Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Lotfi Amin Lazreg

Topic Modelling for Metadata Extraction and Generation from Norwegian Parliamentary Texts

Master's thesis in Informatics (MSIT)
Supervisor: Ole Jakob Mengshoel
June 2023

NTNU
Norwegian University of
Science and Technology

Lotfi Amin Lazreg

Topic Modelling for Metadata Extraction and Generation from Norwegian Parliamentary Texts

Master's thesis in Informatics (MSIT)
Supervisor: Ole Jakob Mengshoel
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Topic modelling is the process of identifying abstract concepts in a collection of documents. It is a widely used technique for extracting meaningful information from textual data. This thesis focuses on investigating the effectiveness of topic modelling architectures, namely BERTopic, Top2Vec, and LDA, in the context of Norwegian transcribed parliamentary speeches. The study explores various aspects of topic modelling, including preprocessing steps such as stopword removal and the selection of appropriate embedding models.

Evaluating the performance of topic modelling techniques is crucial for their effective application. However, the two current primary methods in user testing and automatic metrics, have clear limitations. User testing has the limitation of requiring users, hence being time-consuming and having a long feedback loop. Automatic metrics have their limitations connected to their relation to human interpretability. To overcome these limitations, the thesis proposes a novel evaluation framework (TopicEval), that leverages expert domain knowledge to qualitatively analyse samples of topics using wordcloud representations. This framework enables domain experts to systematically assess the quality and interpretability of topics across a variety of categories.

We applied the evaluation framework to practical use cases in the experiments conducted, including the analysis of a large-scale textual dataset (NPL). Through the experiments, we observed the usefulness of the evaluation framework in assessing the quality of topics generated by models.

Moreover, we address the challenge of automatic topic labelling and compare two methods, NETL and BERTopic, for generating topic labels. We examine their strengths and weaknesses, considering factors such as coherence, relevance to documents, and domain knowledge.

Sammendrag

Emnemodellering er prosessen med å identifisere abstrakte begreper i en samling av dokumenter. Det er en mye brukt teknikk for å ekstrahere meningsfull informasjon fra tekstdata. I denne masteroppgaven undersøker vi effektiveten til en rekke emneodelleringsarkitekturer: BERTopic, Top2Vec og LDA. Arkitekturene blir testet på en samling med transkriberte parlamentariske taler fra det norske Stortinget. Vi utforsker ulike aspekter ved emnemodellering, inkludert forbehandlingstrinn som fjerning av stoppord og valg av passende innlemningsmodeller.

Evaluering av resultater er avgjørende for å kunne anvende emnemodelleringsteknikker på en effektiv måte. Imidlertid har de to primære metoder for evaluering, brukertesting og automatiske metrikker, klare begrensninger. Brukertesting er tidkrevende og har en lang tilbakemeldingsløkke på grunn av kravet om brukerdeltakelse. Automatiske metrikker er begrenset i forhold til deres forbindelse til menneskelig tolkningsevne. For å overvinne disse begrensningene, foreslår avhandlingen et nytt evalueringsrammeverk kalt TopicEval, som utnytter domenekunnskap for å kvalitativt analysere eksempler på emner ved bruk av ordskyrepresentasjoner. Dette rammeverket gjør det mulig for domeneeksperter å systematisk vurdere kvaliteten og tolkbarheten til emner innen ulike kategorier.

Vi anvendte evalueringsrammeverket på praktiske bruksområder i de gjennomførte eksperimentene. Som inkluderer analysen av et stort tekstbasert datasett (NPL). Gjennom eksperimentene observerte vi nytten av evalueringsrammeverket for å vurdere kvaliteten på temaene generert av modellene.

Videre sammenligner vi to metoder for automatisk emne tittel generering, NETL og BERTopic. Vi undersøker deres styrker og svakheter, med tanke på faktorer som relevans til dokumenter og domenekunnskap.

Preface

This master's thesis was conducted as part of the course IT3920 at the Norwegian University of Science and Technology, spring 2023. This thesis was written and carried out by I, Lotfi Amin Lazreg. The project was supervised by Ole Jakob Mengshoel.

I wish to thank my supervisor Ole Jakob Mengshoel for his valuable insights, suggestions on how to structure my thesis and his positivity.

Many thanks to the people at NRK for their participation in the user test and valuable discussions.

I want to thank my family for their continued support. And a special thanks to my Mother and brother, Sami, for providing me with valuable feedback.

I want to thank my roommates Sepehr, Vegard and Robin for never growing tired of my endless discussions.

A special thanks goes to my good friend Viljar who has always been available for sporadic two-hour-long phone calls.

Contents

Figures

Tables

Code Listings

Chapter 1

Introduction

The introductory chapter of this thesis serves as the foundation for the document and the project it describes. It begins by presenting the rationale behind the research conducted in this thesis in Section 1.1 This background information sets the context for our motivations and objectives, which are outlined in Section 1.3 and Section 1.2, respectively. These objectives then lead to the formulation of the research questions in Section 1.4. Finally, the chapter concludes with an outline of the overall structure of the thesis in Section 1.5, providing a roadmap for the subsequent chapters.

1.1 Background

This master's thesis began with a request to do a project under my supervisor Ole Jabok Mengshoel. The original project had the name "Automatic Metadata Generation". There had been one previous work completed on this same project by Rushfeldt [53]. The work by Rushfeldt focused on using Top2Vec and LDA to process a tailor-made dataset of NRK's Subtitled TV (NST), subtitle files. The idea behind continuing on this project was to use the NST dataset and apply different topic modelling techniques to improve the results. Additionally, an area that had not yet been explored was the field of automatic topic labelling (ATL), it was therefore desirable to explore different ATL techniques to see what could be accomplished. Furthermore, certain weaknesses were identified in the automatic metrics used to evaluate the coherence of the topic models, prompting requests for improvements in the area of evaluating topic models. However, due to privacy limitations, the NST dataset was not available for use. While the objectives remained, a new dataset had to be found.

Every four years there is a Parliamentary election in Norway [1]. Leading up to the election there will be many media-covered debates by various politicians and generally a lot of media coverage, including articles and polls and general history about politics. News broadcasters want to create interesting content for their audience and many broadcasters strive to cover different parts of the discussion.

Throughout the spring and fall semesters, we had various interactions with among others, Egil Ljøstad, at the Norwegian Broadcasting Corporation (Norsk rikskringkasting) (NRK). It was evident that there was a keen interest in leveraging any intriguing results that could be obtained from our research. With this in mind, we aimed to provide NRK and any other media entities with useful insights and findings in the realm of topic modelling on political speeches, allowing them to explore practical use cases based on their specific requirements and interests.

When looking for datasets in the realm of political discussions, there are many datasets available in multiple languages including Norwegian. The large availability of political datasets is due to the project "Parlamint: Towards Comparable Parliamentary Corpora", financially supported by CLARIN ERIC. ParlaMint is a project with the goal of contributing to the creation of comparable and uniformly annotated multilingual corpora of parliamentary sessions [2]. Parlamint was conducted in two stages, the first one (ParlaMint I) running from July 2020 to May 2021. The second stage, ParlaMint II ran from December 2021 to May 2023. The dataset we will be focusing on is ParlaMint-NO [3], created as a part of ParlaMint II by the AI-lab at the National Library of Norway (NbAiLab). The NbAiLab have created multiple datasets and artifical neural network models that are made available for use [4], more of their work is discussed in Section 2.5. We will be using a processed version of ParlaMint-NO that we are naming Norwegian Parliament-Large (NPL). A subset of this dataset will also be used that was created by NbAiLab, and that we have named Norwegian Parliament-Mini (NPM).

1.2 Motivations

Topic modelling (TM) if done manually is a time-consuming task. Manually finding abstract topics in a collection of documents is difficult and becomes infeasible as the size of the corpora increases. Topic models are therefore a useful alternative to

[1] Norwegian Parliament

[2] More information about ParlaMint can be found at this URL: `https://www.clarin.eu/parlamint`

[3] Link to ParlaMint-NO dataset: `https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-77/`

[4] `https://huggingface.co/NbAiLab`

do the job. Topic modelling can be used for a variety of tasks. The most popular task is the straightforward task of finding abstract topics. Other tasks include word sense disambiguation [40] , bioinformatics research [11], summarization [41], recommendation [30] and search engine optimization [24], to name a few. The applications run far and wide and new areas of application are tested continuously. The ability to discover topics is very useful for summarization while recommender systems can use the property of finding the most related topic to a word or topic to topic. Additionally, dynamic topic models that can track topics over time are useful to see changes in interests and which political areas are the most discussed as well as how the discussions around topics themselves change over time.

Automatic topic labelling (ATL) is useful when visually representing the topics to humans. If you want to show a user the topics in a more raw format, the topic label is a good way to do that. The raw format can be very useful for professionals such as journalists who work with large collections of documents and quickly have to get an overview of what is available and has been written about. If topic models are used for recommender systems [59], automatic labelling can be used for the explainability of the recommendations.

1.3 Objectives

The primary objective of the master's thesis in 2023, is to conduct a comprehensive evaluation of various embedding models to assess their performance in the context of Norwegian parliamentary sessions. The thesis aims to investigate how different embedding models, such as Doc2Vec and Sentence Transformer models, can effectively capture and represent the semantic meaning of Norwegian parliamentary texts.

Furthermore, we aim to develop a systematic qualitative evaluation method for assessing topic model results. This systematic approach will enable domain experts to make somewhat standardized assessments of the quality and interpretability of the generated topics. By establishing a qualitative evaluation framework, the thesis intends to reduce the need for crowd-sourced human evaluations and enhance the effectiveness of evaluating topic models.

In addition to evaluating the topic models, the thesis will explore and compare various automatic topic labelling methods. Automatic labelling techniques will be employed to try to generate descriptive and meaningful labels for topics.

By conducting these evaluations and experiments, the master's thesis seeks to contribute to the field of topic modelling in parliamentary sessions. The thesis aims to identify the most suitable embedding models, establish a systematic qualitat-

ive evaluation approach for topic models, and evaluate different automatic topic labelling methods. The findings from this work will provide valuable insights into improving the understanding, interpretation, and practical application of topic modelling techniques in the domain of parliamentary sessions in Norwegian.

1.4 Research Questions

Based on the work done during the Master in Informatics Preparatory Project: IT3915, in the fall semester of 2022, overall goals for the master thesis in the spring of 2023 were shaped. Four research questions were created to serve as a guide to keep the work on track. The research questions also served as measurements of success.

Research question 1 (RQ1): What topic modelling techniques exist and how do they perform on Norwegian transcribed parliamentary speeches?

Research question 2 (RQ2): What automatic topic labelling techniques exist and how do they perform on Norwegian transcribed parliamentary speeches?

Research question 3 (RQ3): How can the qualitative evaluation of topic models be improved?

Research question 4 (RQ4): How do the topic modelling techniques from R1, perform when evaluated with the improved qualitative evaluation metrics from R3?

1.5 Thesis Structure

The structure of my thesis follows the outline presented below. It is important to note that this thesis builds upon the work conducted in the Master in Informatics Preparatory Project: IT3915, which took place during the fall semester. Most of the content has been updated, but certain sections still retain their original form, particularly in the related work and background sections.

Chapter 1 - Introduction

The current chapter provides an overview of the thesis, including its background, motivations, objectives, research questions, and the overall structure of the thesis.

Chapter 2 - Background

This chapter presents the problem area and applications of topic modelling and automatic topic labelling. Before giving the necessary background on topic modelling-specific techniques as well as detailing preprocessing and the most important areas of artificial neural networks relevant to this thesis.

Chapter 3 - Related work

The related work chapter explores different topic modelling techniques within the two major realms: generative probabilistic and neural models. It also examines the Word2Vec and Doc2Vec architectures. The chapter also explores some literature on evaluation metrics for topic models as well as the field of automatic labelling of topics. Furthermore, a specific Norwegian Transformer Model (NorBERT) is studied as part of the related work analysis.

Chapter 4 - Methodology and Architecture

Chapter 4 describes the methods and techniques common to multiple experiments. It covers the embedding models used for different topic models, hyperparameters, dataset analysis of NPM and NPL, as well as the explanation of the automatic topic labelling techniques and the development of the evaluation framework.

Chapter 5 - Experiments and Results

Chapter 5 presents the experimental plan and provides detailed explanations for each experiment, including setup details and reproducibility information. The chapter presents the results of each experiment and discusses their implications. The experiments are interconnected and do not necessarily follow a linear order, with connections between experiments illustrated in Figure 5.1.

Chapter 6 - Conclusion and Future Work

The final chapter summarizes the work conducted, highlights the contributions made, and presents potential areas for future research and development.

Chapter 2

Background

In this chapter, we will provide the necessary background knowledge to establish a solid foundation for the subsequent chapters. We begin by describing the problem area in Section 2.1, where we introduce the notation, terminology, and provide a definition of topic modelling.

Next, in Section 2.2, we delve deeper into the most relevant topic modelling methods. We explore the primary approaches along with the commonly used methods in those approaches.

Moving on, in Section 2.3, we examine the metrics used to evaluate the performance of topic modelling. We discuss both qualitative and quantitative metrics that are commonly used to assess the quality of topic models. In Section 2.4 we present a taxonomy of preprocessing techniques. Lastly, in Section 2.5, we discuss the relevant areas of artificial neural networks (ANNs) in the context of topic modelling.

2.1 Problem Area

This section will first cover some basic notation and terminology before defining topic modelling. Afterwards, we will show some examples of topic representation and lastly discuss automatic topic labelling.

2.1.1 Notation and Terminology

This work is mainly focused on text collections and therefore the language of text collections will be used and it is useful to define some notations and terminology. A token is defined as a word with an index in the document. A document the entity

input is separated into. Each document contains tokens that are separated by blank space and can be of various lengths. A document is in essence a sentence or a longer text depending on the entities at hand. A corpus or corpora is a collection of documents.

2.1.2 Topic Modelling Definition

Topic modelling can be defined as finding a set of abstract topics that fits a collection of documents. Although various approaches and methods exist, the fundamental concept remains consistent across all topic models. Topic modelling requires a corpus of m documents, denoted as $D = d_1, ..., d_m$, along with a vocabulary of n words, $V = v_1, ..., v_n$. The purpose of a topic model M is to generate a collection of K topics, represented as $T = t_1, ...t_j, ..., t_k$. A topic model can hence be defined as $M(D, V) = T$. Topic models will typically infer the vocabulary from the input documents. Each document in the corpus is associated with none, one or more topics. Their respective relevance can be defined as: $R_{d,M}(d_i, t_j)$. Additionally, each topic t_j consists of a collection of h topic words from the vocabulary V, expressed as $t_j = w_1, ..., w_h$. These topic words are assigned weights, indicating their significance within the topic: $R_{t,M}(t_j, w_h)$.

The concept of "representativeness" plays a crucial role in topic modelling, as it measures the degree to which a word aligns with a particular topic and how well a topic captures the essence of a document. Higher representativeness scores for a specific topic and document imply that the topic effectively captures the themes present in the document. Similarly, higher representativeness scores for a word and topic indicate its strong association with the given topic. Typically, representativeness is measured using probabilities in generative probabilistic models, while neural topic models employ similarity measures such as cosine similarity.

In summary, topic modelling aims to extract abstract topics from a collection of documents by assigning topics to documents and estimating the significance of words within those topics.

2.1.3 Topic Representation

A topic consists of a list of words or a probability distribution over words. For visualization purposes, a cut-off point of around 10 or 20 words is often chosen. An example of a topic that could be generated from some documents about basketball: [basket, ball, score, three-point, dunk, shoot, dribble, players, guard, forward]. Table 2.1 shows a list visualization of the example topic. The left value refers to the topic word and the right column value is the representativeness of the token to the topic. List visualizations are commonly used to visualize topics, along with

wordclouds. Figure 2.1 shows a wordcloud representation of the topic, where the size of each topic word is based on their representativeness of the topic. The words inside a topic should preferably have some semantic connection to each other as well as being somewhat diverse to fully cover the abstract topic they together represent.

Topic word	Word representativeness
basket	0.15
ball	0.14
score	0.10
three-point	0.09
dunk	0.08
shoot	0.06
dribble	0.05
players	0.04
guard	0.03
forward	0.03

Table 2.1: Example topic generated from a collection of documents about basketball.

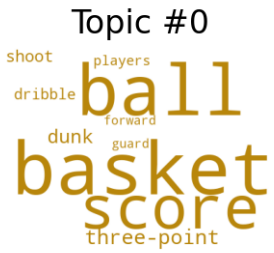


Figure 2.1: Wordcloud created from the topic listed in Table 2.1

2.1.4 Automatic Topic Labelling

Automatic labelling of topic models is the act of finding labels that describe topics to make them humanly comprehensible. The usual way of visualizing topics is by showing the topic words either as a list or a wordcloud. Long lists of words or wordclouds are hard to interpret for humans because the information is in a format we are not used to comprehend. Therefore, the idea is that a word or phrase that summarizes the topic well is preferable. The naive way to labelling topics is

simply choosing the top-topic word. In the case of the example topic in Table 2.1, the chosen word would be "Basket". This label is obviously quite ambiguous, because what kind of basket is it referring to? It would be much preferred to have the label of "Basketball", which in our example topic is not even contained in the top-10 topic words. Another weak point of selecting the first topic word as the topic label is that in some topics none of the topic words themselves describe the topic. An example of this would be a topic consisting of different colours: [red, green, blue, white]. The first topic word is red, but that would not be a good topic label, however, "colour" would be. Further methods for generating topic labels are described in Section 3.6.

2.2 Topic Modelling Methods

In the field of topic modelling you can generally classify the different topic models as probabilistic generative, the so-called classical models, or as neural topic models, topic models that include neural networks in some part of the architecture. The classical topic models have had good performance historically, but the improvements have slowed down in the era of big data and deep learning [60]. Meanwhile, neural topic models have emerged as an area of interest. The architecture of neural topic models typically include: dimensionality reduction techniques such as UMAP [45], clustering techniques such as HDBSCAN [44] and word embeddings such as Word2Vec [47] or Sentence Transformer models [52].

One of the key differences between probabilistic generative models and neural topic models is that probabilistic models seek to find topics that recreate the original document word distributions. Neural topic models on the other hand leverage word embeddings and their semantic information to find topics.

2.2.1 Generative Probabilistic Topic Models

The development of probabilistic topic models in the field of natural language processing has seen various advancements over the years. It began with the release of Latent Semantic Indexing (LSI) by Deerwester *et al.* [18], which introduced a method of analyzing collections of documents to discover statistical co-occurrences of words. This approach was further expanded upon by Hoffmann *et al.* [28], with the introduction of the probabilistic LSI (pLSI) model, which introduces the assumption of latent topics and in addition to focusing on word frequencies, also takes into account the probability of a word appearing in a document given a specific topic.

One significant milestone in the field of probabilistic topic modelling is the Latent

Dirichlet Allocation (LDA) model, proposed by Blei *et al.* [8]. LDA is an extension of pLSI that introduced a Bayesian framework that addresses some of the limitations of pLSI, which will be further discussed in Section 3.1.1.

LDA has since become one of the most widely used and influential models in the field. Most other advances in generative probabilistic models such as PAM [39], D-LDA [7], GK-LDA, [16] are either extensions of LDA or build upon the same concepts.

2.2.2 Neural Topic Models

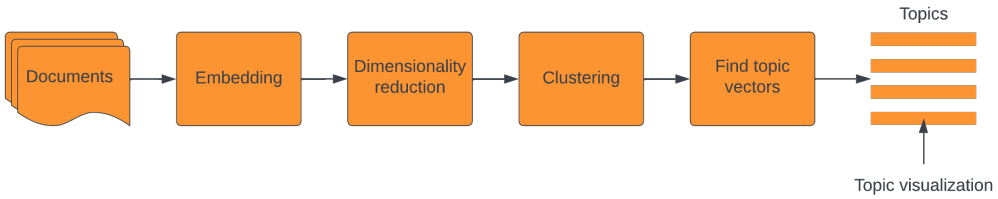


Figure 2.2: Flowchart showing the different steps of a generic neural topic model.

Neural topic models have in common that they do not seek to find topics to recreate the original document word embeddings. Instead, they typically leverage word embeddings and clustering to generate topics based on semantic similarity between documents. An example of a generic neural topic model can be seen in Figure 2.2.

Some of the neural topic models are extensions of LDA such as LDA2Vec [31]. Other models are based on using the output from pre-trained language models to derive document and word embeddings. The methods vary a bit depending on what task and dataset is being used. One area with major breakthroughs involves using pre-trained language models to generate document word embeddings and then clustering to find topics. Top2vec [4] and BERTopic [26] utilize modifications of this method and have shown state-of-the-art results on various benchmark datasets. The idea behind the method is that the pre-trained language models have a fine-grained ability to capture the semantic meaning of text. The clusters in the document embedding space then represent documents that are semantically similar and represent what a topic is and can therefore be used as topics. In addition to this, there are other methods as outlined in Zhao et al. 2021 [60]. More of these methods will be described in Section 3.4.

2.2.3 Embeddings

When applying artificial neural network (ANN) methods to natural language processing (NLP) tasks, the first layer of the neural networks will often be an embedding layer. This layer requires real-value data as input. The data for NLP tasks is most often in the format of lists of strings. To transform the strings to real-valued data, embeddings are used. Depending on how long sequences of data are, the embeddings can be word embeddings, document embeddings or sentence embeddings among others. One of the simpler embeddings is the bag-of-words (BoW) model [1]. There are different ways to create a BoW model, one of them is through building a vocabulary from the unique words in the document collection. The embedding of each sentence or document will then be a list of 0s and 1s where the 1s represent that the word exists in the sentence and 0s represent non-existence. The problem with this method is that it assumes that the order of words in a document can be neglected in the language of probability theory this is called exchangeability for the words in a document [3]. Another problem is data sparsity. Document collections will often have much larger vocabularies than the respective document lengths. the effect is that the BoW representation of each document will contain a lot of 0s compared to the 1s. This means that the information will be spread out hence sparse.

Word2Vec was introduced by Mikolov *et al.* [47]) as an alternative to BoW models. Word2Vec tries to combat the data sparsity issue as well as taking word order into account. The main idea behind the model is to create a vector space where similar words are close to each other. High-quality word vectors are created from huge datasets with millions of words and thousands of words in the vocabulary. The results from using the techniques are very interesting in that the similarity of word representations goes quite deep and some algebraic operations can be done and output semantically meaningful data. For example, if we take the vector representations of "boy" and "adult" and add them to each other we would get the vector representation of "man" as so: Vector("boy")+ Vector("adult") = Vector("man"). These vector operations being possible is a result of being in a vector space where semantic meaning is contained. Further explanation of the Word2Vec architecture can be found in Section 3.2. Another method to generate word vector embeddings is Sentence-BERT [52]. SBERT is explained more in detail in Section 2.5. The basic idea is to use the output from BERT and add a pooling operation to the output to derive a fixed-size sentence embedding.

2.2.4 Clustering Algorithms

Clustering is a well-known machine learning technique employed for unsupervised learning tasks, aiming to identify inherent groups or clusters within a given dataset.

Xu *et al.* [58], provide a comprehensive list of clustering algorithms, among which centroid-based clustering, density-based clustering, and hierarchical clustering are particularly relevant in the context of topic modelling.

Centroid-based clustering methods organize data into non-hierarchical clusters, with K-means being the most widely used algorithm in this category. In the context of topic modelling, a weighted K-means algorithm is described in more detail in Section 3.4.5.

Density-based clustering approaches group data points based on areas of high density, enabling the formation of clusters with arbitrary shapes. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [23] is a commonly used density-based algorithm that has been applied in various domains.

An extension of DBSCAN, called HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), proposed by Malzer *et al.* [44], transforms the algorithm into a hierarchical clustering method. HDBSCAN provides a more robust and flexible clustering approach by incorporating a hierarchical structure.

HDBSCAN is the default clustering algorithm in both Top2Vec [4] and BERTopic [26]. This choice is motivated by HDBSCAN's ability to handle clusters of varying density and its effectiveness in discovering clusters with complex shapes, making it suitable for topic modelling tasks where the data may exhibit diverse cluster structures.

2.2.5 Dimensionality Reduction Techniques

Word embeddings are often used for neural topic models. When dealing with word embeddings, it is common to encounter high-dimensional output. For instance, the default dimension size for Top2Vec is 300[1], while the default embedding model used in BERTopic, "sentence-transformers/all-MiniLM-L6-v2," produce an embedding with an output dimension size of 384 [2].

The high dimensionality of word and sentence embeddings poses several challenges, collectively known as the curse of dimensionality [5]. One of these challenges is data sparsity. The high dimensionality causes each point in the vector space to contain little information as the information is dispersed across many dimensions. Additionally, distance metrics lose their effectiveness in high-dimensional spaces, as the distances between the nearest and furthest points tend to converge. This

[1] Top2Vec github

[2] sentence-transformers/all-MiniLM-L6-v2

poses a significant issue for clustering algorithms, as distance metrics designed for low-dimensional spaces may not work well in high-dimensional spaces [2].

To address these challenges, dimensionality reduction techniques aim to reduce the dimensionality of the data while preserving its local and global features. Various methods have been developed for this purpose, including Principal Component Analysis (PCA) [43], t-SNE [56], and Uniform Manifold Approximation and Projection (UMAP) [45]. UMAP, in particular, has garnered attention for its ability to preserve local and global features of high-dimensional data. It is especially well-suited for use with language models that exhibit varying embedding dimensions, as it imposes no computational restrictions on the embedding dimensions. Notably, UMAP has demonstrated superior performance compared to PCA and t-SNE in preserving the structure of high-dimensional data [45]. Consequently, UMAP is utilized in BERTopic [26] and Top2Vec [4] for dimensionality reduction purposes.

2.3 Metrics

Because topic modelling is commonly approached as an unsupervised learning task, traditional NLP metrics such as accuracy, precision, and recall, which rely on labelled data, are not applicable. Instead, topic models are evaluated using different metrics designed for assessing coherence and coverage as outlined by Churchill and Singh [17]. Quantitative and qualitative metrics have been developed for these areas of assessment with the goal of measuring human interpretability. Human evaluation is costly and time-consuming and it is hard to keep the evaluations objective. For these reasons, automatic measures are preferred. However, it is important that the automatic metrics correlate with human interpretability as that is often the end goal. Although some studies have demonstrated the correlation between automatic metrics and human evaluation [35, 50], there are definitely limitations to these results such as the lack of generalizability across different topic modelling methods and datasets [29]. Hoyle *et al.* [29], looked deeper into the evaluation of topic models and the importance of human evaluation. Section 3.5 presents the work of Hoyle *et al.* [29], while Section 4.4 focuses on the development of a qualitative evaluation framework. The objective of this framework is to overcome the limitations of existing quantitative evaluation methods and minimize the reliance on crowd-sourcing for qualitative evaluation by incorporating the expertise of domain experts.

2.3.1 Quantitative Evaluation Metrics

As previously mentioned the primary metrics used for evaluating topic models are coherence and coverage.

Coverage tries to measure how well the documents in the corpus are covered by the topics produced. Coverage can be divided into two types: document coverage and topic coverage. Topic recall is the most prevalent topic coverage measure. And is defined as the fraction of ground truth topics that the topic model produces [17]. Document coverage measures how well documents are represented by topics. Topic model accuracy is one method used to evaluate document coverage. It is defined as the fraction of documents that are correctly given the right topic as the ground-truth topic. Topic recall and topic model accuracy both necessitate ground-truth topics. There is another method for measuring coverage that can be used without ground truths; Held-out perplexity. The approach to calculating held-out perplexity starts with splitting the dataset into training and test-sets. You then calculate the log-likelihood of the unseen documents and use this as a measure for model performance where a high likelihood indicates a better model. According to Chang et al. 2009 [15], held-out perplexity is not strongly correlated to human judgment which in the end is what we really want to measure, and the metric will therefore not be used.

Coherence tries to measure the individual topic quality. The most common coherence metric is pointwise mutual information (PMI) [17]. PMI draws on the intuition that the best way to weigh the connection between two words is to measure how much higher the probability that the two words occur together in the corpus compared to the probability that it would happen by chance. PMI has been shown to correlate with human interpretability and achieve results at or nearing the level of human annotation correlation [50], [36]. PMI of given word pair (w_i, w_j) is defined in (2.1) as following Newman *et al.* [49] (2009).

PMI is a measure of how much the actual probability of a particular co-occurrence of events $P(w_1, w_2)$ differs from what would be expected based on the probabilities of each event.

$$\mathrm{PMI}(w_i, w_j) = \ln\left(\frac{P(w_i, w_j)}{P(w_i) \cdot P(w_j)}\right) \tag{2.1}$$

PMI has the properties of being 0, When the events $P(w_1)$ and $P(w_2)$ are independent. The lower bound of PMI is $-\infty$ and the upper bound is ∞. An observed property of PMI is that it values frequently occurring words highly and as a way to combat this, a normalized pointwise mutual information has been developed. NPMI was first presented by Bouma [10]. The purpose of the normalization is to introduce upper and lower bounds for the values. NPMI has an upper bound

of 1 in the case of complete co-occurrence and a lower bound of -1 representing the absence of co-occurrences. As well as retaining the property of being 0 from PMI.

$$\text{NPMI}(w_1, w_2) = -\frac{\text{PMI}(w_1, w_2)}{\ln(P(w_1, w_2))} \tag{2.2}$$

Another often-used metric is diversity. Diversity tries to measure how different topics are from each other or how much they overlap. A way to measure it is to calculate the percentage of unique words in the set of topics produced by the topic model.

Diversity is an evaluation method that looks at word occurrences across all the topics and favors topic sets where each topic is unique. A diversity score of 1 means that each word in every topic only occurs once. That's both within the topic and across the topic set. A diversity score of 0 would mean that every word occurs in every topic, or in another way; every topic is identical. The equation to calculate topic diversity can be seen in (2.3). The prerequisites to calculate topic diversity is to have a set of topics. The method is then to count the number of unique words in the top-k topic words, where k corresponds to the number of topic words that are used to represent the topic, usually 10. The number of unique words in the top-k words of the topic set is then divided by k times the number of topics.

$$\text{topic-diversity}(T, k) = \frac{\text{number of unique words in top-}k\text{ words of each topic}}{k * \text{number of topics}} \tag{2.3}$$

The quantitative metrics that will be used in this work are coherence through PMI and NPMI, and diversity. The coherence metrics are chosen because of their correlation with human interpretability previously discussed. Diversity is chosen because it is a metric that is based on the coherence of the whole topic set.

2.3.2 Qualitative Evaluation Metrics

Quantitative evaluation of topic models through automatic metrics is often not enough in itself to ensure the quality of topic models. Human evaluation is sought after, however, it is not trivial to make tests of topic models suitable for humans. The problem is quite a complex one, but there have been developed some decent tests. Chang *et al.* [15], presented the word-intrusion task and topic intrusion task. The direct rating task was presented by Newman *et al.* [50]. These three tasks are the most common ones and will be our focus for qualitative evaluation. In addition to these tasks, other questions such as "What is this topic missing for it to be highly

rated?", can be included to give more insight as to why topic models are rated as they are.

The qualitative evaluation metrics are based on human testing. Typically humans are presented with topics produced by topic models with the task of assessing them in some way. The tests are developed to try and measure the human interpretability and understanding of the topics.

The word intrusion task is a task where a test subject is presented with a topic and typically the top-10 words from the topic where one of the words is replaced with a topic word from another topic. The task is then to correctly identify which word does not belong, the so called "intruder".

The topic ranking test is a simple test where the test subject is to rank topics on an ordinal scale typically from 0 to 3 based on typically either the interpretability or perceived usefulness.

For the topic intrusion task, the test subject is presented with some parts of a document such as the title and a snippet of the text. Along with the document the subject is presented with the three highest probability topics assigned to the document as well as a randomly chosen topic. The task is then to correctly identify the topic which does not fit the document. The topic intrusion task tests whether a topic model correctly labels the documents in the corpora in a way that agrees with human opinion. The topic intrusion tasks relies on topic models to produce multiple topics per document, which is a feature of generative probabilistic topic models. However the neural topic models generally only gives one topic per document. In these cases an inverse variant can be used, where the test subject is presented with some parts of a document similarly to the non-inverse task. The topics presented however now includes three "intruder" topics along with the "real" topic. The task is then to correctly identify the "real" topic.

Examples of the tasks mentioned can be found in Section 5.4.

Because of the often unsupervised nature of topic modelling, automatic measures do not always correlate with human interpretability. Even when we see increases in a model's automated coherence measured by PMI, it does not always imply an improvement in the human scores. This is because of the uncertainty of human judgments. Another issue with quantitative and qualitative metrics for topic modelling, brought up by Hoyle *et al.* [29], is that the metrics are not properly validated for neural topic models. Further discussion on this topic can be found in Section 3.5.

2.4 Preprocessing

When training a classical topic model on large datasets with noisy and unstructured data they tend to perform poorly [17] due to the fact that the topic models try to recreate the original word distribution which favors frequent words that do not contain signficant semantic meaning so called "stopwords". Churchill and Singh [17], defined and formalized a preprocessing taxonomy. The taxonomy contains four different preprocessing classes.

