of animal welfare within the blogs data set. Comparing the findings to literature, an important focus seems to be the health benefits of vegetarian and vegan diets [54]. The literature also focuses on how certain people have negative characteristics associated with vegetarians, which is not as easy to detect within the topic model nor the manual reading[54].

Discussions on the other hand, seem to focus on animals and vegetarian/vegan topics which do not have a strong correlation with the manual reading. Some keywords within the questionnaire from the manual reading of the discussion data set show evidence of animal welfare as words such as, ”animal welfare, animal cruelty and killing animals” are included. Literature do also agree with discussions, and evidence show that there exist lots of people who eat vegetarian and vegan food as a result of animal welfare [55].

Lastly, news were found to focus on launching of plant based menus and were very restaurant focused. Similarities were found within the manual reading as it was found to focus on nutrition. Several of the keywords were found to be directly related to the manual reading as it included words such as, McDonald, meatless-whopper, restaurant-reviews and menu. Once again, literature also indicates an increased launching of vegetarian and vegan options within restaurant [56].

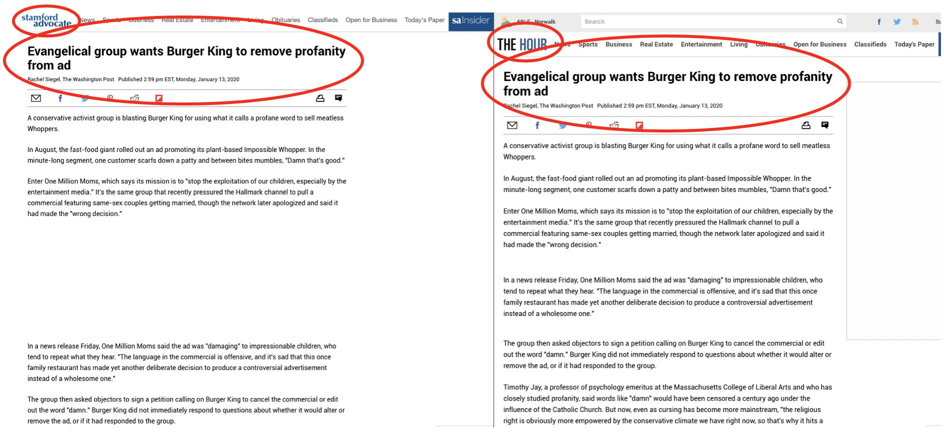
Evaluating the overall quality of the visualisations, there is no doubt that the information detected relates to the literature as well as the manual reading. However, experts within the field of vegetarian and vegan food would have done a better job at interpreting the overall meaning.

Visualisations could also have been performed with alternative tools, such as word clouds or simple tables. However, as these were found to be even harder to interpret, pyLDAvis was decided to be used.

5.5 Validation

Within the manual reading, several valuable findings were detected. As shown within the results, a large percentage of the documents proved to be non-relevant, which means that the reader found no correlation between the text and vegetarian and vegan food. This said, the manual reading of a total of 340 documents were performed by 11 different persons, who all have their own opinions on what is relevant and non-relevant. Humans are known to be subjective, which adds yet another margin of error. This is why some of these numbers can be argued to be incorrect.

The manual reading also found that a large percentage of the news documents were identical to one another. Figure 5.1 is only one example of a situation where Webhose.io has collected identical texts. Looking deeper into this, it seems like many news papers are owned by the same holding company and thereby publishes the same articles within several of their news papers. Blogs and news, on the other hand, did not show many such occurrences.



Evangelical group wants Burger King to remove profanity from ad

Rachel Siegel, The Washington Post Published 2:59 pm EST, Monday, January 13, 2020

A conservative activist group is blasting Burger King for using what it calls a profane word to sell meatless Whoppers.

In August, the fast-food giant rolled out an ad promoting its plant-based Impossible Whopper. In the minute-long segment, one customer scarfs down a patty and between bites mumbles, "Damn that's good."

Enter One Million Moms, which says its mission is to "stop the exploitation of our children, especially by the entertainment media." It's the same group that recently pressured the Hallmark channel to pull a commercial featuring same-sex couples getting married, though the network later apologized and said it had made the "wrong decision."