- **Elementary pattern-based preprocessing**: This type of preprocessing is necessary for most datasets. It includes cleaning up punctuation, lower casing all words and combining words to increase semantic meaning.
- **Dictionary based preprocessing**: Cleaning up stopwords comes under this rule-class as well as matching synonyms with each other.
- **Natural language preprocessing**: Natural language processing (NLP) techniques are commonly used to increase semantic meaning in datasets. Some of these methods include lemmatization, stemming and Part of Speech (POS) removal. When preprocessing for classical topic models NLP techniques can be used to increase the semantic meaning of the dataset because the model itself does not contain this information.
- **Statistical preprocessing**: The statistical methods are quite powerful and serves to reduce the token size by using information about the collection of tokens. The two mainly used methods are Collection Term Frequency(TF) cleaning and Term Frequency Inverse Document (TF-IDF) [54], cleaning. TF cleaning removes words with high and low frequency, while TF-IDF combines the term frequency with its inverse document frequency to find the term relevance and removes tokens with a low relevance.

2.5 Artificial Neural Networks

Artificial Neural Networks (ANNs) have emerged as a powerful computational framework inspired by the intricacies of the human brain. In the realm of topic modelling, ANNs have proven to be instrumental in extracting meaningful insights from textual data.

One prominent technique within ANNs for dealing with textual data is the use of Transformer networks. Transformers is a deep learning architecture, introduced by Vaswani *et al.* [57], designed to process sequential input data. Before transformers, most state-of-the-art NLP systems relied on gated Recurrent Neural Networks (RNNs), such as Long short-term memory (LSTM), with added attention mechanisms. Vaswani *et al.* [57], showed that the attention part of the transformer

model alone manages to outperform an RNN. The superior performance and the ability to parallelize the training, make Transformers the preferred choice for many machine-learning tasks.

BERT [19], is a pre-trained transformer network, which produced state-of-the-art results for various NLP tasks. A disadvantage of BERT is that independent sentence embeddings are not computed and the vector space sentences are mapped to is not suited to be used with common similarity measure such as cosine-similarity. The effect of this is that Transformers create poor sentence representations out of the box. Additionally, pre-trained Transformers require heavy computation to perform common tasks such as finding the most similar pair of sentences. For many applications, this does not matter, but for topic modelling the vector space is particularly interesting because it can be used to find topic vectors.

Reimers *et al.* [52], presents SBERT, a fine-tuned version of BERT. SBERT is an architecture consisting of two BERT networks with connected weights (siamese networks). The output from each BERT network is passed to a pooling layer to get fixed-size sentence embeddings. There are different pooling strategies, but one of the best performing is the MEAN-strategy which is to take the mean of all output vectors. Afterwards, based on the objective function different things are done to the sentence embeddings. For one of the best performing ones, the classification objective function, the sentence embeddings are concatenated with their element wise difference. This result is then multiplied with the trainable weights learned through a softmax classifier. The output of SBERT is sentence embeddings in a vector space where the distances between vectors have semantic significance. This is of clear value and can be used for NTMs.

The process in SBERT does not require BERT, any transformer network could be utilized. The general idea behind Sentence Transformers is to take the output of transformer networks and apply pooling such as mean pooling to get a fixed length embedding. One could use the transformer network directly to create the embeddings, but as Reimers et al. [52] showed: pre-trained Transformers require heavy computation and once trained, Transformers create poor sentence representations out of the box. SentenceTransformers [3] is a Python framework, available on Hugging Face, that offers a large collection of pre-trained models tuned for various tasks. These models can be used as the embedding layer in NTMs. Hugging Face is an open-source platform that hosts a Transformers library, providing many Transformer models available for use in addition to their Datasets library where one can find datasets useful for NLP tasks.

RoBERTa, which stands for "Robustly Optimized BERT Approach," is a state-of-

[3]Hugginface

the-art natural language processing (NLP) model developed by Facebook AI and introduced in Liu *et al.* [42]. It is based on the architecture of BERT, improving it by introducing several modifications to its training methodology. Most notably it employs a larger training corpus and longer a training duration.

Two other notable alternative methods to Sentence Transformers are Transformer Document Embeddings (TDEs) and Transformer Word Embeddings (TWEs). TDEs provide high-quality representations of entire documents, capturing both individual word meanings and contextual relationships. On the other hand, Transformer Word Embeddings focus on generating embeddings for individual words, capturing their semantic properties and relationships within the document.

2.5.1 Norwegian Transformer Model

In Kummervold *et al.* [33], the process of building a large-scale dataset in Norwegian and training a BERT-based language model for Norwegian is shown. The paper tried to mimic the training of mBERT (multilingual BERT), and produce a model trained on Norwegian bokmål and nynorsk. The NB-BERT model is available on Hugging Face [4]. A model we will be utilizing in our work is the nb-sbert-base model [5]. This model is a Sentence Transformers model trained on a machine translated version of the Multi-Genre Natural Language Inference corpus, starting from the nb-bert-base model.

[4]Link to NB-BERT model on Hugging Face
[5]Link to nb-sbert-base on Hugging Face

Chapter 3

Related Work

In the related work chapter, we explore the topic modelling field further. We begin by exploring generative probabilistic topic models in Section 3.1. This is followed by a discussion on Word2Vec and Doc2Vec in Sections 3.2 and 3.3, respectively. Next, we delve into neural topic models in Section 3.4. An interesting article by Hoyle *et al.* [29], on topic evaluation is then presented in Section 3.5. Furthermore, we explore two interesting automatic topic labelling methods in Section 3.6. Lastly, we discuss how the research done is connected to the research questions in Section 3.6.

3.1 Generative Probabilistic Topic Models

The main idea of generative probabilistic topic models is to model topics as a distribution of words with the goal of recreating the original document word distributions. In this section we will discuss four different relevant generative probabilistic topic models.

3.1.1 LDA

Latent Dirichlet allocation (LDA) is a generative probabilistic topic model introduced by Blei *et al.* [8], The general idea of LDA is based on the hypothesis that when a document is being written, the person writing the document has certain topics in mind. Writing a document about a topic can be seen as picking words with a certain probability associated with the topic. Each topic can be seen as a distribution over the words in the vocabulary.

LDA was developed as an advancement to the probablistic latent semantic indexing

(pLSI) model [28]. In the pLSI model, each word in a document is associated with a single topic and different words in documents can be associated with different topics. Each document is then represented through a probability distribution over the fixed set of topics based on the word-topic association. The pLSI model is based on the "bag-of-words" assumption - that the order of words in a document can be ignored. In the language of probability theory this is seen as exchangeability of words [3]. The other assumption is that the order of documents are exchangeable. Two major issues with the pLSI model were identified by Blei *et al.* [8]. The first one was that the number of paramemters in the model grows linearly with the number of documents in the corpus. The second one was that it was difficult to assign probabilities to documents that are unseen. LDA was developed to solve these issues with pLSI and is also based on the idea of exchangeability of words and documents.

Many of the best performing topic models today are adaptions of LDA to account for modern problems. Through introducing embedding spaces (ETM) and general knowledge (GK-LDA), the lexical knowledge contained in models can be increased to help deal with noisy corpora and large vocabularies. Dynamic topic models (D-LDA) are useful to extract topics over time. LDA is at times synonymous with topic models and serves as a good baseline for topic modelling experiments [17].

LDA assumes the following generative process for each document $\mathbf{w}$ in a corpus $\mathcal{D}$:

- Choose $N \sim \text{Poisson}(\xi)$
- Choose $\theta \sim \text{Dir}(\alpha)$
- For each of the N words w_n:
  (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
  (b) Choose a word w_n from $p(w_n|z_n,$

Figure 3.1 shows the plate notation which is a concise way of visualizing the process of LDA. The large rectangle M represents the total number of documents in the corpus. The smaller rectangle N represents the number of words in the documents. The α and β are dirichlet priors. The α stands for the per document topic distributions. A high α indicates that each document is likely to contain a mixture of most of the topics and not one or two in particular. The β is the parameter of the dirichlet prior on the per topic word distribution. A high β indicates that each topic will contain a mixture of most of the words. A low β indiciates that each topic will contain a mixture of few of the words. θ is the topic distribution for document M. z is used to notate each topic. $z_{m,n}$ is the topic for the $n-$th word in document m. The generative process continues until we reach a steady state or

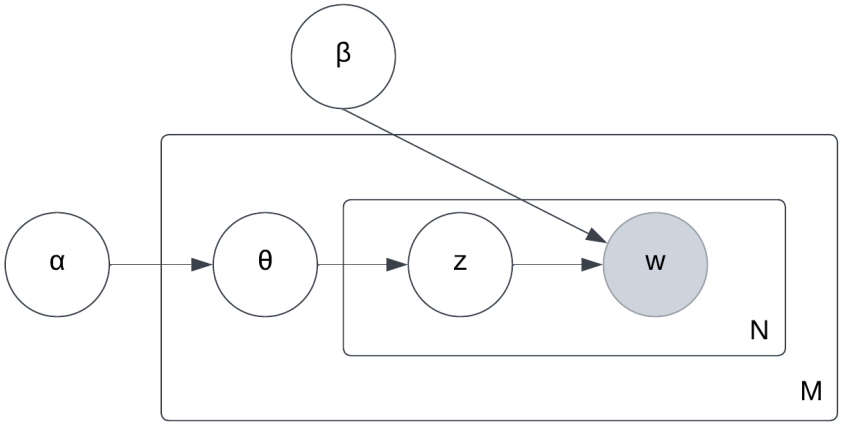


Figure 3.1: Plate notation is a commonly used way to visualize LDA. This Figure is based on Figure 1 from [8].

the desired amount of iterations is reached.

3.1.2 GK LDA

General Knowledge-based LDA (GK-LDA) introduced by Chen *et al.* [16], is an extension of LDA where general knowledge is combined with LDA to give the model more power. The idea is that there is a lot of lexical knowledge about words and their relationships available in online dictionaries and Wordnets and this knowledge can possibly enhance the results of LDA to generate more coherent topics.The lexical knowledge used in GK-LDA is a specific one called lexical semantic relations (LR). The LR are word-to-word relations and include synonyms, antonyms, adjective-attribute relations, hyponym, taxonym etc. An easily available way to include these relations is through Wordnet [48]. Wordnet is a large lexical database of primarily English, but with the option of other languages such as Norwegian as well. A part of the lexical knowledge in Wordnet is of a Theosaurus, but it is even more powerful because it does not just interlink word forms, but specific senses of words.

The lexical relations used in GK-LDA are divided into LR-sets. Each LR-set indicates one sense/meaning of the words inside it. For example if we take the word expensive, an LR-set including antonyms and synonyms could be: expensive, pricey,

cheap. This LR-set indicates one sense of the word and there could be multiple LR-sets for each word.

The architecture of GK-LDA involves introducing a latent variable s, which represents the LR-set assignment to each word. This however has some issues because depending on the domain different LR-sets make more or less sense. For example in the domain of "programming": script, runs, is correct, but for the domain of "running" it does not make sense. The solution to this included in GK-LDA is to give the model ability to choose the LR-set with the right word sense in the modelling. Another issue is that LR-set may contain partially correct relations. For example in the domain "Camera", we have an LR-set 'picture', 'pic', 'flick'. In this LR-set, 'picture' and 'pic' have a relation, but there is no relation between 'picture' and 'flick'. To solve this issue a word correlation matrix is used to estimate the correctness of the LR-set.

In summary, GK-LDA integrates general knowledge in the form of lexical semantic relations into the LDA framework to enhance topic coherence. By incorporating LR-sets and addressing challenges related to word sense and correctness, GK-LDA aims to generate more accurate and meaningful topics. The experiments done by Chen *et al.* , showed good results when comparing GK-LDA with other probabilistic generative topic models.

3.1.3 D-LDA

Another interesting aspect of topic modelling is the time frame of documents. The temporal aspect of documents in topic modelling is of great interest as it allows us to explore how topics evolve and change over time. Depending on the domain and dataset, such as newspaper articles, scientific papers, or political discussions, the chronological order of documents provides valuable insights.

To address the temporal dynamics in topic modelling, Blei *et al.* [7], proposed Dynamic LDA (D-LDA) as a method for temporal topic modelling tasks. D-LDA divides the data into time slices, typically based on a specific time unit like years. Each time slice is then modelled using a K-component topic model, where K represents the number of topics. Importantly, the topics associated with the previous time slice influence the topics in the current time slice, enabling predictions of topic changes in subsequent time slices.

Dynamic topic models are particularly useful for scenarios where data is published sequentially, such as newspapers or scientific articles. By employing dynamic topic modelling, we can observe how topics evolve over time. For example, a topic related to climate change may include words like "ozone" and "atmosphere" in the 1990s, while in 2015, it may feature words like "Paris" and "agreement." Similarly, a

scientific article on "neuroscience" from 1900 would exhibit different topic words compared to an article from 2000, reflecting the changes in the field over time. Another scenario of interest is political discussions such as those covered by NPM and NPL, which are discussed in further detail in Section **??**.

One limitation of traditional LDA is the lack of utilization of document publication dates. By incorporating temporal information through dynamic topic modeling, we can better capture the temporal evolution of topics and align them with the corresponding time periods.

Temporal topic modelling techniques like D-LDA provide valuable insights into how topics change over time, enabling us to analyze the dynamics and evolution of various domains. By considering the temporal dimension in topic modelling, we can gain a deeper understanding of the trends and developments within a given dataset.

3.1.4 PAM

Pachinko allocation model (PAM) was presented by Li and McCallum 2006 [39], with the aim of capturing correlations between topics. The topics discoverd by LDA captures correlations between words, but not explicitly between topics. Topic correlations are however common in real-world text data. If we consider for example a dataset that discusses three topics: health, sports and the weather. Most likely health and sports would co-occur often while weather and health would co-occur less often. LDA struggles with modelling data in which some topics co-occur more often than others. Ignoring the correlations between topics hinders the models ability to predict topics for new data. Additionally the models ability to discover a large amount of highly specific topics. Teh *et al.* (2005) devised hierarchical Dirichlet process (HDP) which can capture correlations between topics. However HDP needs to have a pre-defined data structure which means it can not discover correlated topics automatically.

To capture the correlations between topics, PAM uses a Directed Acyclic Graph (DAG). The leaves in the DAG represent words while the interior nodes reprents topics. If we would simulate LDA as a DAG, the interior nodes would only have leaf nodes as children. Meanwhile, PAM allows interior nodes to have other interior nodes as children. Through these connections, interior nodes in PAM have distributions over both topics and words. This hierarchical structure enables PAM to capture and model the correlations between topics effectively.

The Pachinko Allocation Model (PAM) addresses the limitation of traditional LDA models by explicitly modeling correlations between topics. By utilizing a Directed Acyclic Graph (DAG) structure, PAM captures topic correlations which in some

cases leads to more accurate topic predictions. The incorporation of topic correlations in PAM enhances its performance in modelling real-world text data, where topics often exhibit varying degrees of co-occurrence.

3.2 Word2Vec

The motivations behind Word2Vec were several limitations in the previously most common techniques for distributed representations of words. The biggest limitation was the lack of notion of similarity between words in models such as N-gram and bag-of-word. These models have obvious upsides in their simplicity and robustness, as well as the observation that simple models trained on large amounts of data outperform more complex models trained on less data. The example used in the paper is the N-gram model, that can be trained on trillions of words [12].

Mikolov *et al.* (2013) [47], introduced techniques that can be used to learn high-quality word vectors from huge datasets with large vocabularies. The two main goals of the architectures is to enable training on huge datasets, producing word vectors with high dimensionality. The second goal is to represent similar words close to each other with the notion that words can have multiple degrees of similarity.

Word2Vec proposes two different architectures to generate word-embeddings. A continous bag of word model (CBOW) and a continous skip-gram model. The architectures manages to learn a vector space in which similar words are close to each other based on their degrees of similarity. For example nouns can have different endings and if we search for similar words in the vector space, we can find words that have similar endings. The similarities that the embedding space captures goes beyond syntactic similarities. Because it is a vector space you can perform algebraic manipulations. For instance, if we take the vector representation of "Mother" and subtract "Woman" we would get "Father".

CBOW as an architecture consists of three layers: input, projection and output. At the input layer, N previous words and N future words are encoded using $1-of-V$ coding (one-hot encoding), which encodes the word as a one dimensional matrix with length of the vocabulary and each word in the vocabulary has an unique representation. The $windowsize$ is $2N-1$. The input layer is then projected to a projection layer P. All the words are projected into the same position and their vectors averaged. The data is then passed from the projection layer to a log-linear classifier with the training task of predicting what the middle word is. The predicted word is then the output. The training of CBOW can be seen as training a log-linear classifier to correctly predict a word based on the words around it.

An illustration of the training process of the CBOW model, can be seen in Figure 3.2.

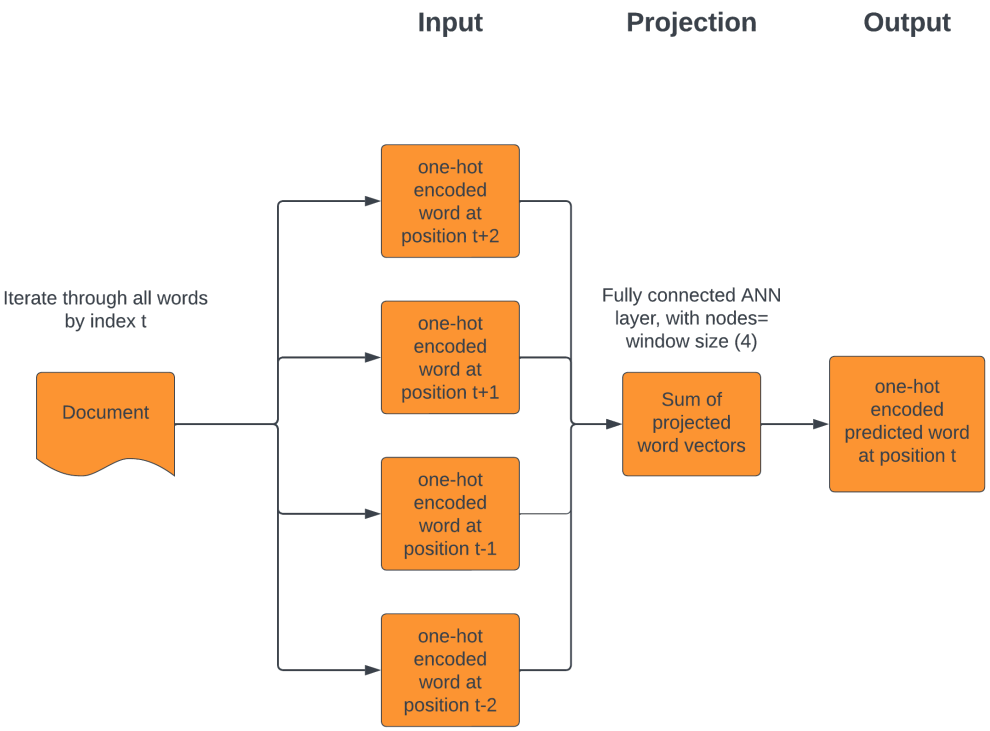


Figure 3.2: Training process of CBOW using $windowsize = 4$.

The continuous skip-gram model (skip-gram) is similar to CBOW, but instead of trying to predict the current word based on the words around it we try to predict the words around the current word taking current word as input and producing words in the $2N$ positions around it, N previous words and N future words. The input to the model is the current word which is projected to the projection layer. The projection layer passes through a log-linear classifier with the training task of predicting the words around the input word. The predicted words are then the output. An illustration of the training process of the skip-gram model, can be seen in 3.3.

Word2Vec has important use-cases in topic modelling among other as a part of Top2Vec (Section 3.4.4 and can be used as word embedding for ETM and other models. It is worth to mention that there have been developed other techniques to generate word embeddings in later years such as GloVe [51] and FastText [9]. However Word2Vec still remains a prominent alternative.

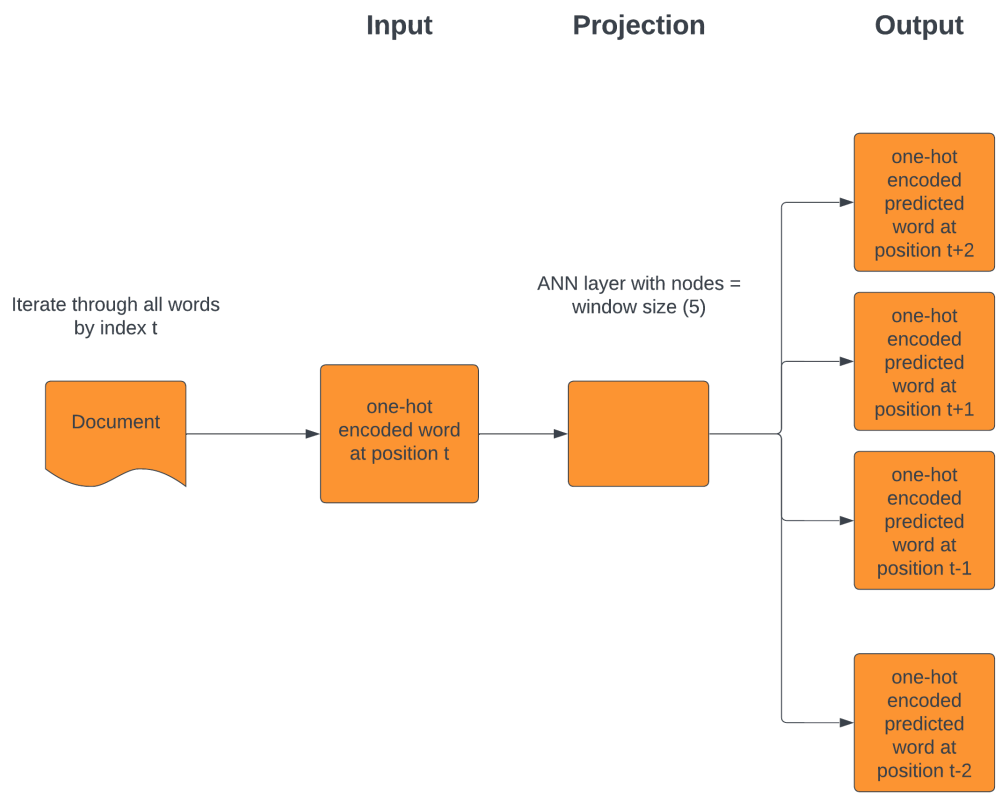


Figure 3.3: Training process of skip-gram using $windowsize = 5$

3.3 Doc2Vec

Mikolov *et al.* [38], presented a way to learn vector representations for variable-sized texts. Doc2Vec is an unsupervised framework that learns continuous distributed vector representations for variable length texts. As opposed to Word2Vec which learns representations for single words, Doc2Vec can take texts ranging from sentences to documents as input.

A document vector is a distributed representation of a document. There are two methods for learning document vectors proposed in the paper. The first one is called the Distributed Memory Model of Paragraph Vectors (PV-DM). This approach is inspired by the CBOW method to learn word vectors in Word2Vec with a specific setup where only previous words of the current word are passed as input to predict the current word. The PV-DM models make a slight modification to

this where it concatenates a document vector to the input word vectors. This document vector acts as a memory of the rest of the document and the context we are in. Each document is represented by a column in document matrix D, and each word by a column in word matrix W. The training task is then to learn D and W through stochastic gradient descent and backpropagation as well as softmax weights.

A second method to document vectors is one without word ordering similar to the skip-gram method in Word2Vec. In this method, only the document vector is used as input and we try to predict randomly sampled words from the document. This version is named Distributed Bag of Words version of Paragraph Vector (PV-DBOW).

The results from using Doc2Vec are word vectors as well as document vectors. The document vectors represent each vector in the embedding space and are a replacement for bag-of-word models. The document vectors contain information about semantic relations between words. Words that often co-occur and are similar to each other are closer in the vector space.

3.4 Neural Topic Models

In this section we will discuss seven different neural topic model architectures. Each approach has some unique characterics. The general approach of neural topic modelling is illustrated in Figure 2.2.

3.4.1 LDA2Vec

LDA2vec was presented by Moody [31], as a model inspired by LDA and word2vec. The goal of LDA2vec is to replace the bag-of-words representation that LDA typically uses and leverage the increased semantic meaning that word embeddings contain. When using bag-of-words representation it is up to the model to extract semantic information, but when using a word embedding the semantic information is contained in the embedding. LDA2vec utilizes a modified version of the skip-gram model from word2vec. In this modified skip-gram model, the current word vector is combined with a document vector, resulting in a context vector. This context vector is then projected and used to predict the words surrounding it. The model incorporates Dirichlet distributed topics, where each topic has a distributed representation in the vector space. The document vectors are derived by combining the document proportions of each topic with the topic vector from the vector space.

3.4.2 ETM

In their work, "Embedded topic model" (ETM), Dieng *et al.* [21], address the challenges faced by LDA when working with large datasets that have large vocabularies including stopwords. ETM combines LDA with word embeddings, specifically using the CBOW variant of Word2Vec, to overcome these limitations and improve topic interpretability.

ETM combines LDA with word embeddings, specifically the CBOW variant of word2vec. ETM uses embedding representations of both words and topics. Topics are represented as vectors in the embedding space as opposed to the distributed representation over words in LDA. To generate topics, ETM measures the agreement between a word's embedding and a topic's embedding. The notion of "agreement" refers to the similarity between the word and topic embeddings. If a word's embedding is similar to a topic's embedding, it is assigned a high probability for that topic.

Dieng *et al.* [21] conducted different experiments to compare LDA and ETM. One of them was to test both topics models for their predictive perplexity of held-out documents. As the vocabulary size increased, LDA's performance decreased while ETM performed better and better. The corpus tested on consisted of 11.2k articles and for 100 topics from them. Another experiment was conducted to demonstrate the ETM's ability to deal with stopwords. A version of the 20NewsGroup [13] dataset where stopwords were retained was used. Topics were generated by LDA and ETM. The result was that LDA included stop words in almost every topic, likely because it struggled to differentiate between content words and stop words. ETM managed to effectively generate interpretable topics except for a few stop-topics that only contain stopwords. This highlights the value of ETM in handling datasets that contain varying genres, because there will be many different domain-specific stopwords.

3.4.3 D-ETM

The Dynamic Embedded Topic Model (D-ETM) was proposed by Dieng *et al.* [20], as a combination of the Embedded Topic Model (ETM) and the temporal aspect of the Dynamic LDA (D-LDA). Like ETM, D-ETM represents each topic as a vector in the embedding space. However, D-ETM incorporates the temporal dimension introduced in D-LDA, allowing for the analysis of topic evolution over time.

In Dieng *et al.* [20], D-ETM is used to analyze the transcriptions of the United Nations (UN) general debates from 1970 to 2015. Inspection of the topics reveals the topics discussed and their change over time, which aligns with historical events.

A topic about climate change was found by D-ETM to transition from the ozone layer in the 1990s to global warming and emissions in 2015. D-ETM was compared to D-LDA on quantitative and qualitative measures on three corpora. In general, D-ETM was found to provide better predictions and topic quality than D-LDA.

3.4.4 Top2Vec

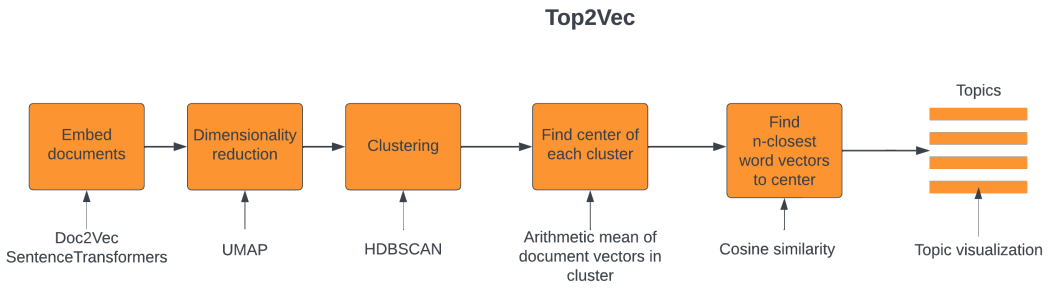


Figure 3.4: Flowchart showing the different parts in the Top2Vec architecture.

Before going into details about the Top2Vec architecture proposed in Angelov *et al.* [4], some further background about the concept of topics specifically related to their use in top2vec has to be done. As mentioned in previous sections, a topic is the theme or subject of a text. It is the underlying thing being discussed. The topic of a conversation can be thought of as a discrete value such as politics or religion, but any of these topics can be further divided into sub-topics so the topics are not discrete. A conversation discussing politics might mention healthcare as well as education and we can continue dividing healthcare and education into more specific topics. This leads us to topics being continuous as there are infinitely many combinations of sub-topics and possible divisions. Any such topic can be described by a set of words and with topic modelling we want to find the best-describing set of weighted words.

An overview of the process used in Top2Vec can be seen in Figure 3.4. The idea behind Top2Vec is to create a joint word-document embedding space so that the distance between documents and words can be found. This embedding is typically high dimensional and therefore suffers from data sparsity. Because of the high dimensionality, the distances between words converge so that similar words are almost as close to unsimiliar words. To combat these issues, dimensionality has to be reduced. Top2Vec uses UMAP which projects the embedding space into a lower dimensional space with high-density clusters of word and document vectors. Afterwards HDBSCAN is used to find clusters of words and documents

while ignoring the outliers. The idea is that these clusters of words and documents should represent topics in the space. To find the topic vector the centroid of the cluster is calculated and the words closest to the centroid are chosen as topic words as the distance in the space should represent semantic distance. Top2Vec had some limitations to which embedding model it could use, however as further discussed in Section 4.3.1, an updated version of Top2Vec can be found here: https://github.com/Lotfi-AL/Top2Vec.

Top2Vec has several advantages over LDA. LDA requires preprocessing and stopword removal which Top2Vec theoretically does not need stopword removal [4]. The reason why is that in the embedding space stopwords are not semantically near other words except for other stopwords. This gives Top2Vec the ability to produce coherent topics except for some "stoptopics". However in practice, as will be seen in the preliminary-experiment in Section 5.2, preprocessing is often required to get usable results with Top2Vec. Another advantage of Top2Vec is that it automatically finds the amount of topics in contrast to LDA where it is required as a parameter. There is no definitive approach to finding the optimal number of topics for LDA, it usually requires guesswork and a lot of trial and failure.

3.4.5 Sia *et al.*

Sia *et al.* [55], proposed a neural topic model approach that leverages pretrained word embeddings, dimensionality reduction, and clustering techniques to generate topics. The method consists of several steps. First, the vocabulary of the corpus is transformed into its embedded representation using a selected word embedding model. Then, K-means clustering is applied to the embedded vocabulary, resulting in k clusters, each corresponding to a topic. The top-100 topic words are extracted by selecting words closest to the centroid center of each cluster.

To enhance the quality of the topic words, TF-statistics are employed to rerank the top-100 words. This reranking step allows for better discrimination and selection of the most representative words for each topic. Notably, Sia *et al.* showed that without the reranking step, clustering yielded "sensible" topics, but lower Normalized Pointwise Mutual Information (NPMI) scores. The number of words to include in a topic can be adjusted by setting a cutoff value, typically around 10 or 20 words.

The neural topic model presented by Sia *et al.* achieves comparable performance to Latent Dirichlet Allocation (LDA) in terms of coherence and demonstrates higher qualitative diversity. Additionally, the proposed method offers the advantage of lower runtime and complexity compared to LDA, making it a viable alternative for topic modelling tasks.

3.4.6 BERTtopic

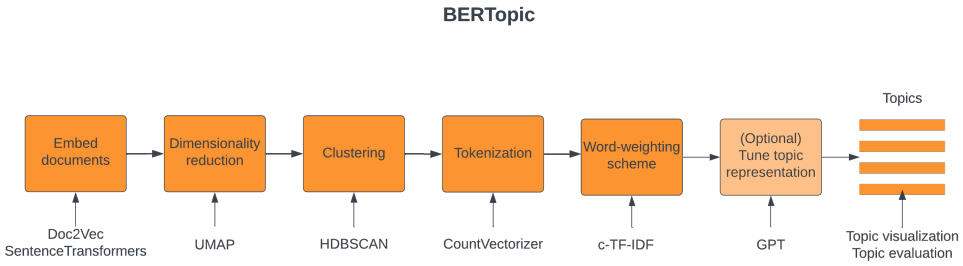


Figure 3.5: Flowchart showing the different components of the BERTopic architecture.

The BERTopic framework, introduced by Grootendorst [26], is a topic modelling approach that combines document embeddings, clustering techniques, and a variation of TF-IDF to generate topic representations. The framework offers an open architecture, allowing for flexibility in selecting methods for each step of the topic modelling process.

The code for BERTopic is available on GitHub at https://github.com/MaartenGr/BERTopic. This enables users to access the implementation of BERTopic, provide feedback and suggest changes.

Figure 3.5, provides an overview of the different components of the BERTopic architecture and illustrates examples of methods that can be used for each step. The framework offers a variety of options for document embeddings, clustering algorithms, and variations of TF-IDF, providing users with the ability to tailor the topic modelling process to their specific needs.

The document embeddings are created through a pre-trained language model such as Doc2Vec or SBERT. Similarly to the approach in Top2Vec, the embedding has to be reduced in dimensionality before clustering can occur. The dimensionality is reduced with a technique such as UMAP before a clustering technique such as HDBSCAN is used to cluster semantically similar vectors. The idea is that each cluster should represent a distinct topic. The difference from Top2Vec lies in that after clustering, Top2Vec uses a centroid-based approach to find the topic words while BERTopic incorporates a class-based variant of TF-IDF (c-TF-IDF) to find the topic words. The process to use the c-TF-IDF is to concatenate all documents in each cluster and treat them as a single document. The first term in c-TF-IDF is then the term frequency in the cluster which represents a topic. The inverse document frequency part is then replaced by the inverse class frequency to measure

how much information a term provides to a class (topic). A term corresponds to a word.

BERTopic can also be used for dynamic topic modelling with a small modification. The idea behind it is that global topics will contain the same words no matter if the data has a time aspect to it. Topics about education will have schools and students in them, no matter the temporal aspect. Therefore the process for dynamic topic modelling with BERTopic starts with fitting BERTopic on the data to get the global representation of topics. Afterwards, we want to find the local representations of topics specific to the time frame. Because we do not completely rely on the clustering but also apply c-TF-IDF to find the topic words we can change this latter part of the process. To create the local representation of each topic we multiply the term frequency in the cluster at the timestep we are interested in with the pre-calculated global IDF values. Because we do not need to embed and cluster documents again to find the local representations this is an efficient method for dynamic topic modelling.

BERTopic serves as a viable architecture for topic modelling. The ability to utilize different methods at each step is highly valued and especially the embedding model choice is useful, considering that new large language models are continuously trained and published. Another positive is that the creator: Maarten Grootendorst is active on platforms such as Twitter and Github, incorporating feedback from the users and continuously launching improvements.

3.4.7 TopClus

In the paper "Topic Discovery via Latent Space Clustering of Pretrained Language Model Representations", Meng *et al.* [46], a novel method called TopClus is proposed. The neural topic modelling methods discussed until now of Top2Vec and BERTopic have had a couple of steps in common. First using dimensionality reduction techqniues, then clustering techniques to find clusters.

In the case of BERT which is the pre-trained language model used for BERTopic, Meng *et al.* created a theorem that reveals that the optimal number of clusters is the length of the vocabulary it is trained on. If the theorem is correct, the method presented in Sie et al. 2020 [55] is a poor one, because it applies clustering directly to the embedding produced from BERT, which should in theory have the length of vocabulary amount of clusters, but the k topics used is going to be much lower. BERTopic gets around this limitation through clustering document embeddings and then using TF-IDF metrics to extract representative terms.

Meng *et al.* presents a new method: TopClus, which leverages the pre-trained language model embeddings by projecting the original embedding space onto a

latent space Z with k soft clusters of words corresponding to K latent topics. The latent space Z, is assumed to be spherical and lower dimensional.

TopClus is an interesting method and makes some good points that could potentially lead to some advancements. However, for the purpose of our thesis, it is favourable to utilize a topic modelling method that has a more comprehensive framework around it. We want the freedom to customize parameters such as the choice of embedding model and various post-processing methods of BERTopic. Therefore TopClus will not be further explored.

3.5 Hoyle *et al.* - Is Automated Topic Model Evaluation Broken?

In the paper: "Is Automated Topic Model Evaluation Broken?: The Incoherence of Coherence", Hoyle *et al.* [29], goes into depth on the quantitative and qualitative evaluation metrics of topic models. There are two main issues brought up: the standardization gap and the validation gap. The standardization gap is explained as researchers being inconsistent in their papers when describing models. Model parameters are left out as well as what preprocessing steps must be done on the corpus before modelling. The result is that it can be difficult to recreate the results. Additionally, a lot of the work done in the field is based on two papers, Chang *et al.* [15], and Lau *et al.* (2014) [35]. Both papers found that different automatic measure for coherence such as PMI, NPMI and OC-Auto-PMI [50], were able to emulate human performance. However, Card *et al.* [14], showed that many NLP experiments using human evaluation are under powered. Hoyle *et al.* , makes the argument that a minimum of fifteen crowdworkers per topic for both the rating task and word intrusion tasks to obtain significance at $\alpha = 0.05$. On this criterion both Chang et al *et al.* and Lau *et al.* (2014), with eight annotators are under powered.

This leads to the next issue presented; the validation gap. These papers do not test the evaluation metrics on neural embedding based topic models such as Top2Vec and ETM. The evaluation metrics are validated on probabilistic generative topic models, which have been shown to produce different topic words than neural topic models [29]. In Hoyle *et al.* different topic models are evaluated on both human evaluation metrics and automated metrics. The results between the models however show that the human judgments differ from automated metrics. One of the differences highlighted is that automated metrics show a tendency to value esoteric(corpus specific) topic terms. Automated metrics especially overstate the differences with models where they will score one model very high and another very

low, while human evaluation score the models with a much smaller difference in score.

For the human evaluation, Hoyle *et al.* proposes an improvement to the word intrusion task with the added question of answer confidence. Answer confidence is added to be able to filter out respondents who are not familiar with the topic terms presented and cannot answer confidently. However this improvement is not enough as even after filtering out respondents who are not confident, the automated metrics still overstate the differences between models. This hints that the automated evaluation metrics also have some issues.

A large takeaway from this paper is that one of the primary uses of topic modelling is for experts to do content analysis with the help of topic models. The way that the right topic model is chosen for a specific task is often based on testing the different models against each other on a validation corpus. This method can be flawed because the validation corpora does not always represent the same content as the expert will analyze. The paper emphasizes the importance of human evaluation.

3.6 Automatic Topic Labelling

In this section we will mainly focus on two papers: Lau *et al.* presented an automatic topic labelling method in their paper: "Best Topic Word Selection for Topic Labelling." [37] in 2010, before making additional improvements in their paper "Automatic labelling of topic models" [34] in 2011. In their first paper Lau *et al.* presented a supervised method to reranking the top-10 topic terms to select topic labels. The idea is that a good label exists in the top-10 topic terms, but the best label is not always the first topic term, therefore a re-ranking is necessary. The re-ranking is done through a ranking support vector machine (rankSVM) with different association measures as features. The results showed that the rankSVM outperformed the baseline which was simply choosing the top-1 topic word without reranking.

The approach mentioned above showed promise and was further developed by Lau *et al.* [34]. In this paper, a point is made that single words are not good descriptions of topics, but good descriptions of topics are often phrases consisting of multiple words. Therefore using phrases would be preferable to single words. Also, sometimes the best-describing words are not contained in the topic terms. For example, a topic consisting ["red", "blue", "green", "yellow"], would be better labelled by "color", than "red". If we want to use phrases as topic labels or at least have the option to use them, we need to expand beyond the top-n topic terms

because phrases are not contained in them. Lau et al [34], proposes an interesting method to automatically label topics. The big assumption made is that because of the size and quality of content in the English Wikipedia, a vast majority of topics should be discussed in articles.

The first step is to produce an arbitrary topic model to generate topics from a corpus. The topics have to be somewhat coherent for the rest of the method to work. We are then interested in producing candidate labels for the topics.

The first set of candidate labels is produced by querying Wikipedia with the top-10 topic terms for each topic. The top-8 article titles for each topic constitute the primary candidate label set. The primary candidate labels are then chunk parsed using OpenNLP chunker [1] and all noun phrases are extracted. For each noun chunk we generate all component n-grams, before pruning any n-gram that is not a Wikipedia article title. The set of resulting n-grams is the secondary candidate label set. Because the secondary candidate label set often contains stopwords or words that are only marginally related to the topic some further pruning is done. We remove outliers and poor labels using RACO (Related Article Conceptual Overlap) [25]. The secondary label candidates with a RACO of 0.1 and above are added to the final candidate labels. Lastly, the top-5 topic terms are added to the final candidate label to account for the instances Wikipedia queries come up empty.

After producing the candidates we need to rank them. There are a variety of different association measures listed in Lau *et al.* [34], but these are mainly useful in the case of supervised learning. To do supervised learning we need gold standard labels which could be acquired through crowdsourcing a label ranking task to for example Amazon Mechanical Turk, or manually performing it. However, this is a tedious task and we want to avoid it if possible. The straightforward way to rank the label candidates is then to use a single one of the association measures mentioned and rank the candidate labels based on the score. An option is to use Pearson's χ^2 test to rank the candidate labels as it was found by Lau *et al.* , to outperform the other association measures.

If we have access to a word embedding through our topic model we can find the distance between the embeddings of the candidate labels and the embeddings of the topic terms. This becomes problematic if the words are not in the vocabulary. If that is the case we need another approach. A method could be to learn a word embedding on the Wikipedia corpus. Then we could average the word embedding of topic terms and find distance of this word embedding for each candidate label

[1] Link to OpenNLP chunker: `https://opennlp.apache.org/docs/1.7.1/apidocs/opennlp-tools/opennlp/tools/chunker/Chunker.html`

and rank them based on their distance. A lower distance is better. If we used Top2Vec to generate topic terms, we could simply find the distance to the centroid of the topic clusters and rank the candidate labels based on their closeness. This method however also fails if the candidate labels are not in the vocabulary.

Overall, these papers provided valuable insights into automatic topic labelling techniques and inspired the development of the method presented by [6] *et al.* , which is the method we will be using for the experiments. Further detail of this method can be found in Section 4.5.1.

3.7 Discussion

In this chapter we have by examining the existing literature and research in the field of topic modelling, addressed the first part of Research Question 1: "What topic modelling techniques exist?" Similarly, by studying the available automatic topic labelling methods, we have provided an answer to the first part of Research Question 2: "What automatic topic labelling techniques exist?"

Chapter 4

Methodology and Architecture

In this chapter, we begin by providing a detailed analysis of the datasets in Section 4.1. We then discuss the common methods that are utilized in multiple experiments. This includes the representation of topics, which is explained in Section 4.2, as well as the configuration of topic models and embedding models, with a brief introduction to them in Section 4.3. Additionally, we present the topic evaluation framework in Section 4.4, followed by the discussion of the automatic topic labelling methods to be used in Section 4.5. Finally, we describe the implementation of automatic metrics in Section 4.6. This chapter serves as an overview of the methods employed across various experiments in Chapter 5.

4.1 Dataset Analysis

In this thesis we will be focusing on two datasets: Norwegian Parliament-Mini (NPM) and Norwegian Parliament-Large (NPL). NPM is a version of a dataset that was created by NbAiLab, the norwegian-parliament dataset, and can be found on their Hugging Face site [1]. The NPL dataset was created as part of ParlaMint 2 [22], by NbAiLab and is available [2]. The norwegian-parliament dataset is a collection of text passages, that were transcribed from speeches from 1998 to 2016 at the Norwegian Parliament. The passages are annotated based on which party spoke; Fremskrittspartiet (The progressive party) and Sosialistisk Venstreparti (The socialist left party) as well as the time and date of speeches. The dataset consists of 5000 text samples split into 3600 Training samples, 1200 Validation samples and 1200 Test samples. The NPM dataset which we will be using is simply only the

[1] Link to NbAiLab Hugging Face site:`https://huggingface.co/NbAiLab`

[2] Link to NPL dataset: `https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-77/`

3600 training samples extracted. The dataset contains both Norwegian bokmål and Norwegian nynorsk. The main differentiating factor of NPM and NPL is that NPL contains text passages from 1998 - may 2022. Additionally NPL includes speeches from all political parties.

To extract NPM and NPL some data processing had to be done to convert the original datasets into an easier to use file format in .csv. The process can be seen in Figure 4.1. NPM was already provided in an easy to use format, but NPL required quite a lot of processing to get to a .csv. The extended data processing of NPL is further deatiled in Appendix A.

For NLP tasks further preprocessing is often needed. Based on the rules shown in Table 4.3, three versions of NPM and NPL were produced: NPM/NPL-raw, NPM/NPL-basic and NPM/NPL-stopwords. The raw datasets can be mostly ignored because they not different from the basic datasets in any meaningful way as further discussed in Section 5.2. We will therefore be focusing on the basic and stopwords datasets.

NP-basic has special characters removed as well as punctuation removed. NP-stopwords has in addition to the NP-basic rules, also token normalization i.e. all strings are lowercase, and stopword removal. The stopword removal was performed using a stopword list compiled from multiple sources [3].

If we take a closer look at the dataset statistics proposed in Churchill and Singh [17], of NPM-basic and NPM-stopwords. We can see that NPM-basic has a quite large vocabulary of 47,141 and includes many tokens with the value of 1,078,981. The documents are quite long as well with the average tokens per document being: 299.72. We can also see that a large part of those tokens are stopwords at 197 average stopwords per document. For NPM-stopwords the total tokens have been reduced drastically to 369,478, more than halved, as an effect of there being many stopwords per document. The average tokens per document is much lower at: 102.6, but the vocabulary is only reduced by the length of the stopword list. For the NPL versions, most of the same remains true, however there is an anamoly with the vocabulary size decreasing quite a lot when going from basic to stopwords. The main difference between NPM and NPL is that the total tokens is drastically bigger in the magnitude of almost 100. The vocabulary is also more than 10 times larger than the NPM vocabulary sizes. The vocabulary increasing this much is not totally expected and could potentially signify some issues with the dataset. One such issue could be that the dataset is not properly processed. However, conducting a more extensive and thorough analysis requires significant effort and is left as

[3]Link to github repo containing the stopword list: `https://github.com/Lotfi-AL/norwegian-stopwords`

a task for future work. The vocabulary sizes also indicate that we could potentially benefit from further preprocessing in the form of NPL preprocessing, such as lemmatization or stemmatization, or statistical preprocessing through the methods of TF-IDF cleaning. Also potentially expand the stopword list even further including domain specific stopwords, however this is definitely a large amount of work and is left for future research. Some preliminary testing was done for the sake of curiosity and domain-specific stopwords showed some promise.

A possible reason for the vocabulary being large is due to the nature of political speeches in them being quite eloquent. Normal human speech is quite limited in the vocabulary, but politicians delivering pre-made speeches probably contain more sophisticated and less commonly used vocabulary.

Because there are only two party speeches included, the topics discovered will be a bit more party specific and not show the full political picture, but we are not looking at what kind of content the topics contain, but rather the quality of the content and therefore it is not relevant. For it to be relevant you would have to argue that these political parties have either lower-quality transcripts or higher-quality, but there is no obvious reason for that to be the case.

Statistic	**NPL-raw**	**NPL-basic**	**NPL-stopwords**
Dataset Size	386,797	386,795	386,795
Vocabulary Size	599,436	580,070	523,405
Total Tokens	94,530,866	85,544,882	31,352,260
Average Token Frequency	157.70	147.47	59.90
Average Tokens per Document	244.39	221.16	81.06
Average Stopwords per Document	131.06	131.11	0.00

Table 4.1: Dataset statistics for NPL-raw, NPL-basic, and NPL-stopwords.

Statistic	**NPM-raw**	**NPM-basic**	**NPM-stopwords**
Dataset size	3600	3600	3600
Vocabulary size	47,972	47,141	46,487
Total Tokens	1,189,592	1,078,981	369,478
Average Token Frequency	24.80	22.89	7.95
Average Tokens per Document	330.44	299.72	102.63
Average stopwords per document	196.98	197.08	0.00

Table 4.2: Dataset statistics for NPM-raw, NPM-basic, and NPM-stopwords.

Dataset	Preprocessing Rules
NP-raw	Character normalization (lower cased)
NP-basic	raw rules + Special characters removed and punctuation removed
NP-stopwords	Basic rules + stopword removal

Table 4.3: Overview of the different versions of NPM and corresponding preprocessing rules applied for Experiment 1.

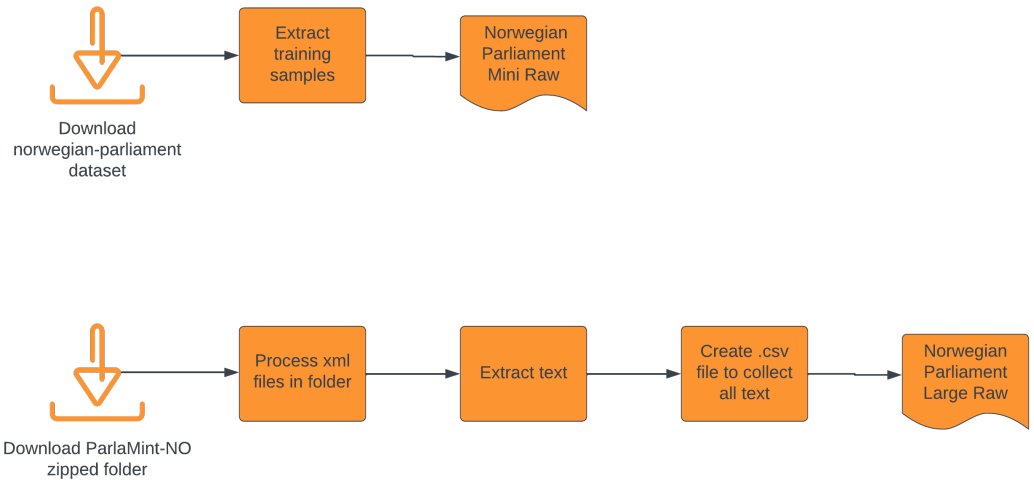


Figure 4.1: Flowcharts showing the process of creating the NPM and NPL datasets.

4.2 Wordcloud Representation

Wordclouds is an often used representation method for topics. One important feature of the wordcloud is that the size of the words in the wordcloud represent the importance of the topic words. Another feature is the number of words to include. During preliminary testing we started with 20 words, but found those wordclouds to be too chaotic and hard to understand. The wordclouds in this work will therefore be of 10 words and with the size of the word representing its importance to the topic. We will be using the "wordcloud" package for python [4], and basing our function on the one Rushfeldt [53] uses, making some modifications

[4]Link to the wordcloud package: `https://pypi.org/project/wordcloud/`

to it. When showing topic modelling results we will be representing them as wordcloud samples, each sample including eight topics. The wordclouds will sometimes have titles explaining the figure (in addition to captions). Other times the margins will be reduced and title removed to save space. The reason we only include eight wordclouds is that they fit on two lines and there are simply too many wordclouds for us to show them all in a way that they are readable even in the Appendix. It would take hundreds of pages at least.

4.3 Topic Model Configurations and Parameters

In this section, we will describe the embedding models that were tested for Top2Vec and BERTopic, as well as the various hyperparameters that were adjusted during the experiments for both Top2Vec, BERTopic, and LDA.

The embedding models along with their respective aliases can be seen in Table 4.4. In Itemize 4.3 you can see a short description of all the embedding models used, along with their parameter setup when applicable.

Embedding Model	**Alias**
all-MiniLM-L12-v2	L12
all-RoBERTa-large-v1	roberta
distiluse-base-multilingual-cased-v2	distiluse-v2
distiluse-base-multilingual-cased-v1	distiluse-v1
NbAiLab/nb-SBERT-base	nb-sbert
all-MiniLM-L6-v2	L6
TDE-NbAiLab/nb-SBERT-base	tde-nb-sbert
TWE-NbAiLab/nb-SBERT-base	twe-nb-sbert
Doc2Vec	doc2vec
universal-sentence-encoder-multilingual	universal

Table 4.4: List of embedding models used for BERTopic and Top2Vec, along with their respective aliases used when the full name is too long.

- **all-MiniLM-L6-v2** - Sentence-Transformers model. Maps sentences to a 384 dimensional dense vector space
- all-MiniLM-L12-v2 - Sentence-Transformers model. Maps sentences to a 384 dimensional dense vector space
- distiluse-base-multilingual-cased-v1 - Sentence-Transformers model. Maps sentences and paragraphs to a 512 dimensional dense vector space.

- distiluse-base-multilingual-cased-v2 - Sentence-Transformers model. Maps sentences and paragraphs to a 512 dimensional dense vector space.
- Doc2Vec - vector size = 300, window = 15, sample = 1e-5, dbow_words = 1
- NbAiLab/nb-sbert-base - nb-sbert-base is a Sentence-Transformers model. The model maps sentences and paragraphs to a 768 dimensional dense vector space.
- Transformer-Document-Embeddings-nb-sbert-base - TDE-nb-sbert-base is a Transformer Document Embedding model.
- universal-sentence-encoder-multilingual - Is a TensorFlow-text embedding. It outputs a 512 dimensional vector space.
- all-RoBERTa-large-v1 - This is a sentence-transformers model: It maps sentences and paragraphs to a 1024 dimensional dense vector space
- Transformer-Word-Embeddings-nb-sbert-base - TWE-nb-sbert-base is a Transformer Word Embedding model.

All the models used except for Doc2Vec and universal-sentence-encoder-multilingual can be found on https://huggingface.co/sentence-transformers.

4.3.1 Top2Vec

For Top2Vec the variation in embedding models available was at first, much lower due to the architecture only supporting a selected few embedding models. To make Top2Vec viable with any Sentence Transformer model and Transformer Document Embedding model some changes to the code had to be made. We forked the original Top2Vec Github project and created our own updated version. We were unable to make Top2Vec compatible with Transformer Word Embeddings (TWE), which is why the TWE-nb-sbert-base was not tested for Top2Vec.

Other input parameters of signifance is the speed parameter if we are using Doc2Vec as the embedding model. Generally want to use "deep-learn" as it is the setting with the longest training duration. Other than that depending on what type of embedding model we are using i.e Transformer Document Embedding or Sentence Transformer model, we have to set some variables to True.

4.3.2 BERTopic

BERTopic offers several useful built-in methods for visualization and postprocessing.

The *visualize_topics*() method generates an intertopic distance map, illustrating the distances between topics in a 2D space after dimensionality reduction.

This visualization provides insights into the relationships and similarities between topics. An example of an intertopic distance map can be seen in Figure 4.2.

The *visualize_heatmap*() method produces a similarity matrix, displaying the pairwise similarity between topics. This heatmap visualization helps assess the similarity levels among topics. An example of a similarity matrix can be seen in Figure 4.3.

These two visualization methods complement each other, allowing for a more comprehensive understanding of topic relationships and distances than using each method individually.

Another notable feature of BERTopic is the ability to incorporate a representation model. Users can fine-tune the topic representation by providing a text-generation model such as GPT2 or GPT-3.5 from OpenAI. By introducing a representation model, topic labels can be generated instead of relying solely on topic words.

The *reduce_outliers*() method is another interesting functionality of BERTopic. When using clustering methods like HDBSCAN, an outlier topic (topic -1) may be generated to capture documents that are identified as outliers. The *reduce_outliers*() method reduces the number of documents assigned to the outlier topic and aims to redistribute them among the remaining topics, enhancing topic coverage across the document collection. Figure 4.5 illustrates the impact of the *reduce_outliers*() method on topic distribution.

4.3.3 LDA

For LDA, there are several parameters to consider for tuning: the number of topics, the number of passes, the decay parameter, the minimum probability, and the random state.

The number of topics determines the desired number of distinct topics that LDA should identify and assign words to.

The number of passes determines the maximum number of iterations over the document collection that LDA should perform. Each pass involves updating the topic assignments and estimating the topic-word distributions. Increasing the number of passes can help improve the quality of the topic modeling.

The decay parameter controls the learning rate of LDA. It determines how much of the new knowledge gained during each iteration should be retained. Higher decay values allow the model to adapt more quickly to new information, but it may also cause the model to forget previously learned patterns.

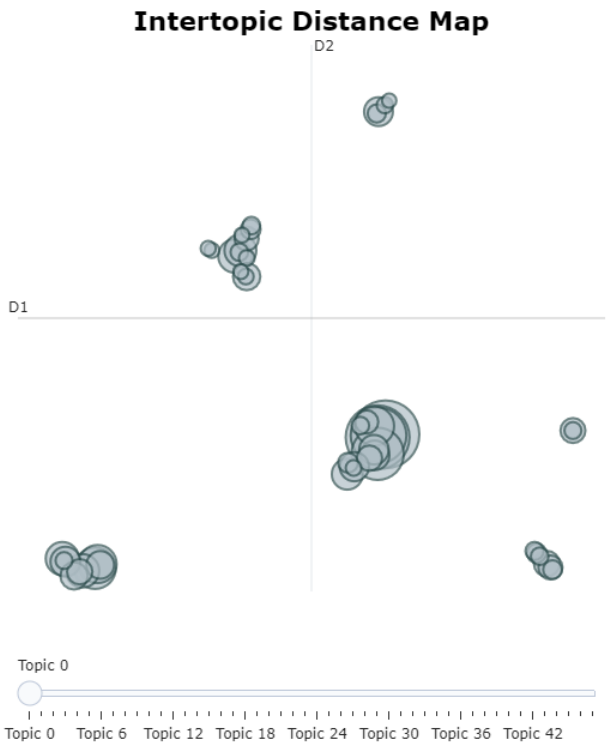


Figure 4.2: Example of a plot created by the visualize-topics method of BERTopic. The graph shows the intertopic distance. Each cluster represents a topic and one can see the distance between the clusters.

The minimum probability parameter sets a threshold for the minimum probability of a word in a topic. Words with probabilities below this threshold are not included in the topic-word distributions. Adjusting this parameter can affect the sparsity or density of the topics.

The random state parameter is used to set a specific seed for the random number generator. By setting the random state, you can ensure that the results of the LDA model are reproducible, as the algorithm involves a stochastic process.

4.4 Developing a Topic Evaluation Framework

In light of Research Question 3 (RQ3) on improving the qualitative evaluation of topic models. We were particularly interested in exploring ways to identify trends in topic model results and sought to develop a systematic approach for this purpose.

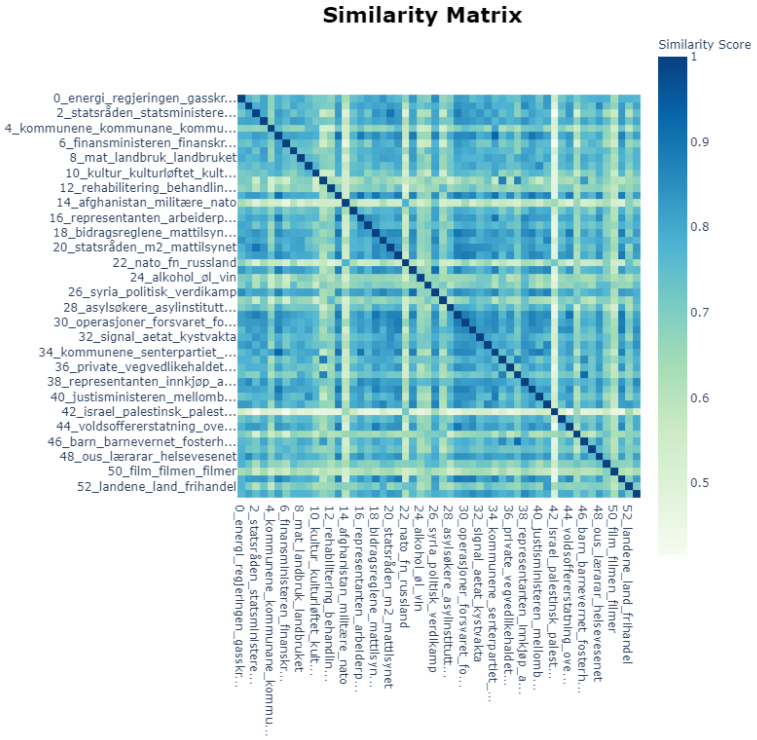


Figure 4.3: Example of a plot created by the visualize-heatmap method of BERTopic. The graph

In this section, we will provide a comprehensive explanation of the background that influenced the development of our proposed evaluation framework (TopicEval). This background information will serve as a foundation for understanding the rationale behind our framework. Subsequently, we will present the evaluation framework itself, which serves as our response to RQ3. By addressing RQ3, our aim is to enhance the qualitative evaluation process for topic models.

As can be seen in Figure 4.4, the evaluation framework is designed to be used to evaluate samples of topics represented as wordclouds.

By studying numerous wordclouds generated by various topic models on the NPM dataset, we developed a familiarity with what constitutes a good topic: what words should be included, and which ones should be avoided. This process allowed us to become something comparable to domain experts. To capture the essence of coherence and diversity, as well as incorporate semantic information such as word classes while prioritizing human interpretability, we devised an evaluation framework. The evaluation framework should have a domain expert doing the ratings.

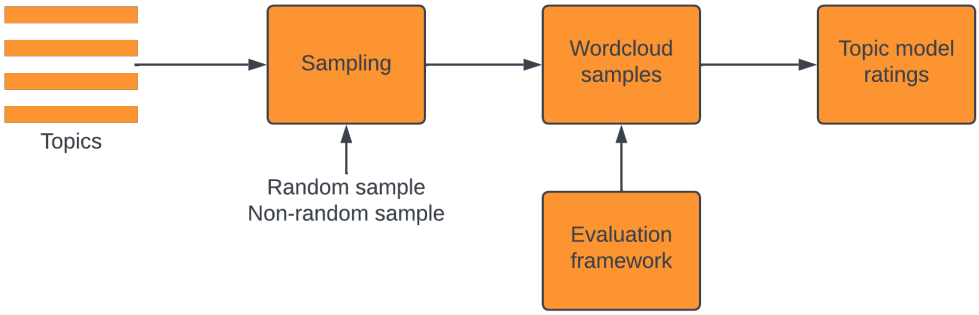


Figure 4.4: Flowchart showing the process of using the evaluation framework. Beginning with topics that are sampled. Then wordcloud samples are rated using the evaluation framework.

A domain expert can be someone who is well-versed with the dataset or has a lot of knowledge about the domain the dataset exists within.

The experiences that laid the ground for each category are as follows: **Recognizable entities** and **Related entities** are two categories devised to measure how much the topics make sense to a domain expert. We observed that topics that included many very similar words, often only differing by their conjugation or being singular/plural, would stand out negatively which is why **word repetition** was added as a category. To account for diversity, **topic similarity** was added as a category, while **depth** was included to measure the level of information conveyed by topics. Lastly, something we experienced to be truly missing from diversity and coherence was the number of topics. While it is not always beneficial to have more topics, in cases where there is a large discrepancy between the number of topics, for example, two models that produced five and 50 topics respectively. The quality of the five topics would need to be significantly better than the 50 topics for it to be the preferred one. To account for this, we introduced the category of **weighting** in our evaluation framework.

4.4.1 The Evaluation Framework

The categories of the evaluation framework will now be listed and the scoring will be explained in detail.

Word Classes

Norwegian has many different word classes with the four foremost being: Nouns, Verbs, Adjectives, and Adverbs. In addition to these major word classes, there are many smaller word classes, although frequently occurring, such as prepositions and determinatives. Most of the words in these smaller word classes are often included in stopword lists and can easily be filtered out through stopword removal. In many NLP tasks such as sentiment analysis and text generation, adjectives and adverbs are useful. However for topic modelling, what we are really interested in is nouns and in some cases verbs as well. Adjectives and adverbs provide little to no value by themselves, although sometimes providing some context to the rest of the topic words and in that way are valuable. We have included the rating category of "Word classes", in which topics get rated based on the ratio of useful word classes to word classes providing little value. In most cases, the topics will simply be rated based on how many nouns and verbs they include, but in some special cases, allowances will be made for adjectives when they provide specific context to the topic. An example of such an exception would be a topic: The scoring for "Word classes" is as follows: 1 point - none, 2 points - few, 3 points - some, 4 points - some, and 5 points- all.

- 1 point is given if none of the wordclouds are made up of exclusively nouns and verbs.
- 2 points are given if at least six out of eight wordclouds are made up of exclusively nouns and verbs.
- 3 points are given if between four and six out of eight wordclouds are made up of exclusively nouns and verbs.
- 4 points are given if between two and four out of eight wordclouds are made up of exclusively nouns and verbs.
- 5 points are given if between zero and two out of eight wordclouds are made up of exclusively nouns and verbs.

Additionally, each wordcloud itself is judged based on the amount of non-essential word classes and the number of words belonging to these word classes included in the topic. Therefore even though there is only one topic with non-essential words, if that topic has a lot of non-essential words the sample could potentially be given four points, instead of five as the list would indicate.

A way to think about the rating of word classes is, can a word be removed from a topic without the topic losing any meaning? If yes, the word should be counted negatively.

Recognizable Entities

In the case of rating topic models, we can make the assumption that there is a domain expert in charge of the rating. A domain expert is simply someone who is a bit familiar with the dataset and has an idea of what kind of topics make sense. The evaluation framework is made to judge topic models by people domain knowledge, it is not made for users of the applications. Therefore this assumption of domain knowledge is within reason. Domain knowledge or expert knowledge in this case does not mean that the person has to know everything about the dataset or the domain. It is a weaker assumption than that, and simply implies that you have a reasonable understanding of the domain and what topics are logical to find. In effect, depending on the domain, people that have a basic, general understanding will be considered sufficiently equipped. For the parliamentary data, an expert would be someone that has followed some political discussions and debates, as well as having followed discussions in the media, especially leading up to parliamentary elections. This category in essence rates the sample of wordclouds based on how many of the topics represented are recognized as a topic, meaning that the wordcloud contains itself and is somewhat meaningful. The scoring for "recognizable entities" is similar to "word classes" and is as follows: 1 point - none, 2 points - few, 3 points- some, 4 points - most, and 5 points - all.