In a news release Friday, One Million Moms said the ad was "damaging" to impressionable children, who tend to repeat what they hear. "The language in the commercial is offensive, and it's sad that this once family restaurant has made yet another deliberate decision to produce a controversial advertisement instead of a wholesome one."

The group then asked objectors to sign a petition calling on Burger King to cancel the commercial or edit out the word "damn." Burger King did not immediately respond to questions about whether it would alter or remove the ad, or if it had responded to the group.

Evangelical group wants Burger King to remove profanity from ad

Rachel Siegel, The Washington Post Published 2:59 pm EST, Monday, January 13, 2020

A conservative activist group is blasting Burger King for using what it calls a profane word to sell meatless Whoppers.

In August, the fast-food giant rolled out an ad promoting its plant-based Impossible Whopper. In the minute-long segment, one customer scarfs down a patty and between bites mumbles, "Damn that's good."

Enter One Million Moms, which says its mission is to "stop the exploitation of our children, especially by the entertainment media." It's the same group that recently pressured the Hallmark channel to pull a commercial featuring same-sex couples getting married, though the network later apologized and said it had made the "wrong decision."

In a news release Friday, One Million Moms said the ad was "damaging" to impressionable children, who tend to repeat what they hear. "The language in the commercial is offensive, and it's sad that this once family restaurant has made yet another deliberate decision to produce a controversial advertisement instead of a wholesome one."

The group then asked objectors to sign a petition calling on Burger King to cancel the commercial or edit out the word "damn." Burger King did not immediately respond to questions about whether it would alter or remove the ad, or if it had responded to the group.

Timothy Jay, a professor of psychology emeritus at the Massachusetts College of Liberal Arts and who has closely studied profanity, said words like "damn" would have been censored a century ago under the influence of the Catholic Church. But now, even as cursing has become more mainstream, "the religious right is obviously more empowered by the conservative climate we have right now, so that's why it hits a

Figure 5.1: *Two identical collected texts. The following figure shows an example of how Webhose.io has collected identical texts. The figure is only a visualisation to better understand how identical texts have been collected many times.*

To solve the issue of identical texts, one could have checked and compared texts with already collected texts. This would have allowed for a text to be included only once. The concern with the same text being collected multiple times, is that the texts suddenly seem more important to the model, than what is actually the case. Removing such identical texts would also allow for a change with respect to the vocabulary as ”special” words occurring within these identical texts would get less important to the overall vocabulary. These words would thereby have a chance of being removed within the vocabulary reduction.

The manual reading also showed that the questionnaire which followed the reading should have been created after a larger amount of texts were read. This would have changed the wording of some of the questions and possibly allowed for even better validation of the results and a deeper understanding of the type of data that one has worked with. It could be argued that some reading also should have been performed, even before the preprocessing, allowing for detection of weaknesses within the data set. However, it would have taken away some of the purpose of the thesis, as this was to check whether or not a set of documents could have been explored, not knowing much of the content in advance but still understanding the major aspects afterwards.

The questionnaire included a set of keywords which were found to relate to the findings within the resulting topics. However, as these words were asked to be created by the reader and not to be taken from the actual texts, a direct comparison is not plausible. Interestingly enough, several words were found to be identical, such as ”plant based”, ”market”, ”eat”, ”meat”, meal”, ”vegetarian”, ”diet” and ”vegan”, among others. For further studies, it is recommend to ask the reader to take words directly from the documents, as this would allow for better comparisons.

Yet another finding is that news, blogs and discussions were found to have slightly different reasons for meat free diets. It shows that people have varying opinions on these topics. It further identifies that nutrition is a common focus area among the three different sub-categories and gives reason to believe that this should be considered as a focus area for further marketing campaigns within vegetarian and vegan food.

5.6 Further applications of topic modeling

There are several possible ways of using topic modeling. After the creation of topics, they can be used to identify similar texts within a different corpus. By this, a user analyses and creates topics from a known corpus, before he or she trains a model on an unknown corpus and then uses a built-in function in Gensim to identify cosine similarities [57]. The idea is then to identify some of the most relevant topics found in the known corpus and only use these for further analysis and comparison.