- 1 point is given if none of the wordclouds include recognizable entities
- 2 points are given if between two and four out of eight wordclouds include recognizable entities
- 3 points are given if between four and six out of eight wordclouds include recognizable entities
- 4 points are given if between six and eight out of eight wordclouds include recognizable entities
- 5 points are given if all wordclouds include recognizable entities that are meaningful

Similarly to "word classes", each wordcloud itself is judged based on how easy it is to recognize the entities and topic represented. What follows is that a wordcloud sample can go up or down a level based on the recognizability of the individual wordclouds.

Related Entities

This category refers to how related entities are within a topic. What is meant by related is that there exists some common theme that the topic words are somewhat connected to and through that connected to each other. The reason behind having this category is to avoid topics that are actually multiple topics combined

to be highly valued. A topic that includes alcohol policies, privatization and peace negations could only have high quality topic words, but due to a lack of common theme it should be scored lower. The scoring for this category is as follows: 1 point - none, 2 points - few, 3 points- some, 4 points - most, and 5 points - all.

- 1 point is given if none of the wordclouds include related entities
- 2 points are given if between two and four out of eight wordclouds include related entities
- 3 points are given if between four and six out of eight wordclouds include related entities
- 4 points are given if between six and eight out of eight wordclouds include related entities
- 5 points are given if all wordclouds include related entities

Each wordcloud is also judged individually based on how strong the common theme is and how related the topicwords are.

Word Repetition

This category refers to many words repeat themselves inside a topic and how many topics include word repetition. When we are scoring topics and look at the top-10 topic words, it is essential that each word is unique for the topic to carry as much meaning as possible. If we imagine the ideal topic, it would include 10 unique topic words. It is not only exact word repetition that counts, but also different conjugations of the same word and plural or singular forms as well. The scoring for this category is as follows: 1 point - all, 2 points - most, 3 points - some, 4 points - few, 5 points - none

- 1 point is given if all of the wordclouds include word repetition
- 2 points are given if between six and eight out of eight wordclouds include word repetition
- 3 points are given if between four and six out of eight wordclouds include word repetition
- 4 points are given if between two and four out of eight wordclouds include word repetition
- 5 points are given if none of the wordclouds include word repetition

Topic Similarity

This category measures how much the different topics in a sample vary and how wide the area covered is. What often happens with some topic models is that a fair amount of topics are produced, but it is essentially the same topic repeated.

The scoring for this category is as follows: 1 point - all, 2 points - most, 3 points- some, 4 points - few, and 5 points - none.

- 1 point is given if all of the wordclouds are similar
- 2 points are given if between six and eight out of eight wordclouds are similar
- 3 points are given if between four and six out of eight wordclouds are similar
- 4 points are given if between two and four out of eight wordclouds are similar
- 5 points are given if all the wordclouds are similar

Additionally, it has to be considered how similar the topics are and take that into account when scoring.

Depth This category measures how much detail the topics contain. We do not want too much depth either, because that would probably compromise the general quality of the topics. This category might be the one with the most subjective scoring, because the sample is rated based on sufficient depth. The sufficient depth will change based on what topics are covered. The scoring for this category is as follows: 1 point - none, 2 points - few, 3 points- some, 4 points - most, and 5 points - all.

- 1 point is given if none of the wordclouds have sufficient depth
- 2 points are given if between two and four out of eight wordclouds have sufficient depth
- 3 points are given if between four and six out of eight wordclouds have sufficient depth
- 4 points are given if between six and eight out of eight wordclouds have sufficient depth
- 5 points are given if all wordclouds have sufficient depth

Weighting

The last part of the evaluation framework is to include a weighting mechanism. Oftentimes when comparing different embedding models two models might get the same score, but if one model has twice the number of topics produced, it is obvious that model should be preferred. An option is to add an additional category "Number of topics" or something similarily named, with a score of 1-5 based on how many topics the model produced. The first issue is that this category would go outside of the scope in which we are rating; a sample of eight wordclouds. This is a not too big of an issue and could be accepted. The larger issue is in how we could make the rating of this category generalizable. The optimal number of topics differs based on the dataset, and can go from 50 to 5000. One idea was to rate the embedding models in comparison to each other. The problem with this approach is

that models that produce low number of topics would not be sufficiently penalized, because the maximum difference would be four points. Another idea was to increase the ceiling for the scoring of this category, but then again that could potentially overpower the score of the other categories and make the number of topics the only deciding factor. The approach that was settled on was to include a weighting mechanism where we would take the equation listed in 4.1. It is a simple weighting where we add 1 to avoid decreasing the score for number of topics under 100. If the number of topics is 50, the weighting would be 1.5 for example.

$$\text{Weighting} = 1 + \frac{\text{number of topics}}{100} \tag{4.1}$$

Category	**1**	**2**	**3**	**4**	**5**
Word classes	None	Few	Some	Most	All
Recognizable entities	None	Few	Some	Most	All
Related entities	None	Few	Some	Most	All
Word repetition	All	Most	Some	Few	None
Topic similarity	All	Most	Some	Few	None
Depth	None	Few	Some	Most	All
Weighting	$1 + \frac{\textbf{number of topics}}{100}$				

Table 4.5: Qualitative Measures

4.4.2 Why Random Sampling?

The logic behind random sampling to evaluate topic models stems from the observation that the first topics of a model will be assigned a large number of documents, while the last topics will be assigned close to the minimum limit of documents. An example of this can be seen in Figure 4.5 The idea is then that because the first topics are assigned many documents, they will be more general topics and therefore uninteresting from a naive human point of view. The later topics were observed to often be more intriguing and hence the idea of random sampling.

	Topic	Count	Name
0	-1	1314	1_norge_regjeringen_representanten_fremskrit...
1	0	290	0_energi_regjeringen_gasskraftverk_norge
2	1	235	1_skolen_skole_elever_utdanning
3	2	178	_statsråden_statsministeren_stortinget_spørs...
4	3	166	3_statsråden_transportplan_kr_nasjonal
5	4	151	4_kommunene_kommunane_kommuner_kommune
43	42	15	42_israel_palestinsk_palestinske_hamas
44	43	14	3_presidenten_president_spanskekongen_felles...
45	44	14	4_voldsoffererstatning_overgriper_permitteri...
46	45	14	45_irak_burma_irakiske_krig
47	46	13	46_barn_barnevernet_fosterhjem_full
48	47	13	47_renten_bank_renta_lav
49	48	13	48_ous_lærarar_helsevesenet_fagmiljøer

Figure 4.5: Illustrating the difference in documents assigned from early topics to late topics.

4.5 Automatic Topic Labelling

In this section we will describe two automatic topic labelling methods and their implementations.

4.5.1 NETL

The original NETL implementation was written in Python 2 and used outdated packages. We, therefore, had to rewrite a lot of the code. The implementation we will be using can be found on Github.

Figure 4.6 shows an overview of the NETL method. The connections between different files and their purpose are shown. A more detailed description of NETL will now ensue.

Bhatia *et al.* [6], presented a process to automatically label topics through topic label generation based on English Wikipedia and topic label ranking based on a supervised learn-to-rank model that is named NETL.

Lau *et al.* [34] ,uses an information retrieval approach to querying Wikipedia and generating candidate labels based on the queries. This approach is difficult to perform in practice because the APIs used are hard to get access to and use.

There are three steps to NETL. First is to train the necessary embedding models. The second step is to generate candidate labels. The third step is to generate topic labels based on the candidate labels.

Before step one, some preliminary steps have to be done. A topic model has to find topics in the training corpus and the top-N topic terms. A Wikipedia corpus of the lanuage in use, is to be used as a basis. The Wikipedia articles has to be cleaned a little bit and the text extracted. Articles with less than 40 words are filtered out as well as articles with titles that are longer than four words. Afterwards, step one can start. A Doc2Vec model is trained on the cleaned corpus. Afterwards a Word2Vec model is trained on the cleaned corpus as well. The topic is represented by the top-N topic words word embeddings.

Next part is to generate candidate labels using Doc2Vec and Word2Vec. We make the same kind of assumption as in Lau *et al.* [34], that Wikipedia article titles are good candidate labels. We represent the article titles in Doc2Vec as the document embedding. And for Word2Vec we treat article titles as a single token by concatenating them and then representing them through their word embedding. Now to get an inital candidate label set we need to measure how relevant each article title is to the topic terms. Two candidate label sets are generated one for Doc2Vec the other for Word2Vec. To measure the relevance, pairwise cosine similarity between the each article title embedding and each of the word embeddings for the top-10 topic terms. We then aggregate the values through the arithmetic mean. This method produces two collections of article titles ranked by their relevance to the topic terms. Bhatia *et al.* [6], experimented with combining the strengths of Doc2Vec and Word2Vec and found it to be the best method. The combining of strengths is done through summing the relevance scores of the top-100 candidates and using the new ranking.

These candidate labels are ranked and could be used as they are by simply using the highest ranked candidate label, but in an effort to find a better label a support vector regression (SVR) model is trained to rank labels. The NETL method has the option of doing this process supervised, but in our case we are interested in the unsupervised version. For the unsupervised version the generation of topic labels from candidate labels is simply based on their LetterTrigram rank which is based on measuring the overlap of letter trigrams between a topic label and the topic words. This feature is used because of the findings in Kou et al. 2015 [32] that letter trigram vectors is an effective method to reranking topic labels.

The end result is that the candidate labels are ranked and based on how many topic labels we want, the cut-off point is chosen. By default, the method selects the top three candidate labels as topic labels.

4.5.2 BERTopic Automatic Topic Labelling Method

Following the release of ChatGPT in November 2022 and the general development of text generation networks, Our hypothesis was that text generation networks could do well at tasks of generating topic labels. As mentioned previously, BERTopic provides a method to fine-tune the topic model with a representation model that can be set up to prompt GPT-3.5 API. The prompt would look something like this: "I have topic that contains the following documents: [DOCUMENTS] The topic is described by the following keywords: [KEYWORDS]

Based on the information above, extract a short topic label in the following format: topic: <topic label> respond in Norwegian"

Originally this is the method we sought to use, but due to rate limitations on the GPT-3.5 API and other issues with using text-generation methods in this format. We had to find another approach.

The new approach was to find the most representative document for a topic and manually prompt ChatGPT with the following prompt: BERTopic: We have used topic modelling to generate topics. We now want to generate topic labels for these topics. A topic label is a descriptive label that best captures the meaning of the topic.

The topic is described by the following topicwords in order of importance: <TOPICWORDS>. The most representative document for the topic is <DOCUMENT>.

Based on the information above, extract a short topic label of about three words in Norwegian. Additionally extract a one word topic label in Norwegian.

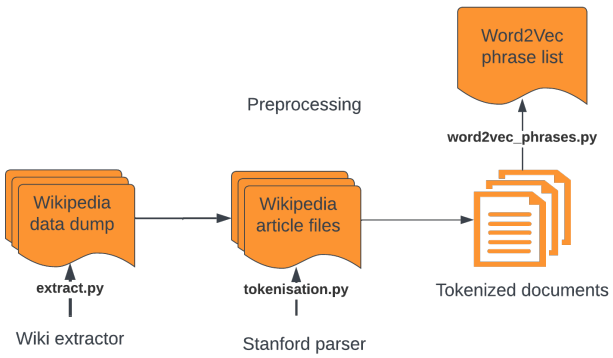
The NETL method and the above described BERTopic method is the ones we will be using for the experiment 6 in Section 5.7, as well as for the user testing experiment in Section 5.4.

4.6 Implementation of Automatic Metrics

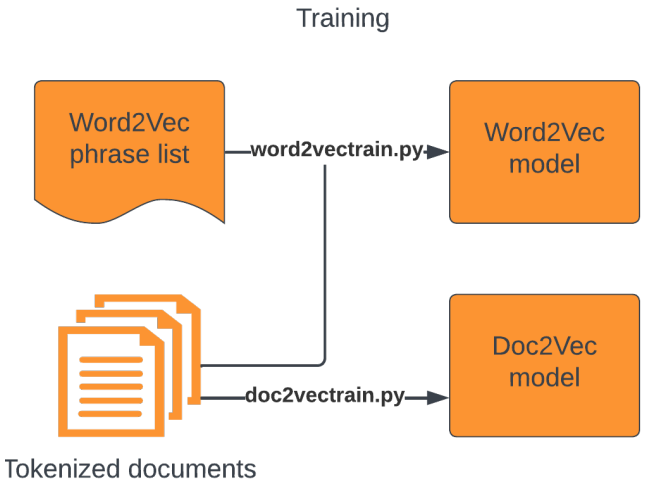
For the automatic metrics to evaluate topic models we used NPMI for coherence and topic diversity as described in Section 2.3.1. For the implementations, we used the TextPrep toolkit [17] implementation of coherence and diversity, the code of which can be found on Github. The NPMI implementation had a bug which made it not correctly normalize, meaning that the values could go outside of the specified ranges of -1 and 1. The failure of the normalization was found during testing and it seemed like the normalization failed in cases where there were few topics and the topics contained infrequently occurring words. Efforts were done to fix the issue,

but they were unsuccessful. What this means for the result is that we can not take into account the floor and ceiling of the values, which should have been -1 and 1 when thinking about how good the results are. Instead, the NPMI values should be compared to each other because they all contain the error and it should even itself out.

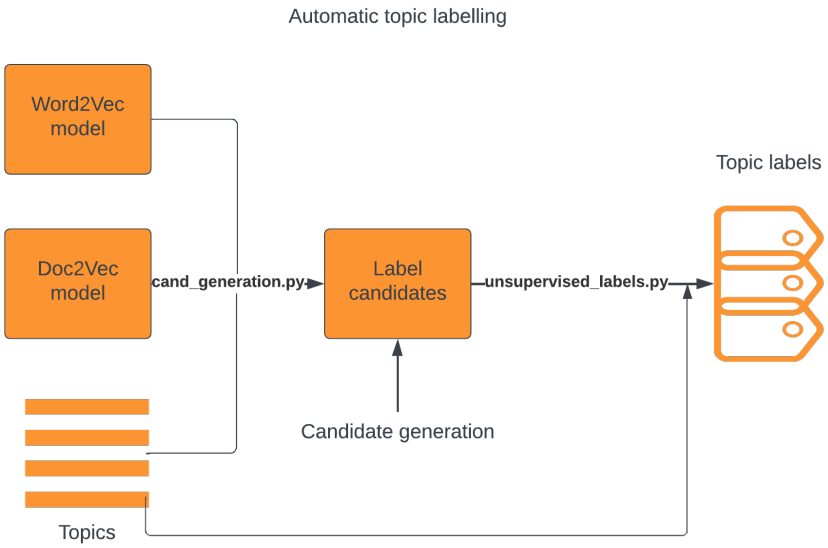
For topic diversity, we used the TextPrep implementation that can be found on Github. There were no noticeable issues with this implementation.



(a) The preprocessing steps necessary for NETL.



(b) The training phase of NETL, where we use data produced in the preprocessing step.



(c) Automatic Topic Labelling step of NETL.

Figure 4.6: Figure showing the different parts of NETL architecture, including preprocessing data, training the models and generating labels.

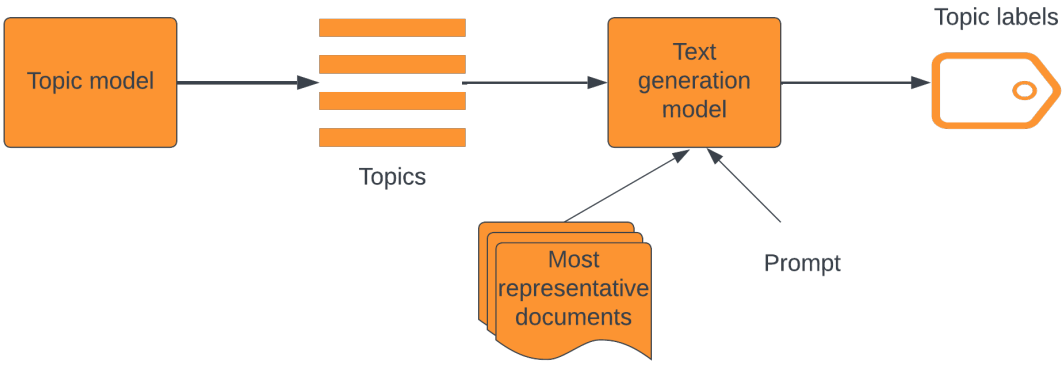


Figure 4.7: BERTopic automatic topic labelling method illustrated.

Chapter 5

Experiments and Results

In this chapter the various experiments conducted and their results will be presented. Section 5.1 lays out the experiments planned, shortly describing each of the experiments and their connection with each other. The ensuing six sections provide a comprehensive explanation of the experiments: 5.1, 5.3, 4.4, 5.4, 5.6, 5.7. The full implementation of the code for the experiments can be found in a Github repository [1]. All the wordclouds are not always shown, but they can all be found in a Github repository [2].

5.1 Experimental Plan

Experiments numbered:

1. Preliminary experiment - Tested different preprocessing rules for the NPM dataset. BERTopic, Top2Vec, and LDA were evaluated with default configurations.
2. Embedding experiment - Compared various embedding models for Top2Vec and BERTopic, along with different variations of LDA. Identified the best-performing embedding model and topic model. Discussed the results within the evaluation framework.
3. User testing experiment - Conducted topic intrusion, word intrusion, word cloud connection to document, automatic topic label rating, and topic representation tasks.
4. Evaluation framework experiment - Performed random sampling and rated samples using the embedding model setups from the previous experiment.

[1] Experiments code: `https://github.com/Lotfi-AL/Topic-Modelling-Experiments`
[2] Wordclouds: `https://github.com/Lotfi-AL/Master-Thesis-Wordclouds`

Discussed the results and the framework.

5. Norwegian Parliament-Large experiment - Applied the best-performing embedding model from the previous experiment to the large NPL textual dataset. Explored methods like outlier reduction and dynamic topic modeling within BERTopic. Rated the results using the evaluation framework.
6. Automatic Topic Labeling experiment - Compared NETL and BERTopic-based methods for automatic topic labeling. Evaluated the results qualitatively and related them to the user testing experiment.

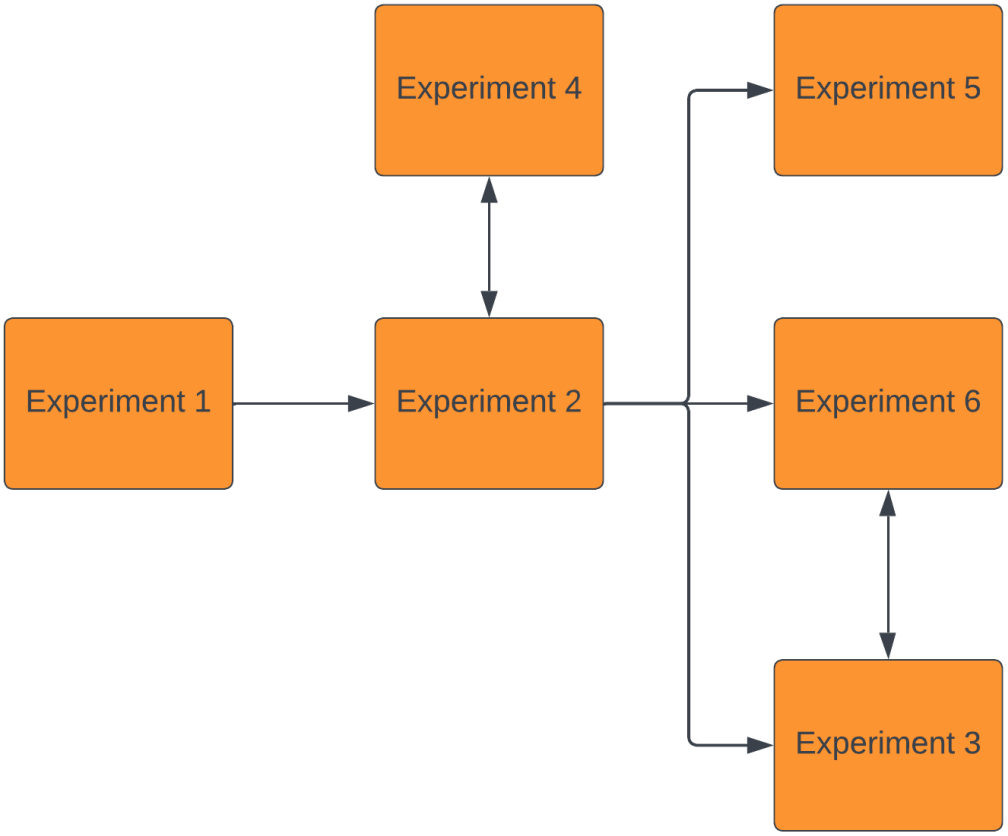


Figure 5.1: Overview of the relations between the different experiments. Experiment 2 uses the results from Experiment 1. Experiment 2 and Experiment 4 run in parallel. Experiment 3 uses the results from both Experiment 2 and Experiment 6.

5.2 Experiment 1 - Preliminary Experiment

The aim of this experiment was to assess the expected outcomes of various topic modelling methods applied to the NPM dataset. The study focused on two main aspects: the impact of different preprocessing rules and the differences among three tested topic modelling methods, namely Top2Vec, LDA, and BERTopic. Three versions of the NPM dataset were tested: NPM-unprocessed, NPM-basic, and NPM-stopwords, as summarized in Table 4.3. NPM-unprocessed refers to the raw NPM dataset without any preprocessing. NPM-basic involved removing special characters, including punctuation, while NPM-stopwords extended this by additionally removing stopwords.

All three topic models were applied to the three NPM dataset versions. Based on the models' performance, a decision was made regarding which dataset version would be utilized for subsequent experiments. Performance evaluation involved considering coherence and diversity scores, as well as a qualitative analysis to assess the quality of the results obtained.

5.2.1 Experimental Setup

For BERTopic and Top2Vec the default parameters were used. For Top2Vec that entails using a Doc2Vec model that is trained on the dataset for 40 epochs. For BERTopic the default embedding is the sentence transformer: all-MiniLM-L6-v2. For LDA the number of topics to generate had to be chosen. Ten topics were chosen and to ensure easy reproduction of results a $random\_state = 42$ was set.

For the automatic metrics used, PMI is used for coherence due to the issues mentioned in Section 4.6, of the NPMI not being properly normalized. In this experiment, we use the metrics to measure the performance of the models from an outside perspective, which means that we need to know the baseline of what constitutes a good model and a bad model. For diversity the standard topic diversity measure is used.

Because the results of NPM-basic and NPM-raw were almost identical for all the topic models, the NPM-raw results were omitted, but can be found in the Github repository

add all the results in appendix and add proper reference

	BERTopic			Top2Vec			LDA		
	Raw	**Basic**	**Stopwords**	**Raw**	**Basic**	**Stopwords**	**Raw**	**Basic**	**Stopwords**
Coherence	-0.02	-0.02	-0.10	0.09	0.08	-0.10	-0.06	-0.06	-0.05
Diversity	0.25	0.26	0.78	1.0	1.0	0.90	0.14	0.14	0.22

Table 5.1: Preliminary experiment coherence and diversity values for BERTopic, Top2Vec and LDA for NPM-raw, NPM-basic and NPM-stopwords.

5.2.2 Results - LDA

Figure 5.2: Eight-topic sample from a total of 10 topics produced by LDA run on NPM-basic for the preliminary experiment.

Figure 5.3: Eight-topic sample from total of 10 topics produced by LDA run on NPM-stopwords for the preliminary experiment.

5.2.3 Results - BERTopic

Figure 5.4: Eight-topic sample from a total of 31 topics produced by BERTopic run on NPM-basic for the preliminary experiment.

Figure 5.5: Eight-topic sample from a total of 49 topics produced by BERTopic run on NPM-stopwords for the preliminary experiment.

5.2.4 Results - Top2Vec

Figure 5.6: The two topics produced by Top2Vec on NPM-basic for the preliminary experiment.

Figure 5.8: Wordcloud representation of topic number 22 from BERTopic run on NPM-basic.

Figure 5.7: Eight-topic sample from a total of 26 topics produced by Top2Vec run on NPM-stopwords for the preliminary experiment

5.2.5 Discussion

When comparing the automatic metrics with the observations from the wordcloud samples, it is important to note that the automatic metrics are calculated based on all the topics, while the wordcloud samples only include a maximum of eight

topics. Hence some of the observed discrepancies might be explained by the fact that we are only looking at a sample, however there are no perfect ways to visualize large numbers of wordclouds. Every method will have some drawbacks that are important to consider when interpreting the results.

LDA

The results of LDA on NPM-basic can found in Figure 5.2. Upon examination, we found the topics to be incoherent and of limited usefulness, because the topics are filled with stopwords and are practically identical to each other. A sample from the stopwords model can be seen in Figure: 5.3. The stopwords model clearly performs better than the basic model, but the quality is nonetheless subpar because as for the basic model, most of the topics are similar and lack significant semantic meaning.

The diversity and coherence scores for all the LDA models can be seen in Table 5.1. The basic model got a coherence score of: -0.06 and diversity of 0.14, which are quite low values that clearly correspond to what can be seen in the wordcloud sample. The stopwords model got a coherence score of -0.05 and diversity of 0.22, which are very close to the values for the basic model. While the stopwords model may appear to be superior based on the wordcloud samples, it is important to note that the repetition of what we might consider as more meaningful words such as "norge" (Norway), "gjelder" (applies to), "regjeringen" (the government), and "statsråden" (the minister), compared to the words in NPM-basic: "er" (is), "det" (that), "til" (to), may contribute to a subjective impression of greater meaning. However, upon closer inspection of the stopwords sample, it becomes evident that the topics themselves are not substantially more meaningful, and the included words may be domain-specific stopwords.

BERTopic

The results of BERTopic on NPM-basic can be found in Figure 5.4. The topics that can be seen in the sample are incoherent and filled with stopwords. However, the model produced 31 topics and some of the later topics are of decent quality, one such can be seen in Figure 5.8. The figure shows topic number 22, which includes some meaningful words such as "spanskekongen" and "voteringen". Altogether, the topics do not give any insight into the documents in the dataset and are therefore of limited use. A sample from the stopwords model can be seen in Figure 5.5. The stopwords model clearly outperforms the basic model and produces some coherent, useful topics.

The basic model got a coherence score of -0.02 and diversity of 0.26, which are quite

low values that do not correspond to what can clearly be seen in the wordcloud sample. The stopwords model got a coherence score of -0.10 and diversity of 0.78, which do not correspond to what can be seen in the wordcloud sample.

Top2Vec

A sample from the results of Top2Vec on NPM-basic can be seen in Figure 5.6. The sample includes the two only topics generated by the model. The topics are clearly of poor quality and provide little value. On the other hand, the stopwords model produces 26 topics, most of which are high-quality topics that fittingly cover parliamentary discussions such as Topic#2 which covers education and teachers, and Topic#7 which covers climate gas emissions, which are both frequent topics of political discussion in Norway.

The basic model got a coherence score of 0.08 and diversity of 1.0. The topics are completely diverse because there are only two of them so this value does not carry any meaning. The stopwords model got a coherence score of -0.10 and diversity of: 0.90. The diversity score is very high considering that the model produced 26 topics, meaning that they cover a wide area of different topics most likely. The coherence score of -0.10 is lower than the basic models score of 0.08 which is surprising considering that the basic model topics do not carry human meaning.

Comparison of the Results

Overall none of the topic models managed to produce any usable results using NPM-basic, which is why NPM-stopwords will be used for the rest of the experiments. Neither the basic nor stopwords version of LDA produced any good results. Top2Vec and BERTopic were clearly the better models. Interestingly Top2Vec only produced two topics that were of poor quality for the basic model, in comparison to the 26 higher-quality topics for the stopwords model. This result is inconsistent with the findings of the original Top2Vec paper [4], where it was found that: *"Common words appear in most documents and, as such, they are often in a region of the semantic space that is equally distant from all documents. As a result, the words closest to a topic vector will rarely be stop-words, which has been confirmed in our experiments. Therefore there is no need for stop-word removal."*. The reason why the basic model struggled might be because the documents in NPM contain a great number of tokens and relatively many average stopwords per document of 197 compared to average tokens per document of 300 as can be seen in Table 4.2. The sheer amount of stopwords could be overpowering the other content and make it too difficult for Top2Vec to find meaningful topics. As can be seen in Table: 4.2, stopword removal more than halves the number of tokens in the dataset as well as the maximum amount of tokens in a document which strengthens

the hypothesis.

The results of the basic versions of LDA, Top2Vec, and BERTopic showed minor differences. When examining the topics in the samples, BERTopic and LDA appeared to be almost identical. However, a closer look at the wordclouds generated by BERTopic revealed some coherent topics, which were the only consistent ones across all the basic models. For the stopwords models, Top2Vec and BERTopic produced similar results, while LDA stood out as a negative outlier.

When considering the automatic metrics, the coherence scores ranged around 0 with slight variations of +-0.1. For diversity the scores varied more, with the best model being Top2Vec-basic with a perfect score of 1.0, and the worst model being LDA-basic with a score of 0.14. The two qualitatively best models: BERTopic-stopwords and Top2Vec-stopwords achieved the same coherence score of -0.10 and a diversity of 0.78 and 0.90 respectively. As previously discussed, the models both produced topics of decent quality, and while the diversity score reflects this, the coherence score is quite low, in fact lower than Top2Vec-basic which got a score of 0.08, whilst only producing two topics, meaning that in our limited trial, coherence did not correlate with human interpretability. As discussed in Section 5.2.5, we are only looking at a sample of the topics, but the coherence scores are still lower than what we would intuitively expect. This observation was important to keep in mind while conducting the embedding experiment.

5.3 Experiment 2 - Embedding Experiment

The primary objective of this experiment was twofold. Firstly, it aimed to generate a comprehensive set of topic model results by exploring different combinations of embedding and topic models as well as variations of LDA. These results were crucial for facilitating the subsequent experiments, either through the utilization of the results themselves, or through the deeper understanding of topic models built up. Secondly, the experiment sought to compare the performance of various embedding and topic models. The comparison was done to determine the best-performing topic model variation that would then be used for experiments 3,5 and 6.

To compare the performance of the different topic model variations, we focused on three different methods of evaluation. The results of the evaluations were then compared. First, we looked at the models in a subjective way to see if we could spot any patterns or differences in how they worked. Then, we used the automatic metrics of coherence and diversity to objectively evaluate the quality of the models. Lastly, a subjective analysis was conducted using the evaluation frame-

work presented in Section 4.4. It is worth noting that there might be some overlap between the findings of this experiment and Experiment 4.

Given the wide range of topic models tested, we will showcase three variations by presenting a sample of the first eight topics as wordclouds. The remaining wordclouds are available in the Appendix B and can also be accessed through the accompanying Github repository.

5.3.1 Experimental Setup

The parameter setup for LDA can be found in Table 5.2. For Top2Vec, $speed =$ "$deep - learn$" was used, as well as specifying which type of embedding model was used. The specific embedding models used in the experiment for Top2Vec are listed in Table 5.3. The only input parameter altered for BERTopic was the choice of embedding model. The chosen embedding models for BERTopic are listed in Table 5.4.

For the coherence metric in this experiment, we used NPMI, even though it is improperly normalized, we are mostly using the metric to compare the embedding models to each other and therefore the error will be existent in all the results and the values are therefore comparable. For diversity, we used the standard topic diversity.

LDA Parameter Setup	
passes	**num_topics**
1000	20
1000	30
1000	40

Table 5.2: LDA variations tested in Experiment 2

Top2Vec Embedding Models Tested	
Embedding Model	**Alias**
all-MiniLM-L12-v2	L12
distiluse-base-multilingual-cased-v1	distiluse-v1
distiluse-base-multilingual-cased-v2	distiluse-v2
Doc2Vec	doc2vec
NbAiLab/nb-SBERT-base	nb-sbert
TDE-NbAiLab/nb-SBERT-base	tde-nb-sbert
universal-sentence-encoder-multilingual	universal

Table 5.3: List of embedding models used for Top2Vec in Experiment 2, along with their respective aliases used when the full name is too long.

Embedding Model	**Alias**
all-MiniLM-L12-v2	L12
all-RoBERTa-large-v1	roberta
distiluse-base-multilingual-cased-v2	distiluse-v2
NbAiLab/nb-SBERT-base	nb-sbert
all-MiniLM-L6-v2	L6
TDE-NbAiLab/nb-SBERT-base	tde-nb-sbert
TWE-NbAiLab/nb-SBERT-base	twe-nb-sbert

Table 5.4: List of embedding models used for BERTopic in Experiment 2, along with their respective aliases used when the full name is too long.

5.3.2 Results - LDA

Figure 5.9: Eight-topic sample from a total of 20 topics produced by LDA for the embedding experiment.

Figure 5.10: Eight-topic sample from a total of 30 topics produced by LDA for the embedding experiment.

Figure 5.11: Eight-topic sample from a total of 40 topics produced by LDA for the embedding experiment.

5.3.3 Results - BERTopic

Figure 5.12: Eight-topic sample from a total of 40 topics produced by BERTopic using distiluse-base-multilingual-cased-v2

Figure 5.13: Eight-topic sample from a total of 46 topics produced by BERTopic using TWE-nb-sbert-base

Figure 5.14: Eight-topic sample from a total of 54 topics produced by BERTopic using all-roberta-large-v1

5.3.4 Results - Top2Vec

Figure 5.15: Eight-topic sample from a total of 20 topics produced by Top2Vec using distiluse-base-multilingual-cased-v2

Figure 5.16: Eight-topic sample from a total of 22 topics produced by Top2Vec using Doc2Vec

Figure 5.17: Eight-topic sample from a total of 23 topics produced by Top2Vec using nb-sbert-base

5.3.5 Automatic Metrics Results

Table 5.5: LDA automatic metrics and number of topics. Coherence is shown as the NPMI value and diversity as topic diversity.

Model	Coherence	Diversity	Topics
p1000-t20	1.09	0.66	20
p1000-t30	0.83	0.72	30
p1000-t40	**1.56**	**0.79**	**40**

Table 5.6: BERTopic automatic metrics and number of topics. Coherence is shown as the NPMI value and diversity as topic diversity.

Model	Coherence	Diversity	Topics
all-miniLM-L12-v2	1.36	0.77	49
all-miniLM-L6-v2	0.74	0.77	48
nb-sbert-base	1.32	**0.88**	43
distiluse-base-multilingual-cased-v2	1.46	0.82	40
TDE-nb-sbert-base	1.12	0.75	1
TWE-nb-sbert-base	**1.63**	0.82	44
all-roberta-large-v1	1.31	0.80	**54**

Table 5.7: Top2Vec automatic metrics and number of topics. Coherence is shown as the NPMI value and diversity as topic diversity.

Model	Coherence	Diversity	Topics
all-miniLM-L12-v2	1.79	0.32	**29**
distiluse-base-multilingual-cased-v2	2.64	0.58	25
distiluse-base-multilingual-cased-v1	2.76	0.61	23
Doc2Vec	1.94	**0.92**	22
nb-sbert-base	2.74	0.63	23
TDE-nb-sbert-base	**2.91**	**1**	2
universal-sentence-encoder-multilingual	2.87	0.65	2

5.3.6 Discussion

In this experiment, we have tested three different topic modelling architectures, BERTopic, Top2Vec and LDA as well as a variety of embedding models. Through the testing of these variations, this experiment has helped answer the first part of RQ1: **What topic modelling techniques exist?**. We have gathered a variety of results that we will now discuss the quality of, hence answering the second part of RQ1: **how do they perform on Norwegian transcribed parliamentary speeches?**. Through the discussion of the results in light of the evaluation framework, this experiment has served to help answer RQ4: **How do the topic modelling techniques from R1, perform when evaluated with the improved qualitative evaluation metrics from R3?**. After considering all the findings, we determined that the all-roberta-large-v1 model with BERTopic demonstrated the best overall performance among the tested variations in this experiment.

Trends from studying the results

When analyzing the results of LDA in Figures 5.9, 5.10, and 5.11, we observe a considerable variation between the models. While there are some similarities, such as the repetition of the topic word "norge" (Norway), the overall topics differ significantly. This is attributed to the random initialization of topics in LDA due to the Dirichlet process.

In contrast, the results of BERTopic in Figures 5.12, 5.13, and 5.17 display a higher level of similarity. For example, Topic#6 in Figure 5.12 closely resembles Topic#7 in Figure 5.13, and Topic#4 in Figure 5.14. These topics look like quite general topics, covering a lot of area and is most likely a result of the topic ordering process in BERTopic, where the first topics are the ones containing the most documents.

Turning our attention to the results of Top2Vec in Figures 5.15, 5.16, and 5.17, we notice that Topic#0 in Figures 5.15 and 5.16 primarily consists of stopwords, particularly in Nynorsk. Interestingly, these stopwords are not included in the current stopword list, which seems like a mishap. Another notable observation is that while there are some similarities among the samples, they are less pronounced compared to the results obtained with BERTopic.

Overall, the first eight topics from different embedding models exhibit significant similarity, prompting the question of how the subsequent eight topics compare to each other or to a random sample. This question forms the basis of Experiment 4, where different random samples will be compared with the initial eight topics.

Discussing the automatic metrics

When examining the results for LDA in Table 5.5, it is surprising to find that the model with the highest coherence score (1.56) and diversity score (0.79) is the p1000-t40 model, which also produces the largest number of topics. Typically, coherence and diversity metrics favor a lower number of topics, making this result unexpected.

For BERTopic, the automatic metric results in Figure 5.6 show that the TWE-nb-sbert-base model achieves the highest coherence score (1.63), while the nb-sbert-base model achieves the highest diversity score (0.88). The all-roberta-large-v1 model produces the most topics (54). The diversity scores among the models are generally similar, as are the coherence scores. One standout model is the TDE-nb-sbert-base, which achieves a coherence score of 1.12 and diversity score of 0.75 while only producing a single topic.

For Top2Vec, the results in Figure 5.7 show that the TDE-nb-sbert-base model achieves the highest coherence score (2.91) and diversity score (1), but it only produces two topics. Excluding the models that produce only one or two topics, the distiluse-base-multilingual-cased-v1 model achieves the second-highest coherence score (2.76), and the Doc2Vec model achieves the highest diversity score (0.92). The all-mini-LM-L12-v2 model produces the most topics (29), but its diversity score is only 0.32.

In summary, when disregarding models with only one or two topics, the distiluse-base-multilingual-cased-v1 model performs the best in terms of coherence for Top2Vec, while the Doc2Vec model achieves the highest diversity score. For BERTopic, the TWE-nb-sbert-base model achieves the highest coherence score, while the nb-sbert-base model demonstrates the greatest diversity. The all-roberta-large-v1 model produces the most topics for BERTopic.

Discussiong the evaluation framework ratings

The ratings for both experiment 4 and experiment 2 can be found in Tables 5.16, 5.8, and 5.12. In terms of LDA models, all three performed well across most categories, with the p1000-t40 model achieving the highest total score as well as the highest weighted total score.

However, qualitatively, Top2Vec exhibited limitations in terms of topic breadth. It tended to generate many similar topics, resulting in a low number of distinct topics. These topics often lacked recognizable entities and exhibited repetitive patterns across topics. While the individual topics themselves included notable words and preferred meaningful word classes such as noun phrases and verbs, they remained relatively shallow. Consequently, the topics only scratched the surface of more profound subject matters, with slight variations observed among repeated groups of topics.

On the other hand, BERTopic consistently produced a larger number of topics (more than 40) for most embedding models, except for the TDE-nb-sbert-base model, which performed poorly. The topics generated by BERTopic were diverse and of decent quality, particularly for the top-performing models such as TWE-nb-sbert-base and all-roberta-large-v1.

Considering the overall performance, the LDA-p1000-t40 model achieved the highest total score. However, when factoring in the number of topics as a weighting factor, the all-roberta-large-v1 model for BERTopic obtained the highest score.

Best performing model

After considering all the factors discussed thus far, it is determined that the best performing model variation is BERTopic with all-roberta-large-v1. This model will be selected as the preferred embedding model for the remaining experiments. The decision is based on the model's satisfactory performance in terms of automatic metrics and its top rating under the evaluation framework. It should be noted that the choice is subjective, and while not perfect, the differences between the best model and the second-best model are not significant, making the selected choice acceptable.

Gaming the Automatic Metrics

Upon further examination of the automatic metrics, a question arises: how can we easily create a model that achieves high scores in both diversity and coherence? Surprisingly, the answer is quite simple. To obtain a high diversity score, we can have a single topic, as having only one topic inherently makes the topics diverse.

Alternatively, if we have two topics, we can ensure they have no recurring words by designating one topic for verbs and the other for nouns. On the other hand, to manipulate coherence, we can use rare words that occur together. This means creating an esoteric topic where all the words are linked and occur in only a small number of documents. Similarly, a model with only one topic and highly frequent words would also score high in coherence due to the nature of the metric.

However, it is essential to recognize the limitations of such an approach. As stated by Goodhart's law, when a metric becomes the goal, it loses its effectiveness as a metric. Therefore, while this discussion sheds light on the topic, it is not particularly relevant in practice. The key takeaway is that coherence and diversity tend to favor a lower number of topics.

For example, in Table 5.7, the Top2Vec-tde-nb-sbert-base model achieves a diversity score of 1.0, primarily because it has only two topics. Similarly, the TDE-nb-sbert-base model for BERTopic (Table 5.6) produces only one topic but manages to achieve a diversity score of 0.75 and a coherence score of 1.12, which are higher than other models. However, it is important to note that a one-topic topic model is practically useless. This highlights the need for a critical evaluation of automatic metrics.

While automatic metrics provide some insights, it is crucial to consider their limitations and interpret the results with caution. The quality and usefulness of a topic model cannot solely be determined by these metrics alone.

5.4 Experiment 3 - User Testing Experiment

The purpose of this experiment was to gain insight into the process of user testing topic model results. Additionally, we aimed to get some external evaluation of the topic modelling results as well as insight into the automatic topic labelling results.

5.4.1 Experimental Setup

For this experiment, we used the topics produced by BERTopic with all-roberta-large-v1.

Furthermore, five tasks were included in the user test, along with some follow-up questions. In the subsequent subsections, for each task, the process of setting up the task will be described before the results are presented and discussed. Lastly we will summarize the results and discuss how they are related to the research questions.

To make things easier for the users, the same topics and documents will be used as much as possible, to avoid the user having to spend their time understanding documents or reading through topic words, instead of solving the task and giving usable results.

The user test setup is inspired by Rushfeldt [53].

The user test was conducted by creating a Google form survey. The survey was sent to individuals at NRK and we received responses from six individuals.

5.4.2 Task One - Inverse Topic Intrusion

Requirements for the inverse topic intrusion task

- All topics used should have high coherence
- Topic#1
- Document#1 included in Topic#1
- Short summary of Document#1
- Topics#2,#3,#4, that are unrelated to Topic #1

To create the task setup for the inverse topic intrusion task, we first chose a coherent topic to be the correct one. The properties we looked for when choosing the topic were for it to be of high quality, meaning that it was coherent and diverse. The next step was choosing a document included in the topic. Many of the documents in the NPM dataset are very long. To make the user test enjoyable for the user, it would be preferable to have shorter documents at the length of maximum two paragraphs. The easiest approach would be to slice the document to the preferred size, but this approach would potentially leave out crucial information and a sliced document could end mid-sentence. The solution to this issue was to use ChatGPT [3] and providing a prompt in the format of: "Create a concise summary of maximum two sentences in Norwegian for the provided text: <Document>." The last step to set up the inverse topic intrusion task was to find three unrelated topics. This was done by using the *visualize_topics*() and *visualize_heatmap*() methods of BERTopic.

Based on the requirements mentioned above, we found the following topic (topic#1) that serves as our correct one. And the following three unrelated topics (topic#2, topic#3, topic#4), that serves as intruders.

topic#1: energi, regjeringen, gasskraftverk, norge, kraft, industri, olje, fornybar, land.

[3]ChatGPT is a text generation model in the form of a chatbot, that can be used for free at `https://openai.com/blog/chatgpt`

topic#2: afghanistan, militære, nato, internasjonale, soldater, sivile, sikkerhet, enduring, krig, freedom

topic#3: asylsøkere, asylinstituttet, asyl, asylpolitikken, mindreårige, norge, asylpolitikk, beskyttelse, opphold, avslag

topic#4: film, filmen, filmer, kino, 1852, kautokeino, reklame, avgiften, statsråden, gaup.

And the following document summary that is connected to the correct topic.

Summary of dokument#1: Dokumentet diskuterer betydningen av fornybar energi i Norge og peker på viktigheten av økt kunnskap og debatt rundt grønne sertifikater og støtteordninger for energiproduksjon og energisparing. Det fremhever at Norge allerede er avhengig av ren, fornybar energi, men advarer om behovet for fortsatt tilgang til kraft i fremtiden. Dokumentet nevner også Norges rolle som en ledende fornybarnasjon og deltakelse i det grønne sertifikatmarkedet. Det understrekes at Norge bør være en pådriver for å oppnå klimamål og at teknologinøytrale støtteordninger for energisparing bør være en del av debatten.

The user was then first presented with the summary of the document, before being shown the four topics along with their respective wordcloud representations. The task was then to choose the topic that was the most descriptive of the document.

As can be seen in Figure 5.18, all participants got the correct answer on this task.

Quite a big limitation of this task and the setup is that the one preparing the user test can make or break it, by determining the difficulty through either choosing a different correct topic which might have a more ambiguous document attached to it or choosing more related topics.

5.4.3 Task Two - Wordclouds connected to documents task

In this task we present the user with a topic as a wordcloud along with a summary of the most representative document before asking them to rate the usefulness of the wordcloud on a likert scale of 5. Where 1 = not useful and 5 = very useful. We use the topic#2 from the previous task. topic#2: afghanistan, militære, nato, internasjonale, soldater, sivile, sikkerhet, enduring, krig, freedom.

The summary of the most representative document is as follows:

Summary of document#2: Dokumentet tar for seg situasjonen i Afghanistan og spørsmålet om forhåndslagring av militært utstyr. Det kritiserer deltakelsen i EU-

Figure 5.18: Inverse topic intrusion task results from the user test. The correct answer is: energi, regjeringen, gasskraftverk, ... , land.

innsatsstyrker og hevder at det representerer en gradvis bevegelse mot integrert forsvarssamarbeid som det norske folk har avvist. Dokumentet stiller også spørsmål ved behovet for fortsatt forhåndslagring av amerikansk utstyr i Norge. Det oppfordrer til en vurdering av Norges bidrag for å fremme langvarig fred og utvikling i Afghanistan, med fokus på demokrati, god styring og samfunnets behov.

In Figure 5.20, the ratings for the wordcloud representation of topic#2 are as follows: four ratings of 3, one rating of 2, and one rating of 1, resulting in an overall below-average rating.

After completing the task, participants were asked if they had any comments regarding the wordcloud. The responses, shown in Figure 5.19, highlight that some participants felt that specific topic words were missing and that the topic appeared to be more general, lacking the specificity of the individual documents. It is important to note that this outcome is expected since the topic covers multiple documents. While having a more general topic can be advantageous in certain cases, it also reveals the limitation of using topics when examining specific documents.

Overall, the ratings and comments from participants emphasize the challenge of accurately representing the content and context of individual documents solely through wordclouds based on topics.

5.4.4 Task Three - Word Intrusion

Requirements for the word intrusion task

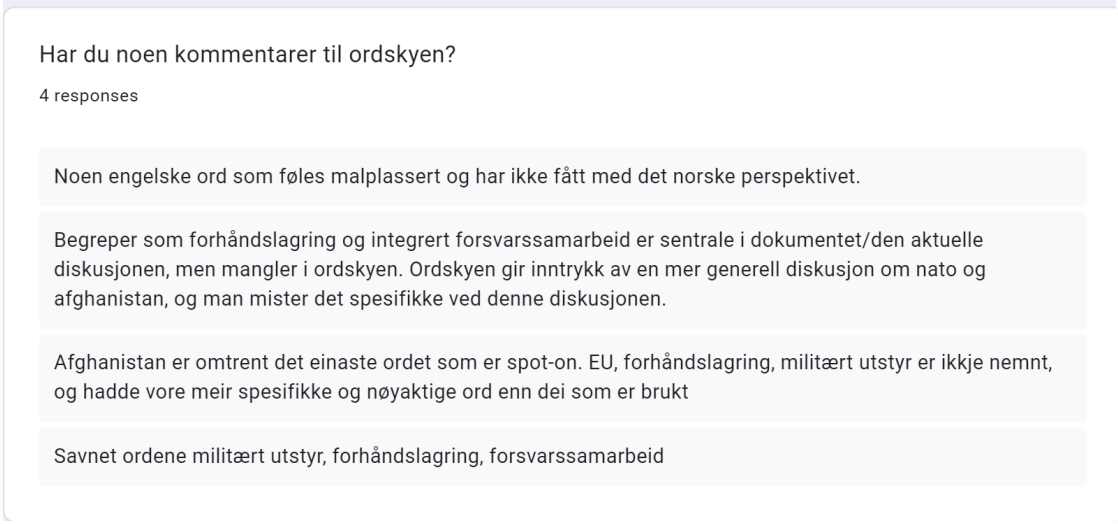


Figure 5.19: Ratings of the usefulness of the topic

- All topics used should have high coherence
- An unseen Topic#8
- Five most probable words from Topic#8
- Topic#N that is unrelated to Topic#8
- Most probable word from Topic#N

In this task we present the user with a single topic shown as six topic words. The catch is that one of the topic words does not belong there and it is up to the user to identify that word.

We chose the first five topic words of a new topic that the user had not seen before. Topic#8: mat, landbruk, landbruket, økologisk, bonden, produksjon, landbrukspolitikken, matproduksjon, bønder, produsere.

We then found a topic that is far away from this topic and chose the most probable word from that topic. The topic was chosen based on the *topic_visualization*() graph produced by BERTopic. The chosen topic was topic#42: israel, palestinsk, palestinske, hamas, palestinerne, stat, israelske, israels, høybråten, side.

Figure 5.21 shows that all 6 participants managed to get this task right. Showing that the topics do cover quite different themes and are unrelated showing the breadth of the results.

However we yet again observe the limitation of picking the topics and intruders. This task is quite easy, because we chose two topics that are very unrelated. However if we on the other hand had chosen a bit more related topics, the task could

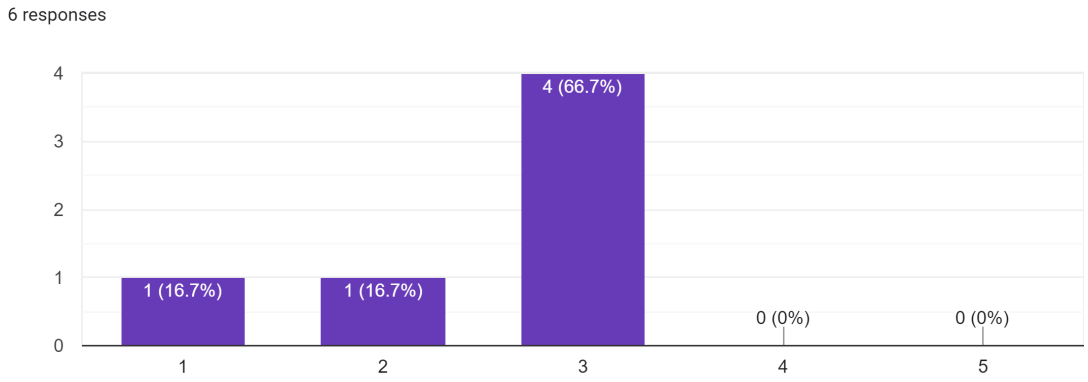


Figure 5.20: Responses to the question: Do you have any comments on the wordcloud in relation to task 2 from the user test.

become quite difficult. That is not to say that the results do not mean anything, because they do indicate that the topic model managed to produce topics that are far from each other and the topics from a qualitative point of view do look coherent and meaningful.

5.4.5 Task Four - Automatic Topic Label Rating

In the automatic topic label rating task, we compared the topic labels generated by the NETL and BERTopic methods to the most probable topic word from Topic#1. The participants were asked to rate the relevance of each label on a likert scale ranging from 1 to 3, where 1 represented "not very relevant" and 3 represented "relevant."

The topic and the corresponding labels are as follows:

Topic#1: energi, regjeringen, gasskraftverk, norge, kraft, industri, olje, fornybar, land.

NETL topic label: Oljeindustri

BERTopic automatic topic label: Fornybar energi

First topic word label: energi

The participants were then asked to provide ratings for each label.

In Figure 5.22, the results of the automatic topic label rating task are displayed.

Figure 5.21: Responses to the word intrusion task (task3) from the user test. The correct answer is Israel.

The NETL topic label received an average score of 2, while the BERTopic topic label received an average score of 2.3. The first topic word label obtained the highest rating with a score of 2.7. These findings indicate that the automatically generated topic labels were not highly relevant to the topics.

Interestingly, the most general topic label which was "energi", received the highest rating. This could be attributed to the presence of the word "energi" within the topic itself, potentially influencing the participants' judgments.

Overall, the results suggest that the automatic topic labelling methods, NETL and BERTopic automatic topic labelling, did not perform exceptionally well in this task. The first topic word label, which represents a more straightforward and specific approach, received the highest rating. It is worth noting that the performance of the automatic labelling methods might differ depending on the specificity of the topic. For this relatively general topic, the more general term served as a suitable descriptor.

5.4.6 Task Five - Topic Representation Preference

In this task we present the user with the two topic labels generated for the previous task along with the wordcloud representation of the topic.

Topic#1: energi, regjeringen, gasskraftverk, norge, kraft, industri, olje, fornybar, land.

NETL topic label: Oljeindustri

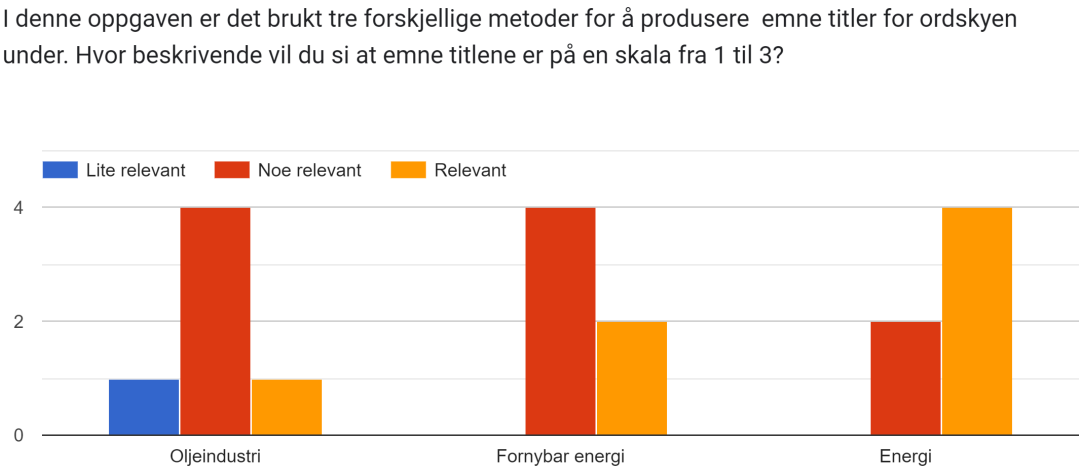


Figure 5.22: The results from the automatic topic label rating task for the topic "energi, regjeringen, gasskraftverk, norge, kraft, industri, olje, fornybar, land".

BERTopic automatic topic label: Fornybar energi

The user is then asked to select the representation method they preferred.

The responses to the task can be seen in Figure 5.23. 4 participants preferred the wordcloud, and 2 participants preferred the BERTopic automatic label.

A follow-up question to this task was also included, the results from which can be found in Figure 5.24. One response was that "the wordclouds are more nuanced than the topic labels, the topic labels give a narrower indication of the topic compared to the wordcloud." Another response was that "It is good to see multiple topic words when doing search for example, but when the topics are grouped it is nice to have topic labels." Overall the responses give the idea that topic labels could be useful, but only in certain settings. In other settings the wordcloud representation is a good alternative. This notion corresponds to the general idea behind topic labels that they are to replace wordclouds, but only in certain situations when you need a clear overview of a large number of topics. If we for example asked the users, which representation do you prefer? And presented a list of either 50 topic labels, or 50 wordclouds, it would be interesting to see the response and it would probably be more skewed towards the topic labels.

Another probing questions was also included, the results of which can be seen in Figure

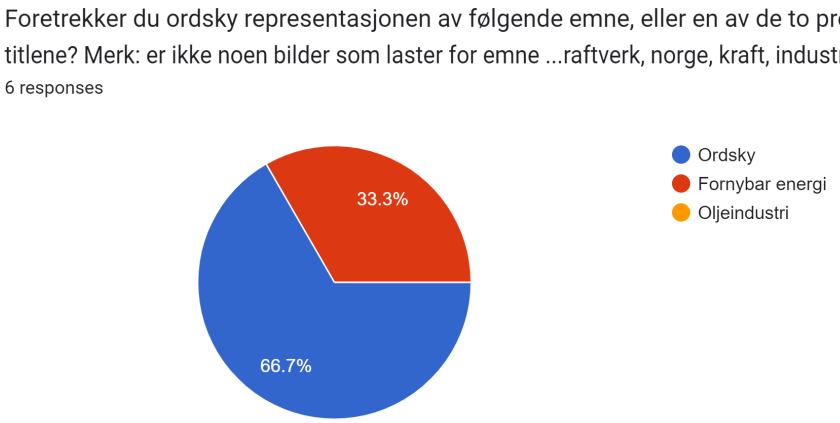


Figure 5.23: Responses to topic representation preference task from the user test.

5.4.7 Conclusion

In the previous subsections, we presented the task setup, results, and discussions from all of the tasks conducted in the user testing experiment. Although the number of participants and questions was limited, the results from the word intrusion task and topic intrusion task served as a validation that the chosen topics were of high quality, if not the topic model itself. The task of connecting documents to word clouds received slightly worse results, indicating the influence of topic selection on user perception. By rating the topics, we were able to provide some insights into Research Question 1 (RQ1).

The results from the preferred automatic topic label task and automatic topic label rating task, along with the follow-up questions, provided valuable insights into the characteristics of effective topic labels. These findings will influence our approach in Experiment 6 as we seek to answer Research Question 2 (RQ2).

It is important to note that the results, while valuable, are limited due to the small number of participants and questions. Despite this limitation, the qualitative nature of the testing proved helpful and informative, albeit time-consuming.

5.5 Experiment 4 - Evaluating Topic Models

The primary aim of this experiment was to apply the method described in Section 4.4 to a practical use case. Experiment 2 served as a suitable use case by focusing

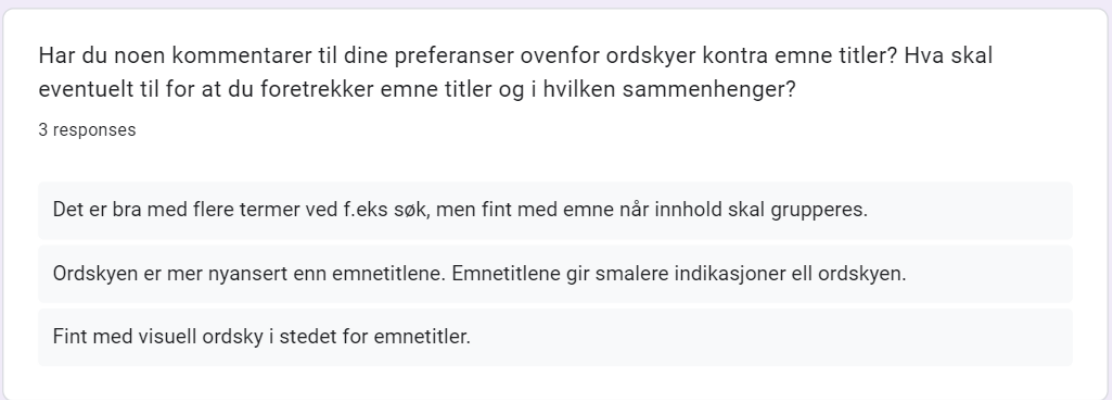


Figure 5.24: Responses to the question: Do you have any comments on your preferences for wordclouds versus topic labels? What does it take for you to prefer topic labels and in what contexts?

on determining the most compatible embedding model. In Experiment 4, we evaluated the results from Experiment 2 and expanded the evaluation by comparing the results with randomly sampled wordclouds. The objective of this extension was to determine if random sampling could provide a more representative representation of large topic models compared to sampling only the first eight wordclouds. Furthermore, the extension offered additional topic model results for evaluation and served as a validation of the evaluation scores. Any significant discrepancies observed between different samples from the same embedding model would weaken the confidence in the framework.

By conducting this experiment, the research aimed to contribute to the understanding of effectively evaluating embedding models through a qualitative evaluation framework.

5.5.1 Experimental Setup

For this experiment, 8-word samples were generated from every topic model variation tested in experiment 2. For every variation, three samples were created: a non-random sample, a random sample with $seed = 42$, and a random sample with $seed = 41$. All the samples were then evaluated using the evaluation framework. Selected complete ratings are included in Section 5.5.2, to illustrate the rating process and provide detailed explanations behind the ratings. Additionally all the ratings can be found in Section 5.5.3

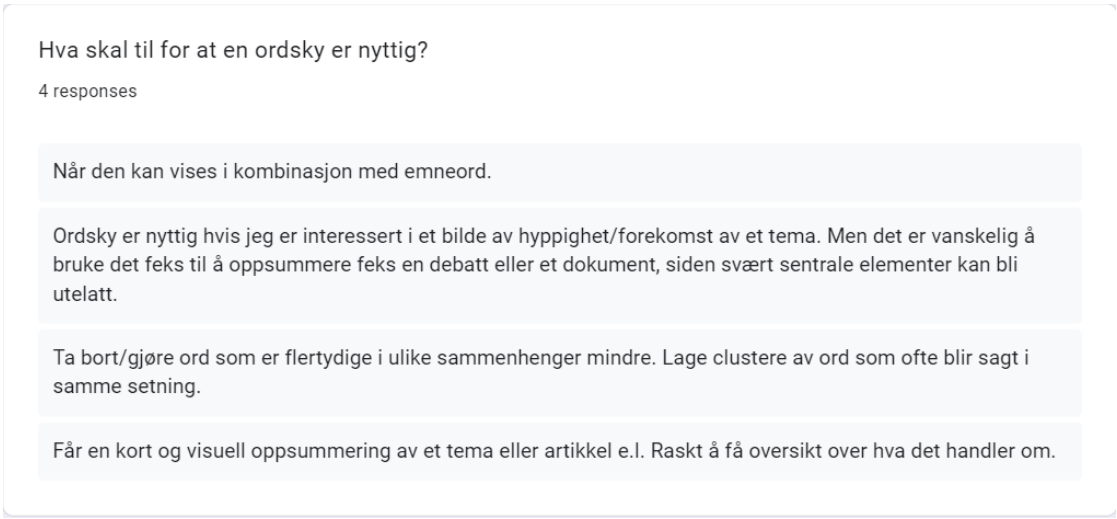


Figure 5.25: Responses to the question: What is needed for a wordcloud to be useful?

5.5.2 Selected Detailed Ratings

Rating BERTopic all-RoBERTa-large-v1 non-random sample

The detailed rating of the eight-topic non-random sample, produced by BERTopic with all-RoBERTa-large-v1 can be found in the list below. The wordcloud sample that is being rated can be seen in Figure 5.26.

Figure 5.26: Eight-topic non-random sample produced by BERTopic with all-RoBERTa-large-v1

- **Word classes: 5/5**

All the wordclouds are made up exclusively of nouns and verbs. The only words that are considered of lower value are: "gjelder" and "takker". "Gjelder" occurs in Topic#2 and Topic#3, while "takker" occurs in Topic#2. These two words provide little value and could be removed without the topics losing any meaning. However, because there are only three occurrences of such words, no points are subtracted.

- **Recognizable Entities: 4/5**
  The point deduction in this category comes from Topic#2 and Topic#5. The entities in these topics are recognizable to a certain degree by themselves, but a topic including all the political parties in the case of Topic#5 or the prime minister and parliament in Topic#2 are difficult to gauge what they are truly about, and hence the point deduction.
- **Related Entities: 5/5**
  Although the recognizability of certain topics is somewhat lacking, the entities included are all clearly related. None of the wordclouds include topic words that are clearly unrelated to the rest.
- **Word Repetition: 3/5**
  Four of the topics include word repetition: Topic#1, Topic#2, Topic#3, Topic#4. Topic#4 suffers the most from word repetition with the word "kommune" being repeated six times, and it could almost be enough for another point deduction. However, the rest of the topics are only mildly affected by the word repetition, and therefore three points are given.
- **Topic Similarity: 5/5**
  All topics are somewhat unique with Topic#2 and Topic#5 being the only topics that are a bit similar. This similarity is, however, not big enough to deduct a point, therefore five points are given.
- **Depth: 4/5**
  In this topic sample, we have good depth on some topics. The most in-depth topic is Topic#0, and Topic#3 is an honorable mention. The topics that cause the point deduction are Topic#2, Topic#4, and Topic#5. Topic#4 and Topic#5 are quite shallow and almost warrant a two-point deduction, but Topic#2 has some depth, and therefore four points are given.
- **Weighting: 1.54**
  The topic model produced 54 topics, which gives it a weighting of: $1 + \frac{54}{100} = 1.54$
- **Total score: 26/30**
  Weighted score: 40.0

Rating BERTopic all-RoBERTa-large-v1 random sample $seed = 41$

The detailed rating of the eight-topic random sample with $seed = 41$, produced by BERTopic with all-RoBERTa-large-v1 can be found in the list below. The wordcloud sample that is being rated can be seen in Figure 5.27.