This seems extremely valuable within varying fields of study. It can for example be used by a researcher who has already created topics from a known corpus. He or she can then take advantage of this in analysing an unknown corpus and compare it to the topics from the known corpus (only the few most relevant topics are used). In other words, the user could be able to detect documents within the unknown corpus that most probably will relate to the topics created from the known corpus. This can further be used among researchers and businesses to evaluate a set of documents and then pick out the topic that seems most interesting for their field of study, and then use the new corpus to pick out the text that seems most valuable.

Topic modeling can also be used to identify semantics in a set of documents by analysing the semantics of the words within each topic. If words are found to be mostly negatively loaded based upon human detection, it can indicate that the texts have a slightly more negative opinion towards a topic. On the other hand, positive words can mean that the topics are more positive. This type of analysis can be very valuable in the detection of user opinions for all types of markets and businesses. However, as seen, this methodology is based upon human judgment, and therefore requires a great amount of time. The manual reading performed within this study could have been used as a validation for such analysis, but as there were no questions which focused on this, it cannot be used. Understanding the semantics of texts can be very valuable to businesses who tries to figure out people's opinion on a product.

Even though manual semantic detection, in most cases, would be the best, there are studies which have aimed at identifying automatic semantic relationships. An example is *A Semantic Cover Approach for Topic Modeling* by Venkatesaramani et al. [58]. As seen, they get some meaningful insight but further work is needed.

All in all, a semantic detection within a topic model would allow companies to analyse large amounts of reviews and they would quickly get an exploring overview of the topics as well as a semantic opinion within the topics. This could further be used to create even better and more accurate hypothesises for new product development. If people working with product development, not only have information about themes, but also about people's opinions, the developers would be able to direct their hypothesises in an even better direction.

5.7 Further work

Within the entire discussion, there are several concerns that have revealed and should be focused on. This said, further work should therefore focus on the awareness of collected texts with regard to identical texts and a higher degree of relevant texts. To account for this, it is important to make better use of the external web-harvesting companies by trying multiple search query combinations. Making the query less or more strict should be considered.

Further work should also pay more attention to hyper-parameter optimization with regards to the parameters within the LDA model. This said, an alternative is to implement tf-idf to identify the possibility of an improved vocabulary.

More investigation and comparison of word intrusion and coherence scores should be identified. This would also allow for the possibility of better topic interpretations.

Lastly, more time and focus on the actual interpretations and evaluations of the resulting topics should be performed. With this, the λ value within the visualisations should be adjusted even more and compared with the resulting topics. Including experts within the field of study should also be considered.

Chapter 6

Conclusion

Within this study, it was found that topic modeling is a method which can be used to explore and detect some valuable information within a given corpus. It allows the user to collect valuable information about a field of study in the initial stages of the process, to further develop relevant hypothesises. The methodology will allow the user to be pointed in the right direction at an early stage. Instead of starting with an ocean of possibilities and information, the user can create an opinion as to what the optimal direction is. Even with the lack of representativeness within this method, the methodology allows the user to get a broad introductory overview of the field of study.

However, there are numerous possible pitfalls which can cause the user to identify and utilize incorrect topics. Examples of pitfalls could be non-relevant documents included within the corpus, incorrect preprocessing steps and wrongly identified topic numbers. Within the stage of preprocessing, there are yet another set of pitfalls, such as including or excluding some stop-words, too strict reduction of vocabulary and not accounting for words relevancy (tf-idf). By using these incorrect topics, there is a chance of creating useless and false hypothesises. This said, it is believed that the benefits greatly outweigh the disadvantages.

The study also identified improvement possibilities by reading a smaller subsection of the corpus. One example of such, is more awareness on the collected corpus, with regard to relevancy and identical texts. In addition, improved grid-search for hyper-parameters for all stages of the model, and more focus on deciding upon the perfect vocabulary, should also be addressed within further work. On top of this, further investigation of the specific topics by experts should be addressed as they would be able to draw conclusions and create more valuable hypothesises which non-experts necessarily would not be able to do.

Bibliography

[1] L. Vidal, G. Ares, L. Machín, and S. R. Jaeger, "Using twitter data for food-related consumer research: A case study on "what people say when tweeting about different eating situations"," *Food Quality and Preference*, vol. 45, pp. 58–69, 2015.

[2] W. Lidwell, K. Holden, and J. Butler, *Universal principles of design, revised and updated: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design*. Rockport Pub, 2010.

[3] W. Zhao, J. J. Chen, R. Perkins, Z. Liu, W. Ge, Y. Ding, and W. Zou, "A heuristic approach to determine an appropriate number of topics in topic modeling," in *BMC bioinformatics*, vol. 16, no. 13. Springer, 2015, p. S8.

[4] N. Mukherjee, S. Neogy, and S. Chattopadhyay, *Big Data in ehealthcare: Challenges and Perspectives*. CRC Press, 2019. [Online]. Available: https://books.google.no/books?id=o_6PDwAAQBAJ

[5] H. Team, "Internet statistics facts (including mobile) for 2020," Mar 2020. [Online]. Available: https://hostingfacts.com/internet-facts-stats/

[6] C. Schneider, "The biggest data challenges that you might not even know you have," May 2016. [Online]. Available: https://www.ibm.com/blogs/watson/2016/05/biggest-data-challenges-might-not-even-know/

[7] A. Hotho, A. Nürnberger, and G. Paaß, "A brief survey of text mining." in *Ldv Forum*, vol. 20, no. 1. Citeseer, 2005, pp. 19–62.

[8] U. of Edinburgh. (2017, Feb) Prof. david blei - probabilistic topic models and user behavior. Youtube. [Online]. Available: https://www.youtube.com/watch?v=FkckgwMHP2s

[9] C. D. Manning, C. D. Manning, and H. Schütze, *Foundations of statistical natural language processing*. MIT press, 1999.

[10] D. Kr, "Journey through nlp research - basics," Jul 2017. [Online]. Available: https://medium.com/@deepukr85/journey-through-nlp-research-part-1-2e93fdeaad9c

[11] O. Davydova, "10 applications of artificial neural networks in natural language processing," Aug 2017. [Online]. Available: https://medium.com/@datamonsters/artificial-neural-networks-in-natural-language-processing-bcf62aa9151a

[12] Sciforce, "Ai hardware and the battle for more computational power," Nov 2019. [Online]. Available: https://medium.com/sciforce/ai-hardware-and-the-battle-for-more-computational-power-3272045160a6

[13] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[15] A. Amado, P. Cortez, P. Rita, and S. Moro, "Research trends on big data in marketing: A text mining and topic modeling based literature analysis," *European Research on Management and Business Economics*, vol. 24, no. 1, pp. 1–7, 2018.

[16] E. Niiler, "An ai epidemiologist sent the first alerts of the coronavirus," Jan 2020. [Online]. Available: https://www.wired.com/story/ai-epidemiologist-wuhan-public-health-warnings/

[17] C. Stieg, "How this canadian start-up spotted coronavirus before everyone else knew about it," Mar 2020. [Online]. Available: https://www.cnbc.com/2020/03/03/bluedot-used-artificial-intelligence-to-predict-coronavirus-spread.html

[18] N. Ko, B. Jeong, S. Choi, and J. Yoon, "Identifying product opportunities using social media mining: application of topic modeling and chance discovery theory," *IEEE Access*, vol. 6, pp. 1680–1693, 2017.

[19] H. Pillai, "Covid-19 and new age technology," May 2020. [Online]. Available: https://www.thedispatch.in/covid-19-and-new-age-technology/

[20] A. McCallum, X. Wang, and N. Mohanty, "Joint group and topic discovery from relations and text," in *ICML Workshop on Statistical Network Analysis*. Springer, 2006, pp. 28–44.

[21] J. Chang, J. Boyd-Graber, and D. M. Blei, "Connections between the lines: augmenting social networks with text," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 169–178.

[22] L. Hong and B. D. Davison, "Empirical study of topic modeling in twitter," in *Proceedings of the first workshop on social media analytics*, 2010, pp. 80–88.

[23] S. Wachter, B. Mittelstadt, and L. Floridi, "Why a right to explanation of automated decision-making does not exist in the general data protection regulation," *International Data Privacy Law*, vol. 7, no. 2, pp. 76–99, 2017.

[24] S. Raschka and V. Mirjalili, *Python Machine Learning: Machine Learning*

and Deep Learning with Python, scikit-learn, and TensorFlow. Packt Publishing Ltd, 2017.