Figure 5.27: Eight-topic random sample with seed = 41, produced by BERTopic with all-RoBERTa-large-v1

- **Word classes: 5/5**
  Some of the wordclouds include word classes other than nouns and verbs, such as Topic#24 and Topic#21 which include adjectives: "dyrare" and "nedsatt". However, these adjectives provide context to the topics; therefore, five points are given.
- **Recognizable Entities: 5/5**
  Every topic includes easily recognizable entities that are of high quality.
- **Related Entities: 5/5**
  Every topic includes entities that are clearly related in most cases. Not a lot to point out other than Topic#10 which is repeating itself, but the topic words are all related.
- **Word Repetition: 4.5/5**
  There are two topics with word repetition in this sample: Topic#36 and Topic#10. However, there is very little word repetition within those topics, and therefore only half a point is deducted.
- **Topic Similarity: 5/5**
  Every topic is unique and covers its own area.
- **Depth: 5/5**
  Every topic has quite some depth including many details about the topic.

- **Weighting: 1.54**
  The topic model produced 54 topics, which gives it a weighting of $1 + \frac{54}{100} = 1.54$.
- **Total score: 29.5/30**
  Weighted score: 45.4

Rating BERTopic all-RoBERTa-large-v1 random sample *seed* = 42

The detailed rating of the eight-topic random sample with seed = 42, produced by BERTopic with all-RoBERTa-large-v1 can be found in the list below. The wordcloud sample that is being rated can be seen in Figure 5.28.

Figure 5.28: Eight-topic random sample with seed = 42, produced by BERTopic with all-RoBERTa-large-v1

- **Word classes: 5/5**
  Every topic includes exclusively nouns and verbs, and the topic words are mostly meaningful.
- **Recognizable Entities: 5/5**
  Every topic includes easily recognizable entities and covers an area well.
- **Related Entities: 5/5**
  Every topic includes clearly related topic words, and it is quite effortless to see how they are related, except for Topic#17 with the topic words: "Kenneth", "øyvind", "korsberg", which requires very specific expert knowledge. Because only one topic includes somewhat unrelated topic words, 5 points are given.
- **Word Repetition: 3/5**
  Topics #47, #1, and #8 include varying degrees of word repetition, with Topic#47 being the most affected by it and the primary cause of the score

of 3 points.

- **Topic Similarity: 5/5**
  Every topic is unique and covers its own area.
- **Depth: 4/5**
  Almost every topic is sufficiently in-depth. The only two topics that have some issues with depth are: Topic#15 and Topic#17. Topic#17 includes some words that are too specific, such as "Kenneth", "øyvind", "korsberg". These words refer to specific parliamentary representatives, but the inclusion of these narrows down the topic too much. Topic#15 includes "Christian" and "tybringjedde" which refer to the politician "Christian Tybringjedde". Due to these two topics being too in-depth, 4 points are given.
- **Weighting: 1.54**
  The topic model produced 54 topics, which gives it a weighting of: $1 + \frac{54}{100} = 1.54$
- **Total score: 27/30**
  Weighted score: 41.6

Rating BERTopic nb-sbert-base non-random sample

The detailed rating of the eight-topic non-random sample, produced by BERTopic with nb-sbert-base can be found in the list below. The wordcloud sample that is being rated can be seen in Figure 5.29.

Figure 5.29: Eight-topic non-random sample, produced by BERTopic with nb-sbert-base

- **Word classes: 5/5**
  Every topic consists exclusively of nouns and verbs, and the topic words are

mostly meaningful, indicating a strong adherence to word classes.

- **Recognizable Entities: 5/5**
  Every topic includes easily recognizable entities that cover a diverse range, demonstrating a high level of topic specificity.
- **Related Entities: 5/5**
  Every topic includes clearly related entities. This indicates that the topics are well organized and make sense.
- **Word Repetition: 2/5**
  Topics #0, #2, #3, #4, #5, #6, and #7 all contain instances of word repetition, but only Topic #6 is significantly negatively affected by it. Hence, 2 points are given to reflect this limitation.
- **Topic Similarity: 4/5**
  Topic #3 and #6 exhibit similarity as they both revolve around political discussions. Topic #6 mentions "kommuner" (municipalities) and "staten" (the state), while Topic #3 includes the names of political parties and "regjeringen" (the government), which is often synonymous with "staten" (the state). This similarity warrants a score of 4.
- **Depth: 4/5**
  It would be preferable if Topic #3 and #6 included more in-depth topic words. Because the two topics are quite shallow, 4 points are given.
- **Weighting: 1.43**
  The topic model produced 43 topics, which gives it a weighting of $1 + \frac{43}{100} = 1.43$
- **Total score: 25/30**
  Weighted score: 35.8

Rating Top2Vec all-MiniLM-L12-v2 non-random sample

The detailed rating of the eight-topic non-random sample, produced by Top2Vec with all-MiniLM-L12 can be found in the list below. The wordcloud sample that is being rated can be seen in Figure 5.30.

Figure 5.30: Eight-topic non-random sample, produced by Top2Vec with all-MiniLM-L12-v2

- **Word Classes: 3/5**
  Because all the topics include the adjective "selvfølgelig" ("of course"), a rating of 3 points is given. The sample could be rated even lower; however, most of the topics only include one unrelated word class, and at most two. Therefore, the overall effect of the adjectives is lower than if a few topics contained many unrelated word classes because the topics still convey some meaning through the rest of the topic words.
- **Recognizable Entities: 2/5**
  With the exception of Topic#5, the topics lack clear and distinct themes, making them challenging to identify. The connections between the words are not strong enough to easily recognize the intended meaning, resulting in a rating of 2 points.
- **Related Entities: 3/5**
  While the words in each topic are somewhat related, the relationships between them are quite weak. The topics may share a broad theme, but the connections among the words are not particularly strong. Considering this, a rating of 3 points is given.
- **Word Repetition: 1/5**
  Unfortunately, all the topics suffer from word repetition, which negatively impacts their quality. The excessive repetition of words reduces the uniqueness and interest of the topics, warranting a low rating of 1 point.
- **Topic Similarity: 2/5**
  Topics #1, #5, and #6 show a slightly higher level of distinctiveness compared to the others. Although there are some differences, the remaining topics could be merged without significant loss of information. Taking this into

account, a rating of 2 points seems appropriate.

- **Depth: 2/5**
  Despite the overall similarity among topics, they do offer some depth by including additional relevant words. While not extensive enough to warrant a higher rating, the topics possess a certain level of depth, justifying a rating of 2 points.
- **Weighting: 1.29**
  The topic model produced 29 topics, which gives it a weighting of: $1 + \frac{29}{100} = 1.29$
- **Total Score: 13/30**
  Weighted Score: 16.8

Rating LDA with 20 topics and 1000 passes non-random sample

The detailed rating of the eight-topic non-random sample, produced by LDA with the parameters $num\_topics = 20$ and $passes = 1000$, can be found in the list below. The wordcloud sample that is being rated can be seen in Figure 5.31.

Figure 5.31: Eight-topic non-random sample, produced by LDA with $num\_topics = 20$ and $passes = 1000$.

- **Word Classes: 4/5**
  Topic#4 includes "dessverre," and Topic#7 includes "bedre." These words are somewhat related to the respective topics, but especially "dessverre" provides little value to the topic, which warrants a rating of 4 points.
- **Recognizable Entities: 5/5**
  All the topics include recognizable entities, and each topic represents a distinct theme. The presence of clear themes across all topics merits a rating of

5 points.

- **Related Entities: 4/5**
  While most topics exhibit a strong connection between the words, Topic#4 has a slightly lower level of relation. Additionally, Topic#2, involving "toget" or "the train," is a bit challenging to see how it relates to the rest of the words. Due to these factors, the topic receives a rating of 4 points.
- **Word Repetition: 4/5**
  There is some repetition observed in Topic#3, Topic#7, Topic#2, and Topic#4. However, the repetition within each topic is not extensive. Considering this, the sample is awarded 4 points.
- **Topic Similarity: 5/5**
  All topics demonstrate uniqueness and possess distinguishing characteristics, earning a rating of 5 points.
- **Depth: 5/5**
  The topics show sufficient depth, covering various aspects and providing substantial information, warranting a rating of 5 points.
- **Weighting: 1.20**
  The topic model produced 20 topics, which gives it a weighting of: $1 + \frac{20}{100} = 1.20$
- **Total Score: 27/30**
  Weighted Score: 34.8

5.5.3 Compiled Ratings

Table 5.8: BERTopic results rated on non-random sample

Metrics	Models						
	L12	L6	nb-sbert	distiluse-v2	tde-nb-sbert	twe-nb-sbert	roberta
Word Classes	5	5	5	4	5	5	5
Recognizable Entities	4	4	5	4	4	4	4
Related Entities	4	5	5	4	4	5	4
Word Repetition	3	2	2	2	5	4	3
Topic Similarity	4	4	4	4	5	4	5
Depth	4	4	4	4	3	4	4
Total	24	24	25	22	26	26	26
Weighting	1.49	1.48	1.43	1.4	1.01	1.44	1.54
Weighted Total	35.76	35.52	35.75	30.8	26.26	37.44	40.04

Table 5.9: BERTopic results rated random sample $seed = 41$

Metrics	Models						
	L12	L6	nb-sbert	distiluse-v2	tde-nb-sbert	twe-nb-sbert	roberta
Word Classes	4	4	5	5	5	5	5
Recognizable Entities	5	5	5	5	4	5	5
Related Entities	5	4	5	4	4	5	5
Word Repetition	4	3	3	3	5	3	4.5
Topic Similarity	5	4	5	5	5	5	5
Depth	5	5	5	5	2	5	5
Total	28	25	28	27	25	28	29.5
Weighting	1.49	1.43	1.43	1.4	1.01	1.44	1.54
Weighted Total	41.72	35.75	40.04	37.8	25.25	40.32	45.43

Table 5.10: BERTopic results rated on a sample $seed = 42$

Metrics	Models						
	L12	L6	nb-sbert	distiluse-v2	tde-nb-sbert	twe-nb-sbert	roberta
Word Classes	4	4	5	4	5	5	5
Recognizable Entities	4	4	5	4	4	5	5
Related Entities	4	4	5	3	5	5	5
Word Repetition	3	2	2	2	5	4	3
Topic Similarity	3	4	5	4	5	5	5
Depth	2	3	5	2	3	5	4
Total	20	21	27	19	27	29	27
Weighting	1.49	1.48	1.43	1.4	1.01	1.44	1.54
Weighted Total	32.34	31.08	38.61	26.6	27.27	41.76	41.58

Table 5.11: Average BERTopic results

Metrics	Models						
	L12	L6	nb-sbert	distiluse-v2	tde-nb-sbert	twe-nb-sbert	roberta
Word Classes	4.3	4.3	5	4.3	5	5	5
Recognizable Entities	4.3	4.3	5	4.3	4	4.7	4.7
Related Entities	4.3	4.3	5	3.7	4.3	5	5
Word Repetition	3.3	2.3	2.3	2.3	5	3.7	3.5
Breadth	4	4	4.7	4.3	5	4.7	5
Depth	3.7	4	4.7	3.7	2.7	4.7	4.3
Total	24	23.3	26.7	22.7	26	27.7	27.5
Weighted Total	35.76	34.484	38.181	31.78	26.26	39.888	42.35
Standard Deviation	4	2.1	1.5	4	1	1.5	1.8
Average STD				2.3			

Table 5.12: Top2Vec results rated on non-random sample

Metrics	Models						
	L12	distiluse-v2	distiluse-v1	doc2vec	nb-sbert	tde-nb-sbert	universal
Word Classes	3	4	4	4	5	3	3
Recognizable Entities	2	4	4	4	4	2	2
Related Entities	3	4	4	4	4	2	3
Word Repetition	1	2	2	4	4	2.5	2
Topic Similarity	2	3	4	4	4	5	1
Depth	2	3	4	4	4	2	2
Total	13	20	22	24	25	16.5	13
Weighting	1.29	1.25	1.23	1.22	1.23	1.02	1.02
Weighted Total	16.77	25	27.06	29.28	30.75	16.83	13.26

Table 5.13: Top2Vec results rated on random sample $seed = 41$

Metrics	Models						
	L12	distiluse-v2	distiluse-v1	doc2vec	nb-sbert	tde-nb-sbert	universal
Word Classes	5	4	5	4	5	5	2
Recognizable Entities	3	4	5	5	5	2	2
Related Entities	4	4	5	5	4	3	1
Word Repetition	3	2	3	3	3	5	3
Topic Similarity	3	3	4	4	5	4	1
Depth	3	3	4	4	3	2	1
Total	21	20	26	25	25	21	10
Weighting	1.29	1.25	1.23	1.22	1.23	1.02	1.02
Weighted Total	27.09	25	31.98	30.5	30.75	21.42	10.2

Table 5.14: Top2Vec results rated on random sample $seed = 42$

Metrics	Models						
	L12	distiluse-v2	distiluse-v1	doc2vec	nb-sbert	tde-nb-sbert	universal
Word Classes	5	4	5	5	4	4	3
Recognizable Entities	2	3	4	4	4	3	1
Related Entities	4	4	4	4	4	2	2
Word Repetition	1	2	3	3	3	4	1
Topic Similarity	2	2	4	4	3	3	1
Depth	2	3	4	4	3	3	2
Total	16	18	24	24	21	19	10
Weighting	1.29	1.25	1.23	1.22	1.23	1.02	1.02
Weighted Total	20.64	22.5	29.52	29.28	25.83	19.38	10.2

Table 5.15: Average Top2Vec results

Metrics	Models						
	L12	distiluse-v2	distiluse-v1	doc2vec	nb-sbert	tde-nb-sbert	universal
Word Classes	4.3	4	4.7	4.3	4.7	4	2.7
Recognizable Entities	2.3	3.7	4.3	4.3	4.3	2.3	1.7
Related Entities	3.7	4	4.3	4.3	4	2.3	2
Word Repetition	1.7	2	2.7	3.3	3.3	3.8	2
Topic Similarity	2.3	2.7	4	4	4	4	1
Depth	2.3	3	4	4	3.3	2.3	1.7
Total	16.7	19.3	24	24.3	23.7	18.8	11
Weighted Total	21.543	24.125	29.52	29.646	29.151	19.176	11.22
Standard Deviation	4	1.2	2	0.6	2.3	2.3	1.7
Average STD	2.0						

Table 5.16: LDA results rated on non-random sample

Metrics	Models		
	p1000_t20	p1000_t30	p1000_t40
Word Classes	4	4	4
Recognizable Entities	5	5	5
Related Entities	4	4	5
Word Repetition	4	5	4.5
Topic Similarity	5	5	5
Depth	5	5	5
Total	27	28	28.5
Weighting	1.2	1.3	1.4
Weighted Total	32.4	36.4	39.9

Table 5.17: LDA results rated on random sample $seed = 41$

Metrics	Models		
	p1000_t20	p1000_t30	p1000_t40
Word Classes	4	4	5
Recognizable Entities	3	4	5
Related Entities	4	5	4
Word Repetition	5	5	3
Topic Similarity	3	4	5
Depth	3	4	4
Total	22	26	26
Weighting	1.2	1.3	1.4
Weighted Total	26.4	33.8	36.4

Table 5.18: LDA results rated on random sample $seed = 42$

Metrics	Models		
	p1000_t20	p1000_t30	p1000_t40
Word Classes	5	4	3
Recognizable Entities	5	4	4
Related Entities	5	4	4
Word Repetition	4	3	5
Topic Similarity	4	4	4
Depth	4	3	4
Total	27	22	24
Weighting	1.2	1.3	1.4
Weighted Total	32.4	28.6	33.6

Table 5.19: Average LDA results

Metrics	Models		
	p1000_t20	p1000_t30	p1000_t40
Word Classes	4.3	4	4
Recognizable Entities	4.3	4.3	4.7
Related Entities	4.3	4.3	4.3
Word Repetition	4.3	4.3	4.2
Breadth	4	4.3	4.7
Depth	4	4	4.3
Total	25.3	25.3	26.2
Weighted Total	30.36	32.89	36.68
Standard Deviation	2.9	3.1	2.3
Average Standard Deviation		**2.8**	

5.5.4 Discussion

As discussed in Section 4.4, our answer to RQ3: **How can the qualitative evaluation of topic models be improved?**, is the evaluation framework. Through the extensive testing in this experiment, we have gained further insight into how the framework works in practice and the detailed rating can serve as a guide to potential future users of the framework.

The evaluation framework turned out to work quite nicely in practice and give some great guidelines to rate the samples and importantly ensure consistency across different samples. Without the framework, one can easily change the rating metrics from sample to sample. However, by following the guidelines set in the evaluation framework, the ratings become systematic and the quality is enhanced.

Possible improvements to the evaluation framework

Further improvements to the evaluation framework could potentially be to remove categories or include new ones. Maybe change the names of the categories some of them might not be precise enough. Also, a potentially confusing part of the framework is that some of the ratings are inversed, as can be seen in Figure 4.5. Sometimes "None", means 1 point, while other times it is 5 points. This is simply due to the wording of the categories, and the category itself could be reversed to make the ratings more consistent.

Another improvement could be to change the wording of the categories for each score or the limits for what constitutes each score. The issue with these limits is that at times they require quite a lot of human thinking and decision-making. Decision-making is quite tiresome, so taking this thinking part out of the evaluation would be helpful for the evaluator, although the ratings themselves might suffer a little bit. Otherwise, you could also take a deeper look at coherence and diversity and really try to make the categories match them more.

Also, the whole evaluation process could actually be automated at least some of the steps. Word repetition and word classes stand out in this sense among others. Some NLP-based methods could potentially replace them.

The rating process

The whole rating process was quite time-consuming, especially during the beginning phases. After rating a couple of samples one got into the groove of things and started remembering how everything should be rated. We started to get an intuition of how a 20 total score rating sample look like compared to a 25 total score sample. A good idea would be to initially start the rating process on some trial samples to warmup before beginning the actually rating task.

The process of how we got the ratings was that we first rated every sample, without writing down our reasons behind them. Then we chose some different samples to write down the detailed reasoning. This step sort of served as an indicator of whether we were accurate or not, and we found that we did not really change our mind about the scoring when we were forced to write down the explanations. The very good topic samples and the very poor samples were easier to rate than the mediocre ones. This stems from the fact that the 1 and 5-point categories are quite strict in the wording with "all" and "none". While for the values in between the wording was less precise.

Discussing the ratings

Now we will discuss the ratings as they are grouped by topic model. We will mostly be focusing on how the framework worked in practice and how random sampling affected the results. The comparison between the embedding models is left for experiment 2.

LDA

The results from LDA non-random sample ratings can be seen in Table 5.16. The difference in total score between each model is quite low in this sample compared to the $seed = 41$ sample in Table 5.17 and $seed = 42$ sample in Table 5.18.

The average rating scores can be seen in Table 5.19. The average total rating is 25.3 for p1000-t20 and p1000-t30, and 26.2 for p1000-40. The average total rating for each LDA variation was quite close to each other, however when looking at the average ratings by category we can see that there are some differences in how each variation scores on the different categories. The model with the lowest standard deviation was p1000-t40 (2.3). The average standard deviation was 2.8, which seems quite high.

Top2Vec

The results from Top2Vec non-random sample ratings can be seen in Table 5.12. For $seed = 41$ sample results can be seen in Table 5.13 and for $seed = 42$ sample results can be seen in Table 5.14. The total value varies quite a lot from embedding model to embedding model which indicates that the choice of model has a lot of influence on the result. The differences in total rating from sample to sample are quite similar.

The average rating scores can be seen in Table 5.15. The average total rating for each Top2Vec variation was quite far from each other similarly to what was seen in the samples. When looking at the average ratings by category there seems to be a lot of information that could possibly be extracted based on which category an embedding model performs well and poorly in, you could extract what type of embedding model it is. The model with the lowest standard deviation was doc2vec (0.6), which is a very low value. Compared to the highest standard deviation from L12 (4), the difference is quite astounding, and it indicates that the quality of the topics from the L12 model was varied. The average standard deviation was 2.0, which was quite a bit less than the 2.8 for LDA. This indicates that the evaluation framework was more consistent in rating the top2vec samples as compared to LDA samples. The reason why might simply be because we have a sample size of 7 models here, compared to 3 for LDA.

BERTopic

The results from BERTopic non-random sample ratings can be seen in Table 5.8. The difference in total score between each model is quite low in this sample compared to the *seed* = 41 sample in Table 5.17 and *seed* = 42 sample in Table 5.18.

The average rating scores can be seen in Table 5.19. The average total rating is 25.3 for p1000-t20 and p1000-t30, and 26.2 for p1000-40. The average total rating for each LDA variation was quite close to each other, however when looking at the average ratings by category we can see that there are some differences in how each variation scores on the different categories. The model with the lowest standard deviation was p1000-t40 (2.3). The average standard deviation was 2.8, which seems quite high.

Comparison between the models

The highest average total rated variation for Top2Vec was the Doc2Vec (24.3), which also achieved the lowest standard deviation (0.6) by far. The Doc2Vec model was also the highest average weighted total (26.65).

The highest average total rated variation for LDA was the p1000-t40 (26.2), which also was the highest total weighted (36.68).

The highest average total rated variation for BERTopic was the twe-nb-sbert (27.7).

The highest weighted total model was the BERTopic all-roberta-large-v1 which achived an average total rating of 27.5, close behind twe-nb-sbert, but it produced 10 more topics than twe-nb-sbert at 54 topics, which gave it a weighted total of 42.35.

General remarks

In general it is time-consuming to use the evaluation framework. One really has to make efforts to focus. In one way, this is a good thing, because it means that the ratings are sort of quality ensured. Especially when writing down detailed ratings it is difficult to "cut corners".

The method has multiple limitations, the biggest one being the human rater, and especially in this case where we have a sample size of 1. At least for user testing, you can aim for about 10 responses, but using this evaluation framework we are limited to a lower sample size, and in our case a sample size of 1 rater.

5.6 Experiment 5 - Norwegian Parliament-Large Experiment

The objective of this experiment was to gain insight into the functionality of topic modelling when applied to large document collections. The NPL-stopwords dataset, outlined in Table 4.1, was utilized. Of particular interest was the utilization of dynamic topic modelling to gain insights into how political discussions evolved over time and were influenced by significant events.

5.6.1 Experimental Setup

For this experiment, BERTopic with all-roberta-large-v1 was selected as the topic model, based on the results from experiment 2. An additional reason for choosing BERtopic was due to its extensive built-in options for configuration, such as the number of topics and dynamic topic modelling capabilities. The NPL-stopwords dataset was used because preliminary findings indicated that the models utilizing NPM-stopwords produced the most promising results. Given that NPM is a subset of NPL, it was assumed that NPL-stopwords would be the most suitable dataset.

Due to the large size of the NPL dataset, cloud computing was required to effectively load the data into memory and perform topic modelling. To accomplish this, a VM instance was created using free student credits on Google Cloud. Jupyter Notebook was run via SSH for seamless execution.

The first set of results were the topics along with the automatic metrics. Additionally, we explored some dynamic topic modelling and we will show some of the dynamic topic modelling results we found interesting.

5.6.2 Results and Discussion

The first thing we noticed was that the topic model produced a lot of topics, 3079 topics to be exact. This was not quite expected, but it seems like scaling up the dataset size, scales up the number of topics which does make sense.

The second thing we noticed was that the number of documents assigned to each topic as can be seen in Figure 5.39, varied quite a lot going from 8531 to 10, which was the minimum number of documents for a cluster to form. This hints to the possibility of further improvements if the minimum number of documents was to be reduced. Another interesting thing to note is that there are many documents assigned to the topic "-1". This topic represents all the documents that are outliers. This number seemed to really scale up, when the dataset increased in size. It

is actually almost half the documents being unassigned. BERTopic provided a convinent method to reduce outliers, and after applying this we got the following results in Figure 5.40, which is difficult to measure if it actually improved.

Third thing we noticed was that the automatic metrics were quite abstract at this point, and could not really be used to correctly gauge the performance of the model, especially when we were unable to rapidly iterate and produce new variations to have something to compare the metrics to. Table 5.22 shows that the model got a coherence score of 1.71 and a diversity of 0.45. Without any models to compare the results with on the same dataset, it was difficult to say how good the scores were. Especially true for coherence, because the NPMI was not properly normalized. Suppose it was properly normalized we could see how far away from the ceiling (1) it was. To get an idea of how the results fared, we chose to use the evaluation framework with the same approach that was experimented with in experiment 4; producing three different samples. The first one is the first eight topics, the second one is a random sample with $seed = 41$, and the third one is a random sample with $seed = 42$. We will now present the compiled ratings in Table 5.20 and the average ratings in Table 5.21.

Table 5.20: BERTopic all-roberta-large-v1 compiled ratings from three different samples from NPL.

Metrics	**non-random**	**seed 41**	**seed 42**
Word Classes	4	4	3
Recognizable Entities	3	3	3
Related Entities	3	4	2
Word Repetition	1	3.5	3
Topic Similarity	4	4	3
Depth	1	2	2
Total	16	20.5	16
Weighting	3080	3080	3080
Weighted Total	49280	63140	49280

Table 5.20 shows the ratings for each of the random samples. What we notice is that the total rating is quite similar with the standout being $seed = 41$ sample, with a total of 20.5 compared to 16 for the other two samples. However within the categories the scores do vary. The worst scoring categories on average were depth and word repetition. Another thing to notice is that the weighting becomes a very large number, causing the weighted total to be large as well, however because the weighting is the same for all the samples, we can still use it. If we were to compare two models that had a different number of topics, both of them large. As long as the values are within a certain range the weighted total makes sense. However if there is a large difference between number of topics the weighted total becomes less of an efficient metric. The average total rating of the samples can be seen

Table 5.21: BERTopic all-roberta-large-v1 average results from the three samples tested from NPL.

Metrics	Value
Word Classes	3.7
Recognizable Entities	3
Related Entities	3
Word Repetition	2.5
Topic Similarity	3.7
Depth	1.7
Total	17.5
Weighting	3000
Weighted Total	53900
Standard Deviation	2.6

Coherence	1.71
Diversity	0.45

Table 5.22: Coherence and diversity scores for BERTopic all-roberta-large-v1 run on NPL

in Table 5.21 to be 17.5 If we compare this value to the ones in the embedding experiment we find it to be quite a low value indicating that the topic model is performing a fair bit under average actually. It is easy to be tricked into thinking that the topic model is very good because it has produced a lot of topics, but we can see here that many topics do not always equal good results.

During our evaluation of wordcloud samples using the evaluation framework, we observed certain patterns and aspects that stood out. One notable observation was the presence of domain-specific stopwords, such as "replikkordskiftet," which appeared in multiple topics, including topic#0, topic#2, and a randomly sampled topic#2826. Additionally, when working with larger datasets, the inclusion of specific numbers in the topics became more challenging to interpret and understand, as they lacked contextual information. Overall, these factors contributed to lower-quality topics, particularly in terms of recognizable entities and depth, as reflected in Table 5.21.

The random samples provided a much wider point of view to the topics and is quite enjoyable to have and compare with the first eight topics, which are more general topics.

In the context of dynamic topic modelling, we focused on four topics that exhibited changes in frequency over time, indicating their sensitivity to external events. The topics were as follows:

Topic#5: energiministeren, energi, fornybar, olje, energidepartementet, energief-

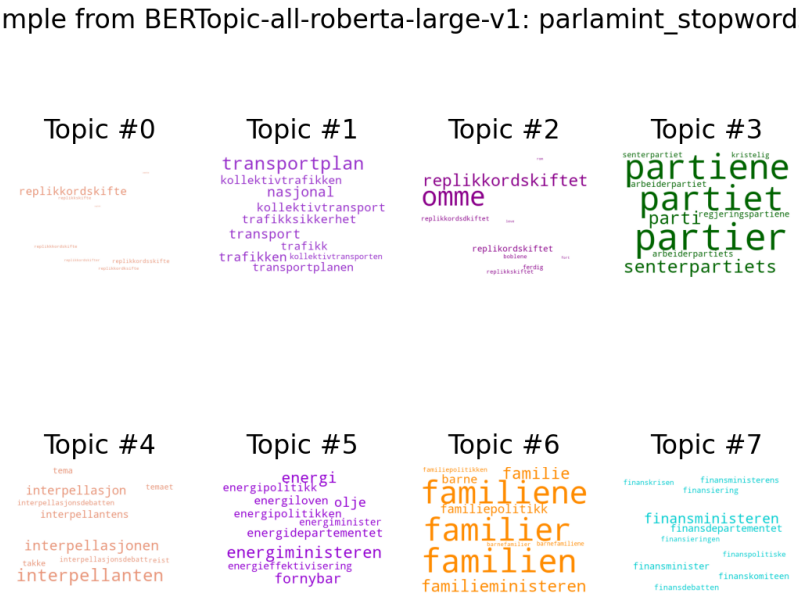


Figure 5.32: Eight-topic non-random sample, produced by BERTopic with all-RoBERTa-large-v1 on NPL

fektivisering, energipolitikken, energiloven, energipolitikk, energiminister

Topic#7: finansministeren, finansdepartementet, finansminister, finanskomiteen, finansministerens, finansdebatten, finansiering, finanskrisen, finanspolitiske, finanseringen

Topic#15: psykisk, psykiske, helsevern, psykiatrien, psykiatri, lidelser, rus, opptrappingsplanen, ungdomspsykiatrien, psykiatriske

Topic#17: landbruk, landbruksministeren, landbruket, landbrukspolitikk, landbrukspolitikken, matproduksjon, mat, landbruks, økologisk, landbruksmeldingen

Figure 5.35 illustrates the frequency fluctuations of these topics. Looking at specific topics and points in time, we can see in Figure 5.36, that topic#5 includes the topic word "enova" in 2009, coinciding with the event where Enova was allocated an additional 1.2 billion NOK in response to the financial crisis of 2008. Similarly, Figure 5.37 showcases the inclusion of "strømpriser" ("energy prices") in a topic, which aligns with the period of exceptionally high energy prices that started in 2021. Another example is topic#7 in Figure 5.38, which exhibits a spike

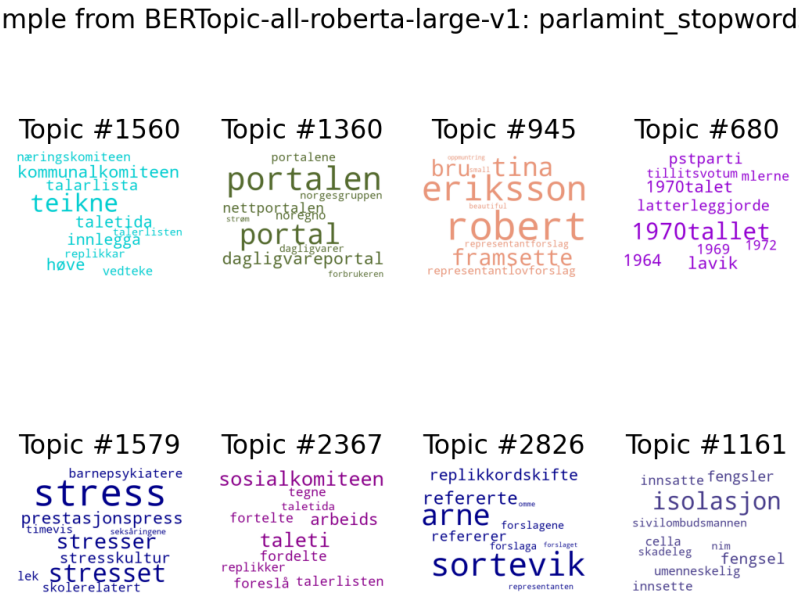


Figure 5.33: Eight-topic random sample with $seed = 41$, produced byBERTopic with all-RoBERTa-large-v1 on NPL

in frequency in 2008, along with the presence of the term "finanskrisen" (financial crisis) indicating that the topic picked up on the financial crisis. These intriguing findings highlight how dynamic topic modelling captures shifts in topic prevalence linked to historical events.

Our qualitative ratings and exploration of wordclouds shed light on the characteristics and nuances of the topics. We encountered domain-specific stopwords and challenges with interpreting specific numbers.

5.6.3 Limitations

A large limitation to this approach was that the compute requirements were quite ridiocolus. The main limitation was that BERTopic primarily used CPU, as far as we were aware of it was the only option for the bottleneck operations. We used the maximum available CPU setup on google cloud which was 24 virtual CPUs with 96gb memory. However the estimate to train on the full dataset was around 80 hours, so we cropped each document to size 255. Which took the time down to around 14 hours. Next, the 96gb of memory was not enough to calculate the full

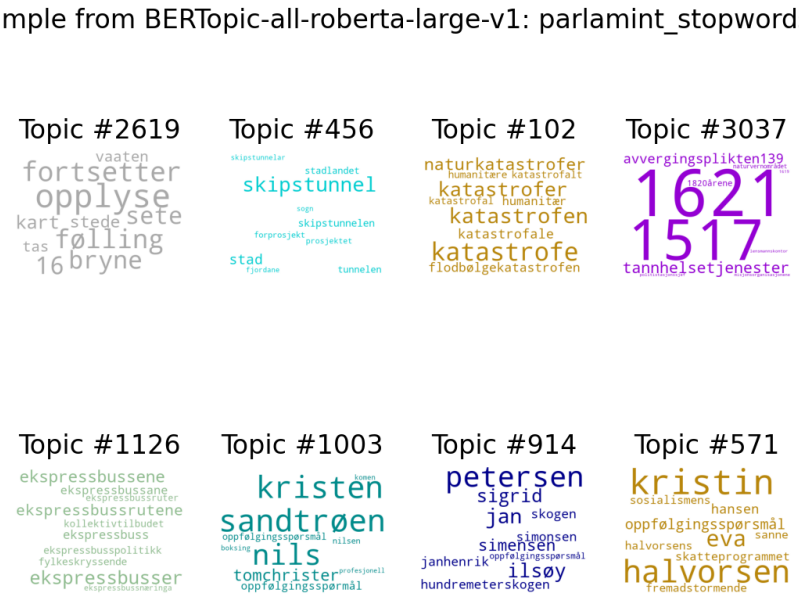


Figure 5.34: Eight-topic random sample with $seed = 42$, produced by BERTopic with all-RoBERTa-large-v1 on NPL

word-co frequency matrix. The kernel would crash when doing the operations so we had to crop the dataset to the first 95% of the documents. It is possible that due to cropping the documents, the results were negatively affected.