[25] R. Mitchell, *Web scraping with Python: Collecting more data from the modern web*. " O'Reilly Media, Inc.", 2018.

[26] J. Phoenix, "What are the advantages and disadvantages of web scraping data?" Feb 2020. [Online]. Available: https://understandingdata.com/the-advantages-disadvantages-of-web-scraping-data/

[27] J. Tang, Z. Meng, X. Nguyen, Q. Mei, and M. Zhang, "Understanding the limiting factors of topic modeling via posterior contraction analysis," in *International Conference on Machine Learning*, 2014, pp. 190–198.

[28] M. Hajjem and C. Latiri, "Combining ir and lda topic modeling for filtering microblogs," *Procedia Computer Science*, vol. 112, pp. 761–770, 2017.

[29] T. Beysolow II, "Topic modeling and word embeddings," in *Applied Natural Language Processing with Python*. Springer, 2018, pp. 77–119.

[30] S. Vijayarani, R. Janani *et al.*, "Text mining: open source tokenization tools-an analysis," *Advanced Computational Intelligence: An International Journal (ACII)*, vol. 3, no. 1, pp. 37–47, 2016.

[31] V. Balakrishnan and E. Lloyd-Yemoh, "Stemming and lemmatization: a comparison of retrieval performances," 2014.

[32] M. Pitchford, "Preparing your data for topic modeling," Nov 2017. [Online]. Available: https://publish.illinois.edu/commonsknowledge/2017/11/16/preparing-your-data-for-topic-modeling/

[33] B. Bengfort, R. Bilbro, and T. Ojeda, *Applied Text Analysis with Python: Enabling Language-aware Data Products with Machine Learning*. " O'Reilly Media, Inc.", 2018.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[35] H. M. Wallach, "Topic modeling: beyond bag-of-words," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 977–984.

[36] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proceedings of the eighth ACM international conference on Web search and data mining*, 2015, pp. 399–408.

[37] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *Advances in neural information processing systems*, 2009, pp. 288–296.

[38] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the*

22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999, pp. 50–57.

[39] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1.* Association for Computational Linguistics, 2009, pp. 248–256.

[40] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *Advances in neural information processing systems*, 2008, pp. 121–128.

[41] J. Kim, M. Park, H. Kim, S. Cho, and P. Kang, "Insider threat detection based on user behavior modeling and anomaly detection algorithms," *Applied Sciences*, vol. 9, no. 19, p. 4018, 2019.

[42] J. Chuang, C. D. Manning, and J. Heer, "Termite: Visualization techniques for assessing textual topic models," in *Proceedings of the international working conference on advanced visual interfaces*, 2012, pp. 74–77.

[43] S. Syed and M. Spruit, "Full-text or abstract? examining topic coherence scores using latent dirichlet allocation," in *2017 IEEE International conference on data science and advanced analytics (DSAA).* IEEE, 2017, pp. 165–174.

[44] C. Sievert and K. Shirley, "Ldavis: A method for visualizing and interpreting topics," in *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, 2014, pp. 63–70.

[45] J. Sirisha and M. B. Reddy, "Unstructured data: Various approaches for storage, extraction and analysis," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 15, no. 7, 2017.

[46] R. Řehůřek, "gensim topic modelling for humans." [Online]. Available: https://radimrehurek.com/gensim/about.html

[47] Explosion, "Software." [Online]. Available: https://explosion.ai/software

[48] G. Andersen, "Valg av informanter - ndla," Jan 2019. [Online]. Available: https://ndla.no/subjects/subject:19/topic:1:195989/topic:1:195829/resource:1:56943

[49] J. Børsting, "Metoder for datainnsamling: SpØrreundersØkelser, intervju amp; fokusgrupper." [Online]. Available: https://www.uio.no/studier/emner/matnat/ifi/INF2260/h17/timeplan/chapter_5_8-norsk.pdf

[50] H. Rolston, "Are values in nature subjective or objective?" *Environmental Ethics*, vol. 4, no. 2, pp. 125–151, 1982.

[51] D. Hall, D. Jurafsky, and C. D. Manning, “Studying the history of ideas using topic models,” in *Proceedings of the 2008 conference on empirical methods in natural language processing*, 2008, pp. 363–371.