5.6.4 Conclusion

Overall this experiment showed the process of doing topic modelling on larger datasets which can be more transferable to real-world applications. In relation to the research questions this experiment served as an answer to the second part of RQ1: **how do they perform on Norwegian transcribed parliamentary speeches?**, as well as being a practical usecase of the evaluation framework, hence answering RQ4: **How do the topic modelling techniques from R1, perform when evaluated with the improved qualitative evaluation metrics from R3?**

The evaluation framework proved itself as a helpful tool to judge the topic models performance. Dealing with such large topic models can be an large task, and automatic metrics proved themselves to be of little value. User testing requires a lot of

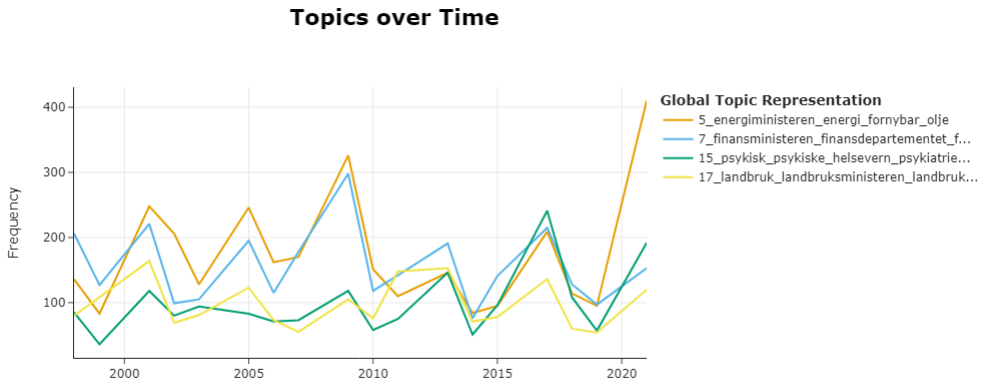


Figure 5.35: Dynamic topic modelling visualization. Showing four topics and how they change over time. The frequency refers to how many documents per year are included in each topic.

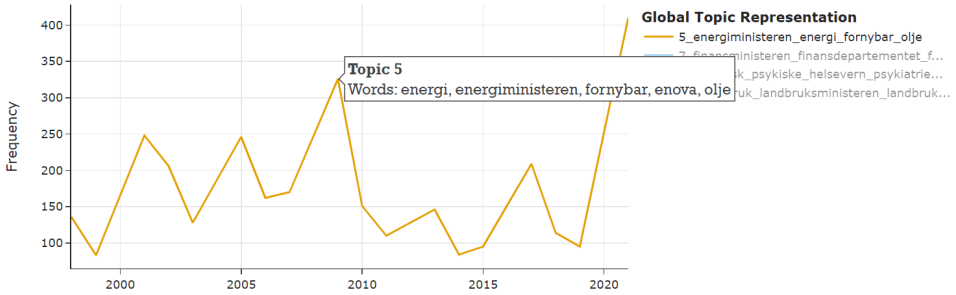


Figure 5.36: Dynamic topic modelling visualization. Showing a single topic#5, highlighted at the year of 2009, where we can see the topic words.

time in preparation and execution. Therefore the evaluation framework provided a systematic approach to judging the results. The random sampling provided a broader point of view to the topics and was quite useful in giving deeper insights into how a more specific topic would look like. Quite enjoyable to compare the random samples with the non-random sample.

Additionally, our analysis of dynamic topic modelling provided an interesting look at how temporal events affected the parliamentary discussions as reflected in the topics.

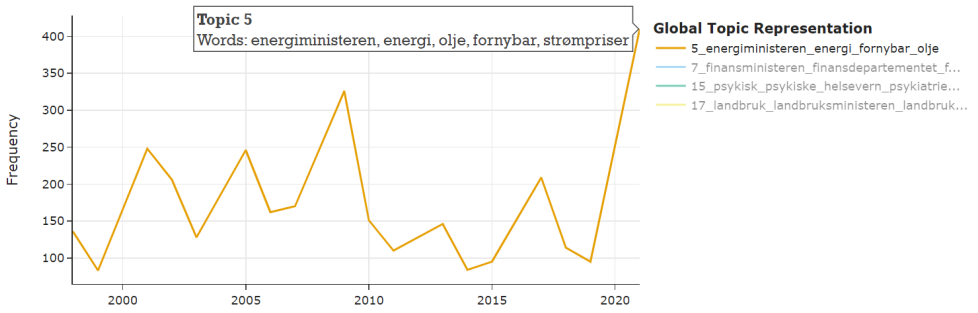


Figure 5.37: Dynamic topic modelling visualization. Showing a single topic#5, highlighted at the year of 2022, where we can see the topic words.

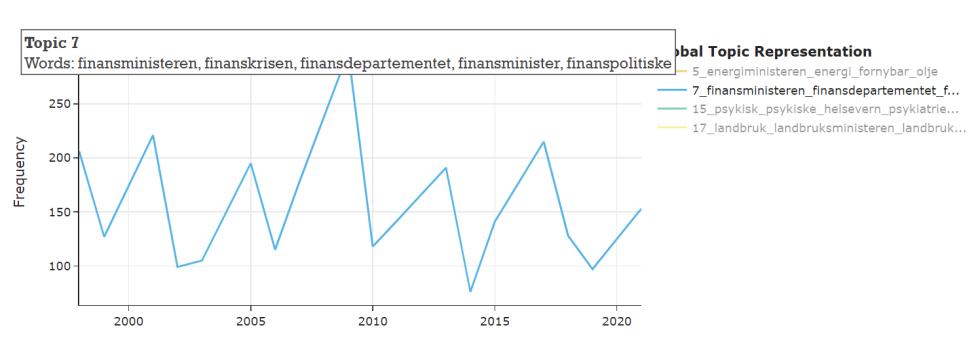


Figure 5.38: Dynamic topic modelling visualization. Showing a single topic#7, highlighted at the year of 2008, where we can see the topic words.

5.7 Experiment 6 - Automatic Topic Labelling Experiment

The goal of this experiment was to test the NETL and BERTopic-AL methods and see how they would perform. Would the generated topic labels make any sense and how would the two methods compare against each other? To help answer these questions we included two tasks in experiment 3. Additionally, we will be using our expert domain knowledge from working on the dataset to evaluate if the topic labels make sense.

5.7.1 Experimental Setup

For this experiment, we will be using the topic model BERTopic with all-roberta-large-v1. The dataset that we will be using is the NPM-stopwords. We will be producing automatic topic labels for the first eight topics using two methods: the

	Topic	Count	Name
0	-1	148805	-1_statsråden_kr_pst_000
1	0	8531	0_replikkordskifte_replikkordsskifte_replikksk...
2	1	5150	1_transportplan_nasjonal_transport_trafikken
3	2	4973	2_omme_replikkordskiftet_replikordskiftet_repl...
4	3	3957	3_partier_partiene_partiet_senterpartiets
...	...	...	...
3075	3074	10	3074_11_12_sakene_behandlet
3076	3075	10	3075_voteringsforklaring_voteringen_xxvi_xxviii
3077	3076	10	3076_maritime_næringsklynger_næringene_kontrah...
3078	3077	10	3077_travel_jordmødre_reiseutgifter_utviklede
3079	3078	10	3078_inspirerte_treminutters_innlegg_represent...

3080 rows × 3 columns

Figure 5.39: The figure shows the number of documents or count, per topic. The topic words are called name and are separated by _.

	Topic	Count	Name
0	-1	10	-1_yesss_mitthode_kapasitetsbyggingstiltak_jepp
1	0	8547	0_replikkordskifte_replikkordsskifte_selvassur...
2	1	5520	1_transportplan_nasjonal_trafikken_transport
3	2	4989	2_omme_replikkordskiftet_replikordskiftet_joys...
4	3	4913	3_partier_partiene_partiet_senterpartiets
...	...	...	...
3075	3074	30	3074_11_kapitla_miljøpakke_12
3076	3075	13	3075_xxiv_voteringstemaet_xxii_xvi
3077	3076	36	3076_visjon_næringene_næringsklynger_maritime
3078	3077	48	3077_travel_jordmødre_travelt_reiseutgifter
3079	3078	26	3078_treminutters_inspirerte_læretørste_lovsiden

3080 rows × 3 columns

Figure 5.40: The figure shows the number of documents or count, per topic after reducing outliers. The topic words are called name and are separated by _.

NETL method and BERTopic method. Further explanation of the methods can be seen in Section 4.5.

For the NETL method we will be training the Doc2Vec model on 1 epoch and Word2Vec model on 1 epoch as well.

Requirements for BERTopic method

1. Eight somewhat coherent topics
2. Most representative document for each topic
3. Prompt for ChatGPT or GPT3.5 API

Requirements for NETL method

1. Norwegian Wikipedia data dump
2. Word2Vec and Doc2Vec models trained on Wikipedia dump
3. Eight somewhat coherent topics

We used the following BERTopic prompt:

We have used topic modelling to generate topics. We now want to generate topic labels for these topics. A topic label is a descriptive label that best captures the meaning of the topic.

The topic is described by the following topicwords in order of importance: <TOPICWORDS>. The most representative document for the topic is <DOCUMENT>.

Based on the information above, extract a short topic label of about three words in Norwegian.

The process of generating topic labels for the NETL method was to copy the topics to a file that was used as input for the *get_labels.py* file. The method generated 19 candidate labels and based on those candidate labels, three topic labels. We did not find any good ways to combine the topic labels, therefore we mostly focused on the first topic label, however they are included in the results.

THe process of generating topics for the BERTopic automatic topic labelling method was to find the most representative document for each topic respectively. By using the document id, we then found the raw document which we used as input to ChatGPT with the prompt specified. The BERTopic method produced topic labels of varying sizes, but a maximum of three words not including binding words.

Table 5.23: Automatic Topic Labelling Methods Compared

Topic	NETL	BERTopic
0	Oljeindustri, tungindustri, gasskraft	Fornybar energi og industrien i Norge
1	skole, lærere, skolefritidsordning	Utdanningstiltak for minoritetsspråklige
2	vetorett, svalbardtraktaten, gjennopptagelse	Statsrådens svar
3	kollektivtransport, afis, kommuneplan	Nasjonal transportplan
4	kommune, kommunen, samkommune	Kommunenes økonomi
5	sv, senterpartiet, arbeiderpartiet	Samarbeid i norsk politikk
6	finanspolitikk, finansdepartement, bankkrise	Regulering av finanssektoren
7	børsnotering, kings_bay, voldsoffererstatning	Forskningspolitikk i Norge

5.7.2 Discussion

Table 5.24 presents the topics with their respective topic words, while Figure 5.26 provides a wordcloud representation of the topics.

Table 5.24: Numbered topics along with their topic words, that were used for the automatic topic labelling methods.

Topic	Topic words
0	energi, regjeringen, gasskraftverk, norge, kraft, industri, olje, fornybar, land, industrien
1	skolen, skole, elever, utdanning, elevene, videregående, studenter, lærere, forskning, opplæring
2	statsråden, statsministeren, stortinget, spørsmål, svaret, takker, regjeringen, saken, gjelder, spørsmålet
3	statsråden, transportplan, kr, nasjonal, veier, jernbane, jernbanen, trafikken, bergen, gjelder
4	kommunene, kommunane, kommuner, kommune, kommunar, kommunen, kommunale, staten, statsråden, kommunale
5	senterpartiet, arbeiderpartiet, folkeparti, kristelig, fremskrittspartiet, sv, representanten, forslag, partiene, politikk
6	finansministeren, finanskrisen, finansnæringen, banker, pst, husbanken, statens, staten, økonomi, regjeringen
7	kr, mill, pst, regjeringen, gardermobanen, forskning, as, fremskrittspartiet, arbeiderpartiet, 000

In Table 5.23, we present the results of the different automatic topic labelling methods tested, alongside the topic numbers. The two columns show the outcomes obtained from each method.

The NETL method shows mixed results. In some cases, the generated labels appear appropriate and coherent, while in others, they seem to be a jumble of words without clear reasoning behind them. It is important to note that the method has limited knowledge of the domain and the specific documents from which the topics are derived. As topics can be generated from different documents, different labels may be desired.

On the other hand, the BERTopic method performs well, particularly with coherent topics. The labels generated by this method tend to make sense and align with the provided documents. The short phrases produced by BERTopic are generally effective and meaningful. However, for less coherent topics, the method still manages to generate more general labels.

Overall, the BERTopic method demonstrates better performance, especially in terms of coherence and relevance to the documents. While both methods have their strengths and limitations, the results suggest that BERTopic leverages the information from the documents to generate topic labels.

Regarding the user ratings in Figure 5.22, the NETL method received an average rating of 2 out of 3, while the BERTopic method obtained an average rating of 2.3 out of 3. It is important to consider the limited number of responses when interpreting these results. Nonetheless, these insights provide valuable information on what makes a topic label useful.

The ratings are influenced by the document summary presented to the users. This is due to the summaries creating an expectation of what the topic should be con-

tained, without taking into account that the topic represents more than one document. generating a topic label for a single document topic is a different task than topic modelling itself. This aspect should be taken into account when considering the task and rating.

From a qualitative standpoint, the BERTopic labels are generally preferred. However, whether they are better representations than the wordclouds is debatable. It could be beneficial to expand the label length to form a descriptive sentence and provide more value to the user. Instead of using only three words, a sentence explaining the topic could be employed. Additionally, including more documents in the topic could enhance its representation.

Overall, the BERTopic method demonstrates better performance than NETL, in generating relevant and coherent topic labels.

Limitations

One limitation of our approach, as discussed in Section 4.5.2, is the rate limitation imposed on the manual evaluation process. Due to this limitation, the evaluation had to be conducted manually using ChatGPT, which was time-consuming. However, this manual process could be scripted and automated, serving as a proof of concept for future research.

It is important to note that the topics themselves are generated by topic modelling techniques, which are efficient in handling large datasets. The automatic topic labelling process, on the other hand, serves as a fine-tuning step or post-processing method to extract more information from the topics. If we were to solely rely on the ChatGPT alternative for topic modelling, we would face the challenge of rate limitation and significantly increased processing times.

Answering the research questions

Overall through the exploration of BERTopic and NETL for automatic topic labelling, along with the results and the ensuing discussion, we have in this subsection answered RQ2: **What automatic topic labelling techniques exist and how do they perform on Norwegian transcribed parliamentary speeches?**.

Chapter 6

Conclusion and Future Work

This chapter concludes the thesis. In Section 6.1 we summarize our findings, detailing what each Chapter of the thesis consists of. We propose future work in Section 6.2.

6.1 Conclusion

Topic modelling is a powerful technique for uncovering latent themes and extracting meaningful insights from textual data. In this thesis, we present an in-depth exploration of different topic modelling techniques and automatic topic labelling approaches. We present an evaluation framework and apply it to several use cases to showcase its viability.

The thesis is structured into several chapters. In Chapter 1, we provide the background and motivations behind the work. In Chapter 2, we provide a comprehensive background, covering key concepts and metrics relevant to topic modelling. Chapter 3 presents an extensive literature review, where we delve into various topic modelling models to identify suitable techniques for our study.

In Chapter 4, we outline the methodology employed in multiple experiments conducted throughout the research. As well as presenting a novel method to qualitatively evaluate topic modelling results through our evaluation framework.

In Chapter 5, we detail the process of setting up each experiment, before presenting and discussing the results.

Experiment 1 serves as a preliminary investigation, comparing the performance of Top2Vec, BERTopic, and LDA on different datasets, namely NPM-raw, NPM-basic, and NPM-stopwords.

Experiment 2 focuses on the exploration of different embedding models in conjunction with Top2Vec, and BERTopic, aiming to understand the impact of varying embedding models on topic modelling results. The experiment also served as a practical use case for the evaluation framework.

In Experiment 3, we shift our focus to user testing, assessing the quality of topics generated by different models and evaluating the effectiveness of automatically generated topic labels. This user-centred evaluation provides valuable insights into the strengths and limitations of topic modelling techniques.

Experiment 4 examines the utilization of an evaluation framework on topic modelling results as well as the effects of random sampling. The experiment offers detailed ratings using the evaluation framework, to serve as a guide for practical applications. This experiment sheds light on the framework's intended usage and its effectiveness in evaluating topic models.

In Experiment 5, we explore the performance of topic modelling techniques on large datasets (NPL), with a particular emphasis on dynamic topic modelling and the reduction of outlier topics. This experiment showcases the applicability of topic modelling approaches in handling extensive textual data.

Finally, Experiment 6 presents two distinct automatic topic labelling techniques: NETL and BERTopic. We compare their performance and highlight their respective strengths and limitations in the context of topic labelling.

Through these experiments and analyses, we gain valuable insights into the performance, strengths, and limitations of different topic modelling techniques and automatic topic labelling methods. Through the deeper understanding of the evaluation of topic modelling results gained, we created an evaluation framework that we utilized in experiments 2, 4 and 5.

6.2 Future Work

In this thesis, we have mentioned various limitations and possibilities for future work in the experiments. The four primary areas of future work area: further data processing, topic modelling optimization, improving the evaluation framework and further developing the automatic topic labelling methods.

As most of the experiments conducted were done using NPM, future work could potentially be to conduct similar experiments, but using a different sample of NPL. A possible way to create a new sample (NPM2) would be to take a random sample based on the dates across all years, to get a full view of the political landscape over time. Another option would be to choose one, or a couple of years and use all

documents from that year. Deciding on a specific year would probably have the benefit of having more documents per topic, but then fewer topics as a consequence of that, while overall years would include many topics unless the sample was skewed randomly towards any single topics. The upside of hand-crafting a sample would be that more of the biases of the dataset would be known beforehand and could more easily be taken into account.

Hand-crafting a sample offers the advantage of having more control over the biases present in the dataset, allowing for a more informed consideration of these biases in the analysis. This approach would facilitate a deeper understanding of the dataset and enable the authors to address potential biases more effectively.

Another potential future work would be to conduct more extensive preprocessing experiments on either a sample of NPL or the whole dataset. Seeing as the only preprocessing rules properly tested were stopword removal, other preprocessing rules such as statistical preprocessing through TF-IDF or NLP preprocessing such as lemmatization or stemming. Additionally as noted in Section 4.1 extending the stopword list could potentially improve the results. A more extensive study of the dataset would have to be conducted to identify the domain-specific stopwords, and perhaps take inspiration from the data preprocessing conducted in Hoffman *et al.* [27], where they extended the stopword list to include the titles of officials, which was something that repeated itself in our results, "statsråden" (the Minister of State)

For topic modelling, other techniques mentioned in Chapter 3, such as TopClus [46] or the method specifie in Sia *et al.* [55] could be tested. Otherwise, different embedding models could be tried with BERTopic.

For the evaluation framework, further testing would be beneficial and eventual improvements to it such as more categories and more refined criteria.

For the automatic topic labelling methods, it would be interesting to test the BERTopic representation model way of generating topic labels, providing the text generation model with more documents than one.

Bibliography

[1] W. Abdulaziz, M. M. Ameen and B. I. Ahmed, 'An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges,' *ResearchGate*, pp. 200–204, Jun. 2019. DOI: 10.1109/IEC47844.2019.8950616.

[2] C. C. Aggarwal, A. Hinneburg and D. A. Keim, 'On the Surprising Behavior of Distance Metrics in High Dimensional Space,' in *Database Theory — ICDT 2001*, Berlin, Germany: Springer, Oct. 2001, pp. 420–434. DOI: 10.1007/3-540-44503-X_27.

[3] D. J. Aldous, 'Exchangeability and related topics,' in *École d'Été de Probabilités de Saint-Flour XIII — 1983*, P. L. Hennequin, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 1–198, ISBN: 978-3-540-39316-0.

[4] D. Angelov, 'Top2Vec: Distributed Representations of Topics,' *arXiv*, Aug. 2020. DOI: 10.48550/arXiv.2008.09470. eprint: 2008.09470.

[5] R. Bellman, *Dynamic Programming*, 1st ed. Princeton, NJ, USA: Princeton University Press, 1957.

[6] S. Bhatia, J. H. Lau and T. Baldwin, 'Automatic Labelling of Topics with Neural Embeddings,' *arXiv*, Dec. 2016. DOI: 10.48550/arXiv.1612.05340. eprint: 1612.05340.

[7] D. M. Blei and J. D. Lafferty, 'Dynamic topic models,' in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 113–120, ISBN: 1595933832. DOI: 10.1145/1143844.1143859. [Online]. Available: https://doi.org/10.1145/1143844.1143859.

[8] D. M. Blei, A. Y. Ng and M. I. Jordan, 'Latent dirichlet allocation,' *J. Mach. Learn. Res.*, vol. 3, no. null, pp. 993–1022, Mar. 2003, ISSN: 1532-4435. DOI: 10.5555/944919.944937.

[9] P. Bojanowski, É. Grave, A. Joulin and T. Mikolov, 'Enriching Word Vectors with Subword Information,' *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017. DOI: 10.1162/tacl_a_00051.

[10] G. Bouma, 'Normalized (Pointwise) Mutual Information in Collocation Extraction,' *Proceedings of the Biennial GSCL Conference 2009*, Jan. 2009. [Online]. Available: https://www.researchgate.net/publication/267306132_Normalized_Pointwise_Mutual_Information_in_Collocation_Extraction.

[11] J. L. Boyd-Graber, D. M. Blei and X. Zhu, 'A Topic Model for Word Sense Disambiguation.,' *ResearchGate*, pp. 1024–1033, Jan. 2007. [Online]. Available: https://www.researchgate.net/publication/221013017_A_Topic_Model_for_Word_Sense_Disambiguation.

[12] T. Brants, A. C. Popat, P. Xu, F. J. Och and J. Dean, 'Large language models in machine translation,' in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 858–867. [Online]. Available: https://aclanthology.org/D07-1090.

[13] D. Cai, X. Wang and X. He, 'Probabilistic dyadic data analysis with local and global consistency,' in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML'09)*, 2009, pp. 105–112.

[14] D. Card, P. Henderson, U. Khandelwal, R. Jia, K. Mahowald and D. Jurafsky, 'With little power comes great responsibility,' in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 9263–9274. DOI: 10.18653/v1/2020.emnlp-main.745. [Online]. Available: https://aclanthology.org/2020.emnlp-main.745.

[15] J. Chang, S. Gerrish, C. Wang, J. Boyd-graber and D. Blei, 'Reading tea leaves: How humans interpret topic models,' in *Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams and A. Culotta, Eds., vol. 22, Curran Associates, Inc., 2009. [Online]. Available: https://proceedings.neurips.cc/paper/2009/file/f92586a25bb3145facd64ab20fd554ff-Paper.pdf.

[16] Z. Chen, A. Mukherjee, B. Liu, M. Hsu and R. Ghosh, 'Discovering Coherent Topics Using General Knowledge,' *International Conference on Information and Knowledge Management, Proceedings*, pp. 209–218, Oct. 2013. DOI: 10.1145/2505515.2505519.

[17] R. Churchill and L. Singh, 'textPrep: A Text Preprocessing Toolkit for Topic Modeling on Social Media Data [textPrep: A Text Preprocessing Toolkit for Topic Modeling on Social Media Data],' *Proceedings of the 10th International Conference on Data Science, Technology and Applications*, Jan. 2021. DOI: 10.5220/0010559000600070.

[18] 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, Sep. 1990, ISSN: 0002-8231. DOI: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9. [Online]. Available: https://doi.org/10.1002/(SICI)1097-4571(199009)41:6%3C391::AID-ASI1%3E3.0.CO;2-9.

[19] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,' *arXiv*, Oct. 2018. DOI: 10.48550/arXiv.1810.04805. eprint: 1810.04805.

[20] A. B. Dieng, F. J. R. Ruiz and D. M. Blei, 'The Dynamic Embedded Topic Model,' *arXiv*, Jul. 2019. DOI: 10.48550/arXiv.1907.05545. eprint: 1907.05545.

[21] A. B. Dieng, F. J. R. Ruiz and D. M. Blei, 'Topic Modeling in Embedding Spaces,' *arXiv*, Jul. 2019. DOI: 10.48550/arXiv.1907.04907. eprint: 1907.04907.

[22] T. Erjavec, M. Ogrodniczuk, P. Osenova, N. Ljubešić, K. Simov, A. Pančur, M. Rudolf, M. Kopp, S. Barkarson, S. Steingrímsson, Ç. Çöltekin, J. de Does, K. Depuydt, T. Agnoloni, G. Venturi, M. C. Pérez, L. D. de Macedo, C. Navarretta, G. Luxardo, M. Coole, P. Rayson, V. Morkevičius, T. Krilavičius, R. Darǵis, O. Ring, R. van Heusden, M. Marx and D. Fišer, 'The parlamint corpora of parliamentary proceedings,' *Language Resources and Evaluation*, vol. 57, no. 1, pp. 415–448, Mar. 2023, ISSN: 1574-0218. DOI: 10.1007/s10579-021-09574-0. [Online]. Available: https://doi.org/10.1007/s10579-021-09574-0.

[23] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, 'A density-based algorithm for discovering clusters in large spatial databases with noise,' in *KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Aug. 1996, pp. 226–231. DOI: 10.5555/3001460.3001507.

[24] N. Fox, 'Helping you along your Search journeys,' *Google*, Sep. 2018. [Online]. Available: https://www.blog.google/products/search/helping-you-along-your-search-journeys.

[25] K. Grieser, T. Baldwin, F. Bohnert and L. Sonenberg, 'Using ontological and document similarity to estimate museum exhibit relatedness,' *J. Comput. Cult. Herit.*, vol. 3, no. 3, pp. 1–20, Feb. 2011, ISSN: 1556-4673. DOI: 10.1145/1921614.1921617.

[26] M. Grootendorst, 'BERTopic: Neural topic modeling with a class-based TF-IDF procedure,' *arXiv*, Mar. 2022. DOI: 10.48550/arXiv.2203.05794. eprint: 2203.05794.

[27] K. Hofmann, A. Marakasova, A. Baumann, J. Neidhardt and T. Wissik, 'Comparing Lexical Usage in Political Discourse across Diachronic Corpora,' *ACL Anthology*, pp. 58–65, May 2020. [Online]. Available: https://aclanthology.org/2020.parlaclarin-1.11.

[28] T. Hofmann, 'Probabilistic latent semantic indexing,' pp. 50–57, Aug. 1999. DOI: 10.1145/312624.312649.

[29] A. Hoyle, P. Goel, D. Peskov, A. Hian-Cheong, J. Boyd-Graber and P. Resnik, 'Is Automated Topic Model Evaluation Broken?: The Incoherence of Coherence,' *arXiv*, Jul. 2021. DOI: 10.48550/arXiv.2107.02173. eprint: 2107.02173.

[30] T.-C. Huang, C.-H. Hsieh and H.-C. Wang, 'Automatic meeting summarization and topic detection system,' *Data Technologies and Applications*, vol. 52, no. 1, Apr. 2018, ISSN: 2514-9288. DOI: 10.1108/DTA-09-2017-0062.

[31] *Introducing our Hybrid lda2vec Algorithm | Stitch Fix Technology – Multithreaded*, [Online; accessed 29. Nov. 2022], Nov. 2022. [Online]. Available: https://multithreaded.stitchfix.com/blog/2016/05/27/lda2vec/#topic=38%5C%CE%5C%BB=1&term=.

[32] W. Kou, F. Li and T. Baldwin, 'Automatic labelling of topic models using word vectors and letter trigram vectors,' in *Information Retrieval Technology*, G. Zuccon, S. Geva, H. Joho, F. Scholer, A. Sun and P. Zhang, Eds., Cham: Springer International Publishing, 2015, pp. 253–264, ISBN: 978-3-319-28940-3.

[33] P. E. Kummervold, J. De la Rosa, F. Wetjen and S. A. Brygfjeld, 'Operationalizing a national digital library: The case for a Norwegian transformer model,' in *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, Reykjavik, Iceland (Online): Link"oping University Electronic Press, Sweden, 2021, pp. 20–29. [Online]. Available: https://aclanthology.org/2021.nodalida-main.3.

[34] J. H. Lau, K. Grieser, D. Newman and T. Baldwin, 'Automatic labelling of topic models,' in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 1536–1545. [Online]. Available: https://aclanthology.org/P11-1154.

[35] J. H. Lau, D. Newman and T. Baldwin, 'Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality,' in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden: Association for Computational Linguistics, Apr. 2014, pp. 530–539. DOI: 10.3115/v1/E14-1056. [Online]. Available: https://aclanthology.org/E14-1056.

[36] J. H. Lau, D. Newman and T. Baldwin, 'Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality,' *ACL Anthology*, pp. 530–539, Apr. 2014. DOI: 10.3115/v1/E14-1056.

[37] J. H. Lau, D. Newman, S. Karimi and T. Baldwin, 'Best Topic Word Selection for Topic Labelling.,' *ResearchGate*, vol. 2, pp. 605–613, Jan. 2010. [Online]. Available: https://www.researchgate.net/publication/221102629_Best_Topic_Word_Selection_for_Topic_Labelling.

[38] Q. V. Le and T. Mikolov, 'Distributed Representations of Sentences and Documents,' *arXiv*, May 2014. DOI: 10.48550/arXiv.1405.4053. eprint: 1405.4053.

[39] W. Li and A. McCallum, 'Pachinko allocation: Dag-structured mixture models of topic correlations,' in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 577–584, ISBN: 1595933832. DOI: 10.1145/1143844.1143917. [Online]. Available: https://doi.org/10.1145/1143844.1143917.

[40] Q. Liang, X. Zheng, M. Wang, H. Chen and P. Lu, 'Optimize Recommendation System with Topic Modeling and Clustering,' *ResearchGate*, pp. 15–22, Nov. 2017. DOI: 10.1109/ICEBE.2017.13.

[41] L. Liu, L. Tang, W. Dong, S. Yao and W. Zhou, 'An overview of topic modeling and its current applications in bioinformatics,' *Springerplus*, vol. 5, no. 1, p. 1608. Sep. 2016, ISSN: 2193-1801. DOI: 10.1186/s40064-016-3252-8. eprint: 27652181.

[42] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, 'RoBERTa: A Robustly Optimized BERT Pretraining Approach,' *arXiv*, Jul. 2019. DOI: 10.48550/arXiv.1907.11692. eprint: 1907.11692.

[43] A. Maćkiewicz and W. Ratajczak, 'Principal components analysis (PCA),' *Comput. Geosci.*, vol. 19, no. 3, pp. 303–342, Mar. 1993, ISSN: 0098-3004. DOI: 10.1016/0098-3004(93)90090-R.

[44] C. Malzer and M. Baum, 'A Hybrid Approach To Hierarchical Density-based Cluster Selection,' *arXiv*, Nov. 2019. DOI: 10.1109/MFI49285.2020.9235263. eprint: 1911.02282.

[45] L. McInnes, J. Healy and J. Melville, 'UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction,' *arXiv*, Feb. 2018. DOI: 10.48550/arXiv.1802.03426. eprint: 1802.03426.

[46] Y. Meng, Y. Zhang, J. Huang, Y. Zhang and J. Han, 'Topic Discovery via Latent Space Clustering of Pretrained Language Model Representations,' in *WWW '22: Proceedings of the ACM Web Conference 2022*, New York, NY, USA: Association for Computing Machinery, Apr. 2022, pp. 3143–3152, ISBN: 978-1-45039096-5. DOI: 10.1145/3485447.3512034.

[47] T. Mikolov, K. Chen, G. Corrado and J. Dean, 'Efficient Estimation of Word Representations in Vector Space,' *arXiv*, Jan. 2013. DOI: 10.48550/arXiv.1301.3781. eprint: 1301.3781.

[48] G. A. Miller, 'Wordnet: A lexical database for english,' *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995, ISSN: 0001-0782. DOI: 10.1145/219717.219748. [Online]. Available: https://doi.org/10.1145/219717.219748.

[49] D. Newman, S. Karimi and L. Cavedon, 'External Evaluation of Topic Models,' *ADCS 2009 - Proceedings of the Fourteenth Australasian Document Computing Symposium*, Jan. 2009. [Online]. Available: https://www.researchgate.net/publication/255602484_External_Evaluation_of_Topic_Models.

[50] D. Newman, J. H. Lau, K. Grieser and T. Baldwin, 'Automatic Evaluation of Topic Coherence.,' *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pp. 100–108, Jan. 2010. [Online]. Available: https://www.researchgate.net/publication/220817098_Automatic_Evaluation_of_Topic_Coherence.

[51] J. Pennington, R. Socher and C. D. Manning, 'GloVe: Global Vectors for Word Representation,' *ACL Anthology*, pp. 1532–1543, Oct. 2014. DOI: 10.3115/v1/D14-1162.

[52] N. Reimers and I. Gurevych, 'Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,' *arXiv*, Aug. 2019. DOI: 10.48550/arXiv.1908.10084. eprint: 1908.10084.

[53] M. R. Rushfeldt, 'Automatic Topic Generation for Broadcasters: Usable Metadata from Topic Models on Systematically Preprocessed TV Subtitles,' Ph.D. dissertation, NTNU, 2022. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3032265.

[54] G. Salton and C. Buckley, 'Term-weighting approaches in automatic text retrieval,' *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, Jan. 1988, ISSN: 0306-4573. DOI: 10.1016/0306-4573(88)90021-0.

[55] S. Sia, A. Dalmia and S. J. Mielke, 'Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too!' In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 1728–1736. DOI: 10.18653/v1/2020.emnlp-main.135. [Online]. Available: https://aclanthology.org/2020.emnlp-main.135.

[56] L. Van Der Maaten and G. Hinton, 'Viualizing data using t-SNE,' *Journal of Machine Learning Research*, vol. 9, no. 2605, pp. 2579–2605, Nov. 2008, ISSN: 1533-7928. [Online]. Available: https://www.researchgate.net/publication/228339739_Viualizing_data_using_t-SNE.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, 'Attention Is All You Need,' *arXiv*, Jun. 2017. DOI: 10.48550/arXiv.1706.03762. eprint: 1706.03762.

[58] D. Xu and Y. Tian, 'A Comprehensive Survey of Clustering Algorithms,' *Ann. Data. Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015, ISSN: 2198-5812. DOI: 10.1007/s40745-015-0040-1.

[59] N. Yang, J. Jo, M. Jeon, W. Kim and J. Kang, 'Semantic and explainable research-related recommendation system based on semi-supervised methodology using BERT and LDA models,' *Expert Syst. Appl.*, vol. 190, p. 116 209, Mar. 2022, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2021.116209.

[60] H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du and W. Buntine, 'Topic Modelling Meets Deep Neural Networks: A Survey,' *arXiv*, Feb. 2021. DOI: 10.48550/arXiv.2103.00498. eprint: 2103.00498.

Appendix A

Additional Material

A.1 NPL Data Processing

Preprocessing the Parlamint dataset caused a lot of issues. First of all the parlamint dataset is in a special format, so a .xsl file which converted from tei to text was used. This left us with a xml file containing the content. This xml file in itself was quite difficult to comprehend just looking at it. Even when using the parser, the format of the text was quite specific, containing a tab and newline. A special function was created which probably only works on this specific dataset where the inputdata is first split on tab, then again split on newline. To iterate through all the files all xml files per year was counted and a dictionary containing this information was used to find for example all 58 files for the year of 1998. Eventually we were able to read all the entries and write to file, but using "utf-8" encoding to write and then read did not work because there were some unknown characters at the end of some entries. Therefore "latin-1" which is an encoding that can handle almost all text, but not always give the correct strings was used. The generated dataset passed the eyetest and after randomly sampling some entries it was decided as acceptable with the notion that through the basic preprocessing all notation marks should be removed and if there was anything left it would be shown in the results of topic modelling.

```
import ElementTree as ET
XSLT = "Extensible Stylesheet Language Transformations"
xsl_format = get .tei to text formatting rules from file
xsl_transformation_format = initialize ET parser with xsl_format
xml_parser = initialize ET.XSLT with xsl_transformation_format

Iterate through all directories from ParlaMint-NO by year:
    for file in directory:
        if file is an xml_file:
            raw_text = parse file with xml_parser
```

```
            text, date = extract text and date from raw_text
            write text and date to output file
```

Appendix B

Experiment 2 - Complete wordcloud samples

Sample from BERTopic-all-miniLM-L12-v2: np_mini_stopwords

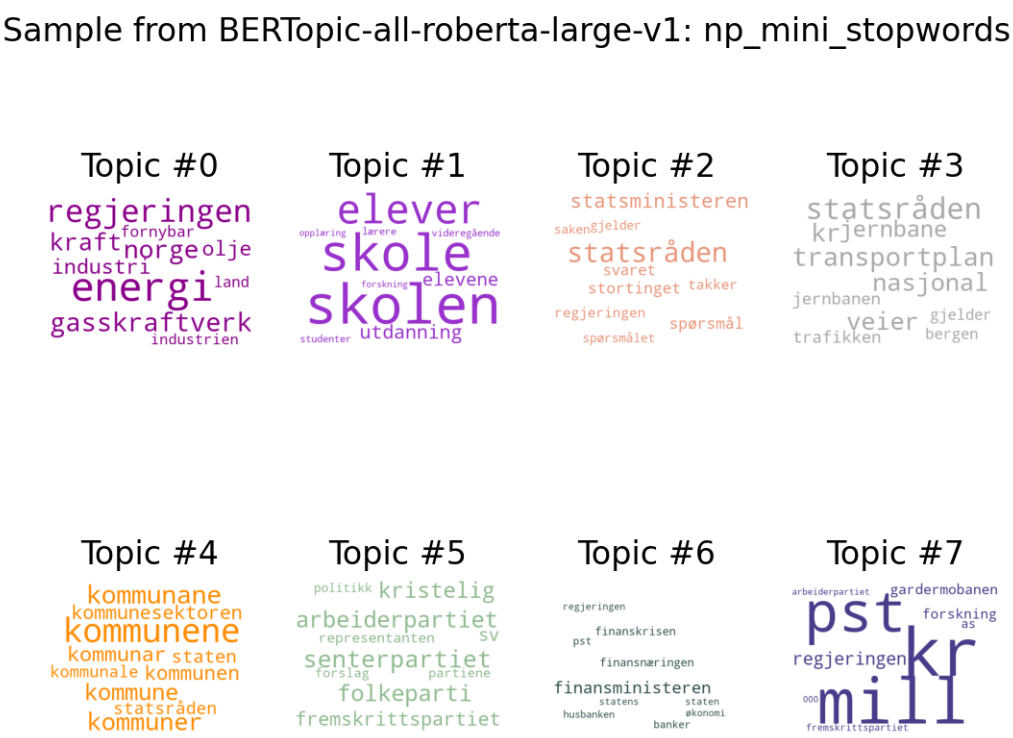


Figure B.1: Additional wordclouds

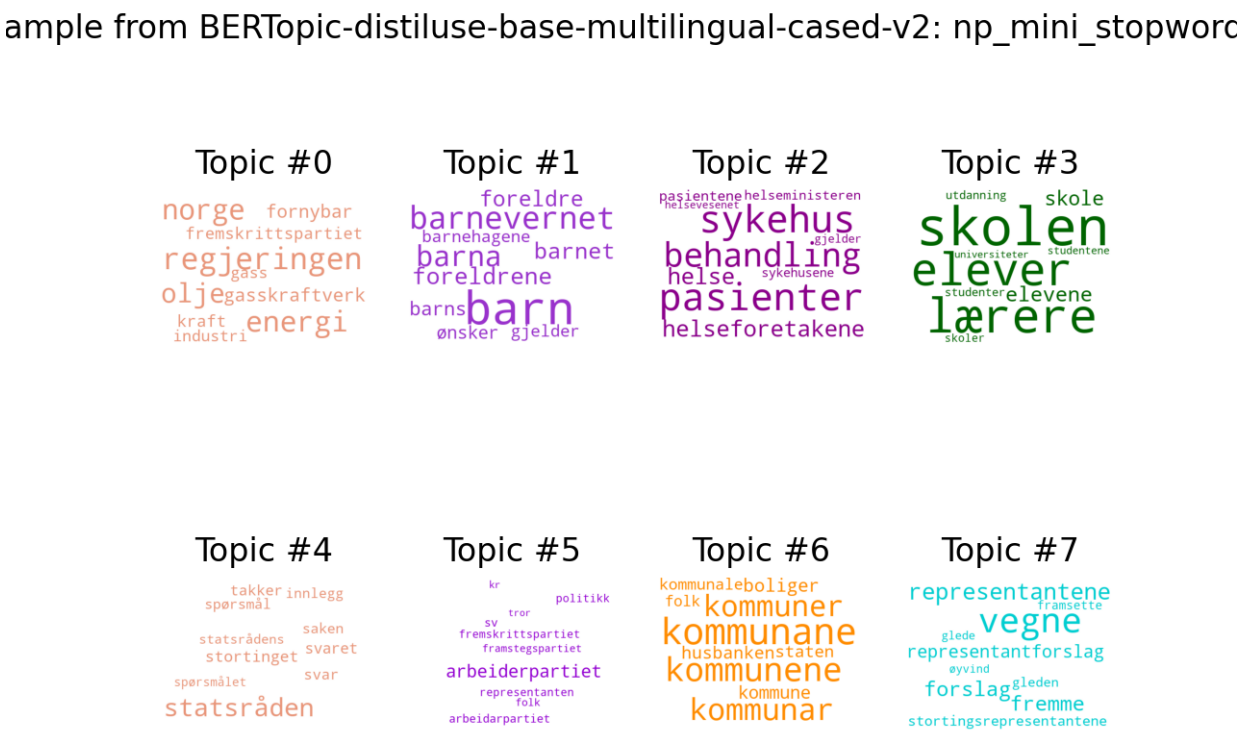


Figure B.2: Additional wordclouds

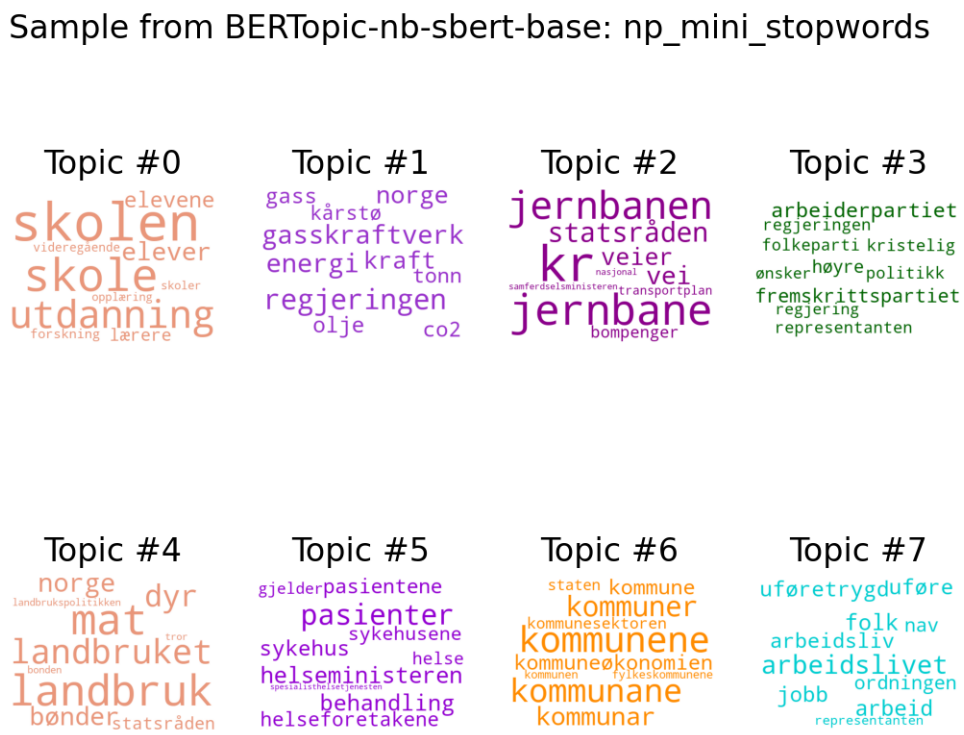


Figure B.3: Additional wordclouds

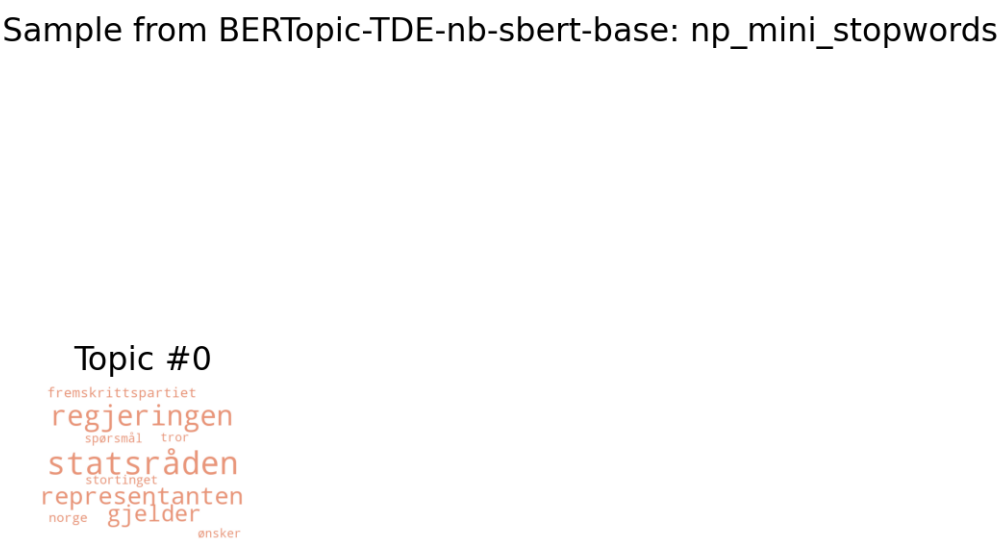


Figure B.4: Additional wordclouds

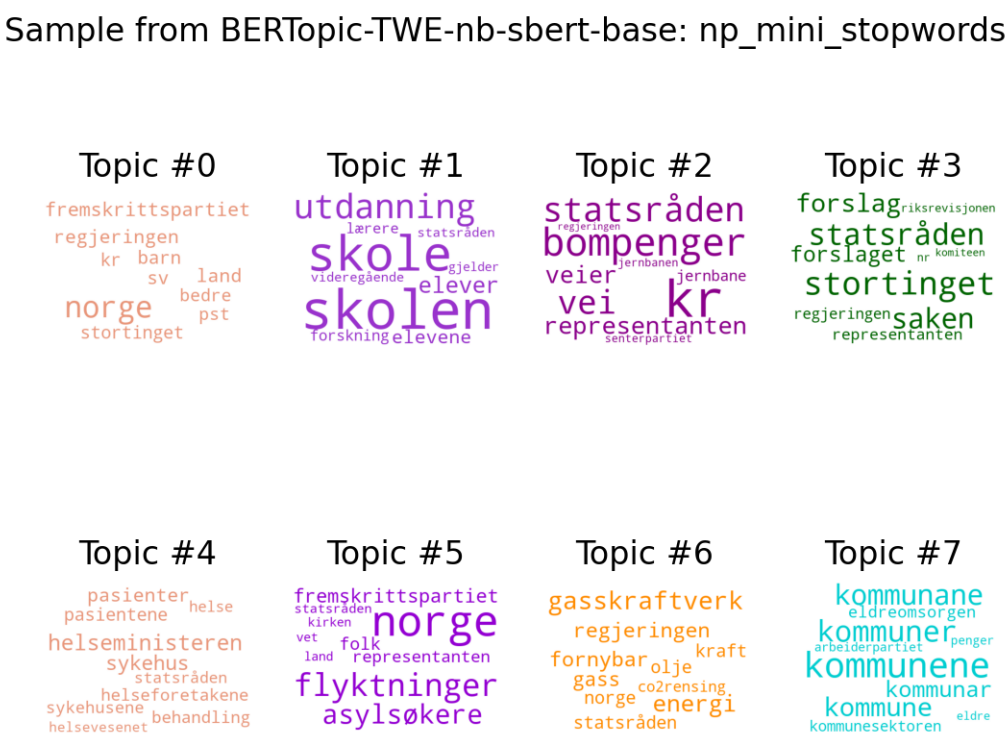


Figure B.5: Additional wordclouds

Figure B.6: Additional wordclouds

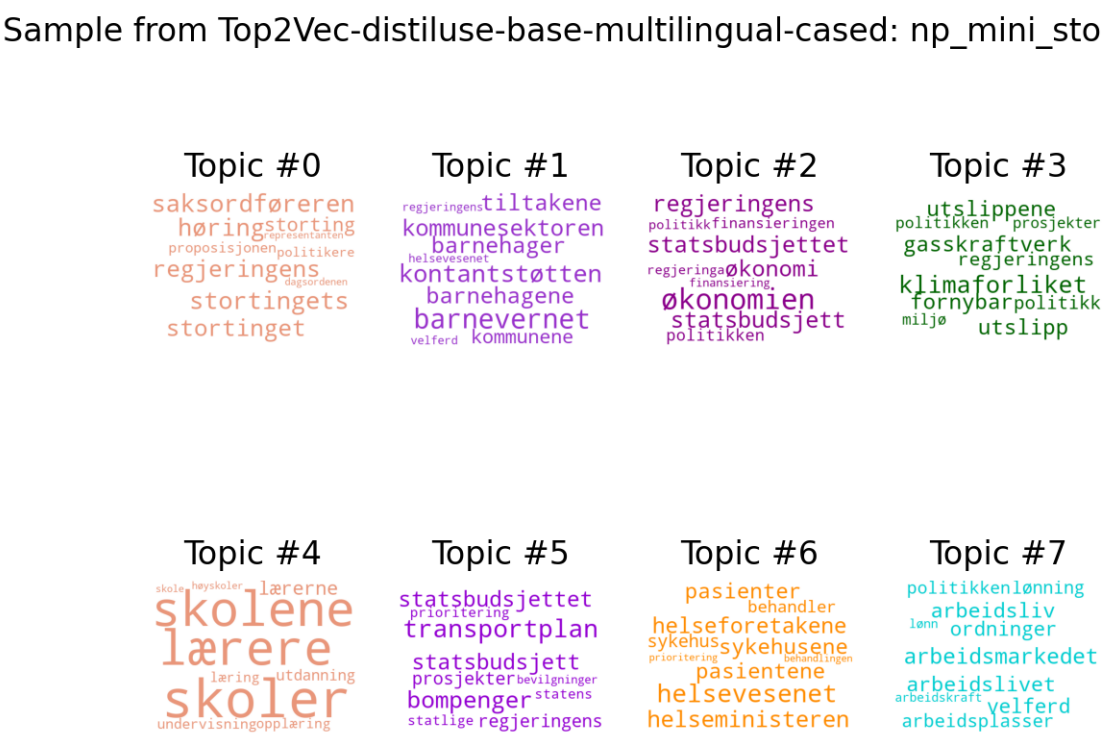


Figure B.7: Additional wordclouds

Figure B.8: Additional wordclouds

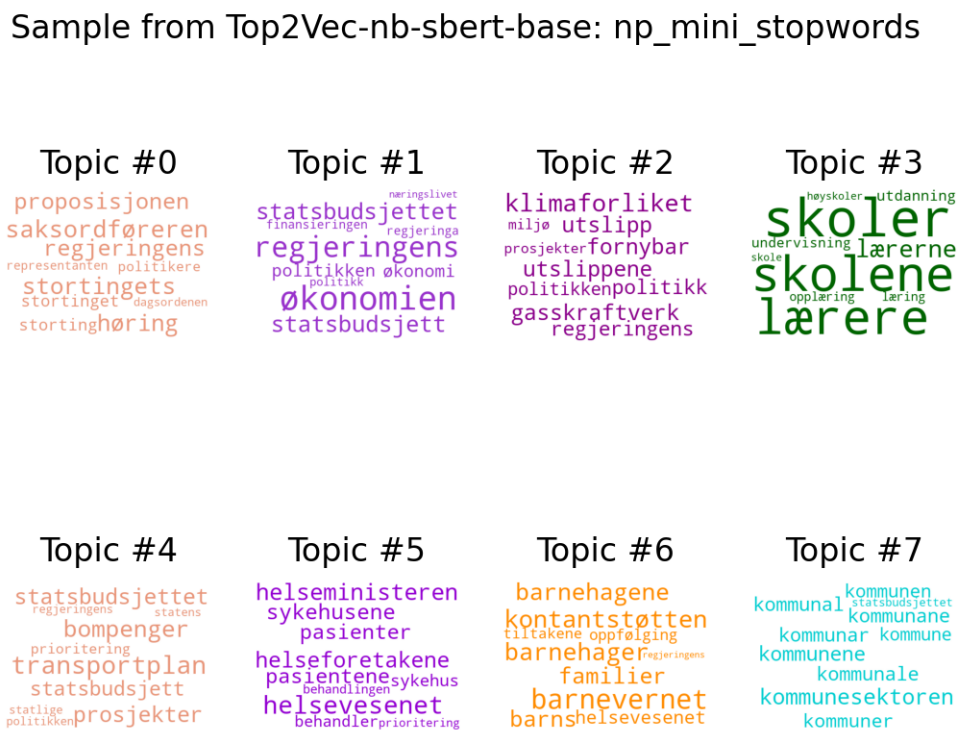


Figure B.9: Additional wordclouds

nple from Top2Vec-universal-sentence-encoder-multilingual: np_mini_stopwo

Figure B.10: Additional wordclouds

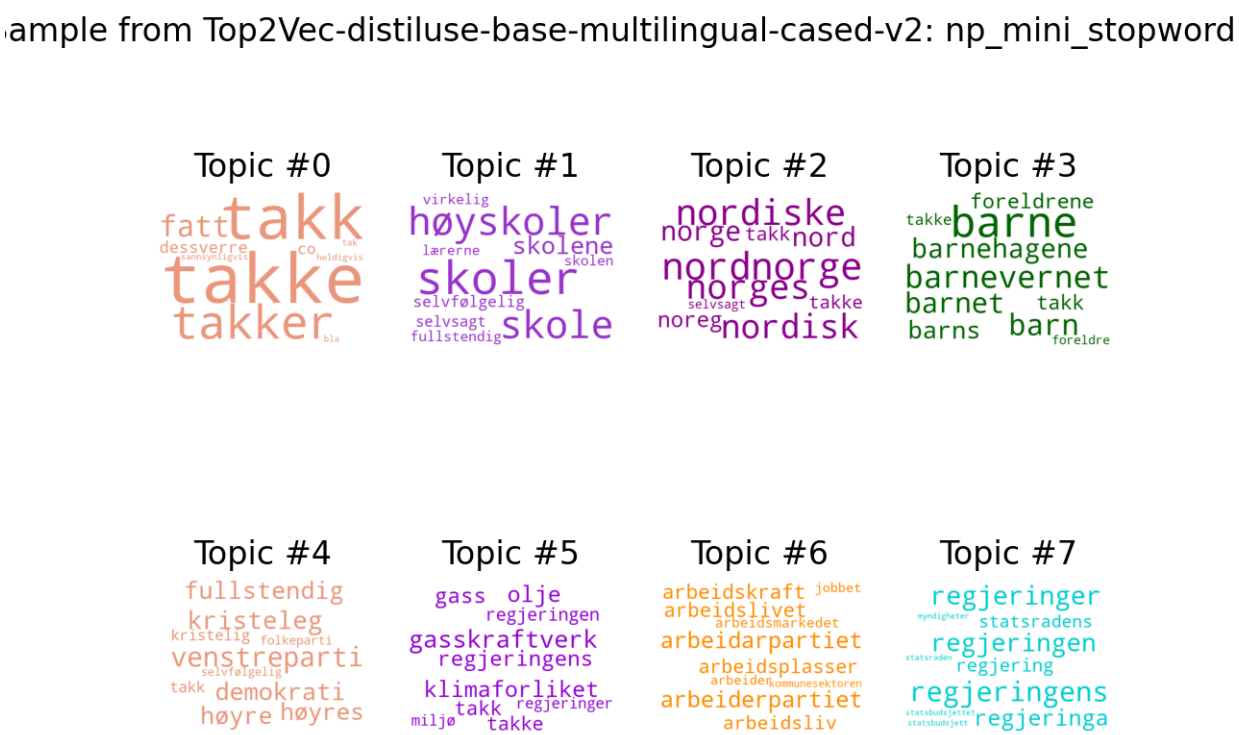


Figure B.11: Additional wordclouds

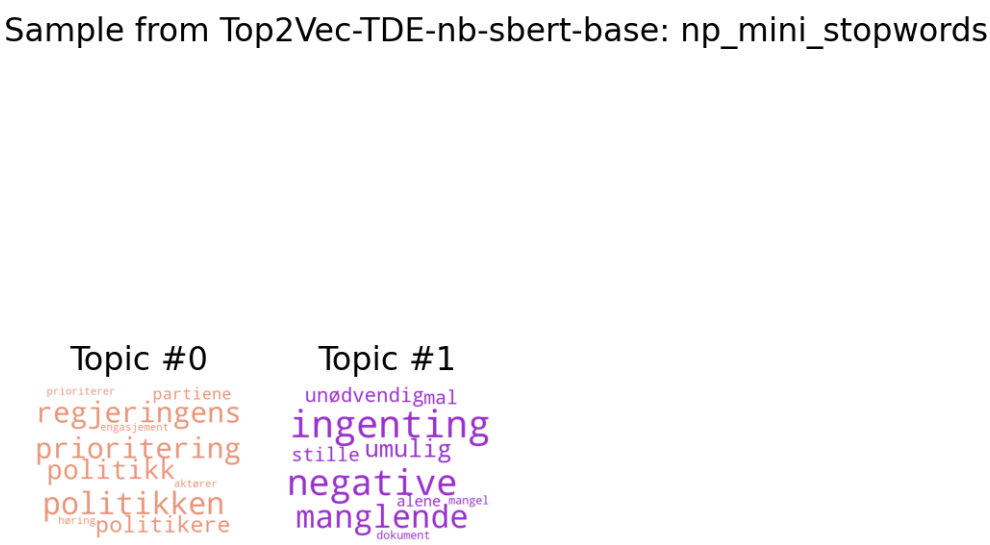


Figure B.12: Additional wordclouds

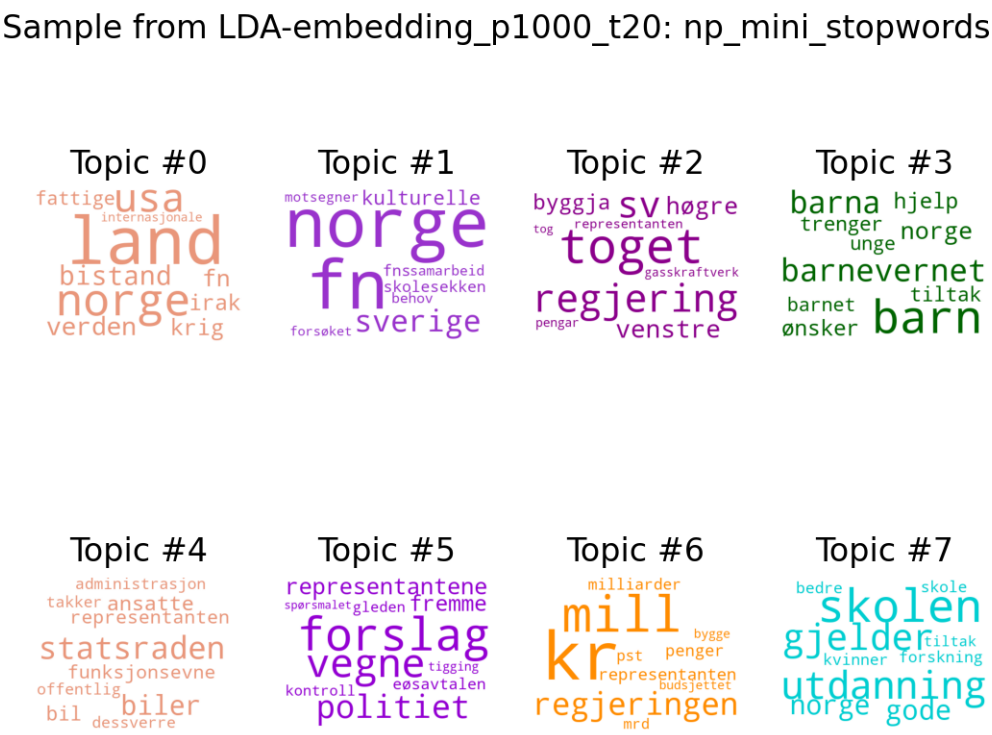


Figure B.13: Additional wordclouds

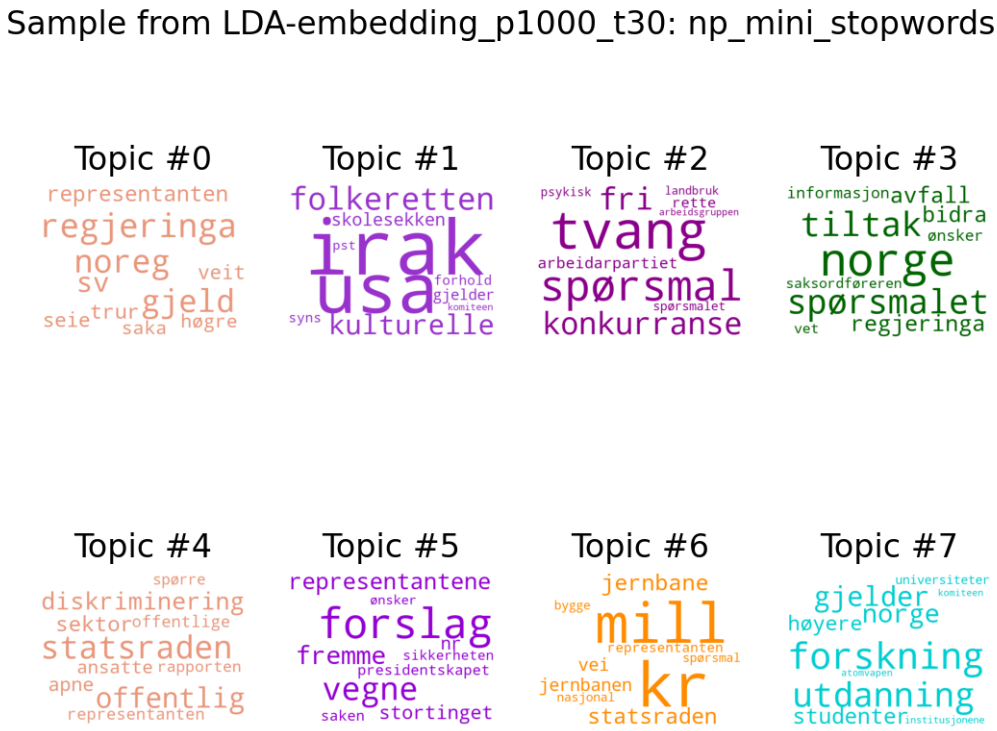


Figure B.14: Additional wordclouds

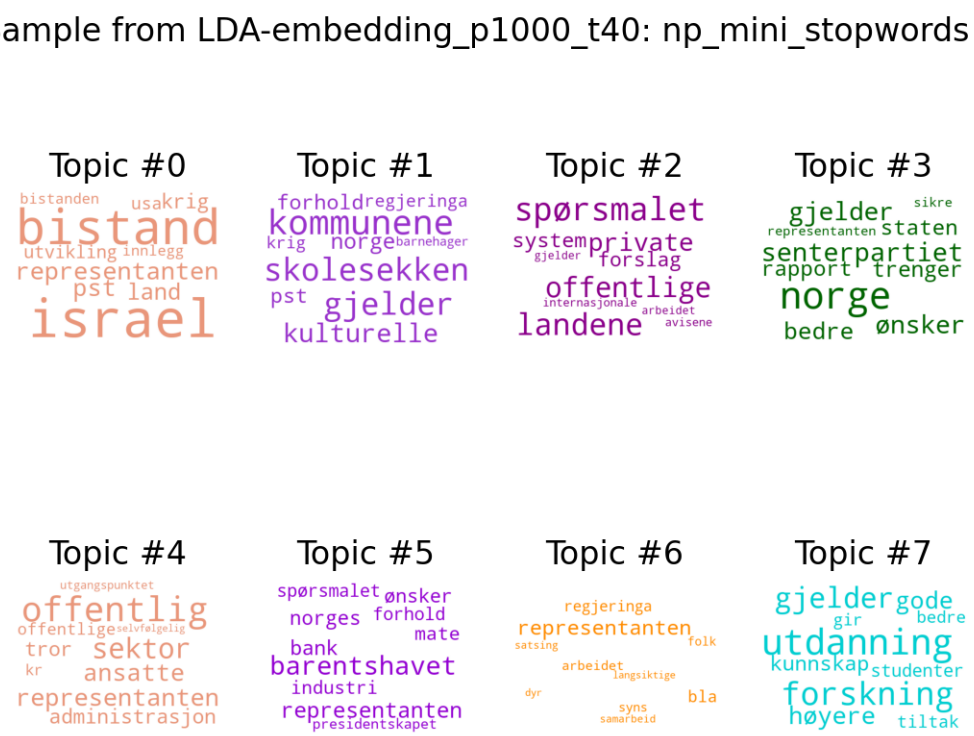