[52] Z. Tong and H. Zhang, “A text mining research based on lda topic modelling,” in *Proceedings of the Sixth International Conference on Computer Science, Engineering and Information Technology (CCSEIT)*, 2016, pp. 21–22.

[53] S. Mahanty, F. Boons, J. Handl, and R. Batista-Navarro, “Studying the evolution of the ‘circular economy’concept using topic modelling,” in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2019, pp. 259–270.

[54] T. Corrin and A. Papadopoulos, “Understanding the attitudes and perceptions of vegetarian and plant-based diets to shape future health promotion programs,” *Appetite*, vol. 109, pp. 40–47, 2017.

[55] S. Burgess, P. Carpenter, and T. Henshaw, “Eating on campus: Vegan, vegetarian, and omnivore stereotyping,” 2014.

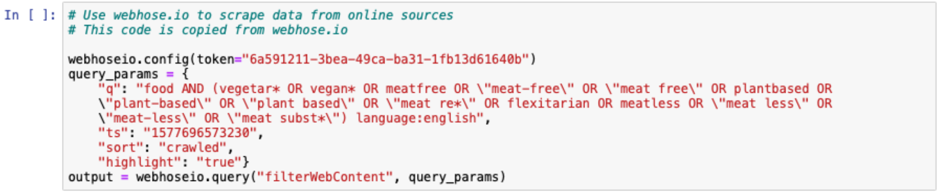
[56] W. J. Craig, A. R. Mangels *et al.*, “Position of the american dietetic association: vegetarian diets.” *Journal of the American Dietetic Association*, vol. 109, no. 7, pp. 1266–1282, 2009.

[57] F. S. Al-Anzi and D. AbuZeina, “Toward an enhanced arabic text classification using cosine similarity and latent semantic indexing,” *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 2, pp. 189–195, 2017.

[58] R. Venkatesaramani, D. Downey, B. Malin, and Y. Vorobeychik, “A semantic cover approach for topic modeling,” in *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 92–102. [Online]. Available: https://www.aclweb.org/anthology/S19-1011

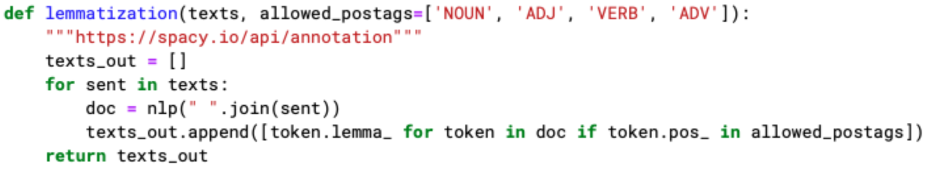
Appendix

Code snippets



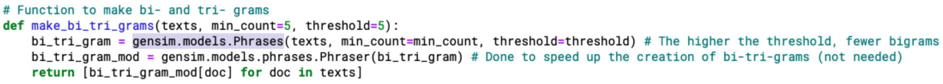
```
In [ ]: # Use webhose.io to scrape data from online sources
        # This code is copied from webhose.io

        webhoseio.config(token="6a591211-3bea-49ca-ba31-1fb13d61640b")
        query_params = {
            "q": "food AND (vegetar* OR vegan* OR meatfree OR \"meat-free\" OR \"meat free\" OR plantbased OR
            \"plant-based\" OR \"plant based\" OR \"meat re*\" OR flexitarian OR meatless OR \"meat less\" OR
            \"meat-less\" OR \"meat subst*\") language:english",
            "ts": "1577696573230",
            "sort": "crawled",
            "highlight": "true"}
        output = webhoseio.query("filterWebContent", query_params)
```

Figure 6.1: *Python implementation of webhose.io. This is a screenshot of how webhose.io is implemented in Python. The numbers shown in the code snippet is critical for proper functioning. The token is similar to an API key, and is uniquely created by Webhose.io for each of their customers; it indicates the allowances given to each customer. The "ts" number is used to decide the number of days back in time that the the documents were included from.*



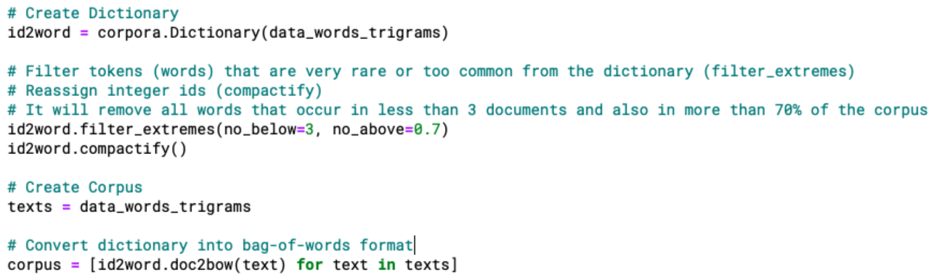
```
def lemmatization(texts, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV']):
    """https://spacy.io/api/annotation"""
    texts_out = []
    for sent in texts:
        doc = nlp(" ".join(sent))
        texts_out.append([token.lemma_ for token in doc if token.pos_ in allowed_postags])
    return texts_out
```

Figure 6.2: *Python implementation of Spacy lemmatization. This screenshot shows how Spacy lemmatization is implemented.*



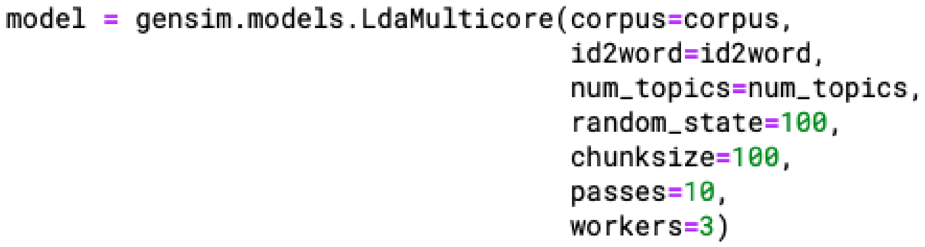
```
# Function to make bi- and tri- grams
def make_bi_tri_grams(texts, min_count=5, threshold=5):
    bi_tri_gram = gensim.models.Phrases(texts, min_count=min_count, threshold=threshold) # The higher the threshold, fewer bigrams
    bi_tri_gram_mod = gensim.models.phrases.Phraser(bi_tri_gram) # Done to speed up the creation of bi-tri-grams (not needed)
    return [bi_tri_gram_mod[doc] for doc in texts]
```

Figure 6.3: *Python implementation of bi- and tri-gram creation. This screenshot shows how Gensim Phraser is used to create bi- and tri-grams.*



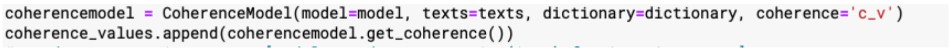
```
# Create Dictionary
id2word = corpora.Dictionary(data_words_trigrams)

# Filter tokens (words) that are very rare or too common from the dictionary (filter_extremes)
# Reassign integer ids (compactify)
# It will remove all words that occur in less than 3 documents and also in more than 70% of the corpus
id2word.filter_extremes(no_below=3, no_above=0.7)
id2word.compactify()

# Create Corpus
texts = data_words_trigrams

# Convert dictionary into bag-of-words format
corpus = [id2word.doc2bow(text) for text in texts]
```

Figure 6.4: *Python implementation of bag-of-words. The following shows how bag-of-words can be implemented within Python using Gensim's corpora.Dictionary function and Gensim's doc2bow function. Further, it shows how the vocabulary is reduced, and new ids are assigned.*



```
model = gensim.models.LdaMulticore(corpus=corpus,
                                   id2word=id2word,
                                   num_topics=num_topics,
                                   random_state=100,
                                   chunksize=100,
                                   passes=10,
                                   workers=3)
```

Figure 6.5: *Python implementation of LDA multicore. The figure shows the parameters that needed to be set by the user to perform the LDA. Multicore allows for multiprocessing and thereby improves the speed.*



```
coherencemodel = CoherenceModel(model=model, texts=texts, dictionary=dictionary, coherence='c_v')
coherence_values.append(coherencemodel.get_coherence())
```

Figure 6.6: *Python implementation of coherence calculations. Gensim's CoherenceModel is used to validate the topic quality of a set of given topics.*

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway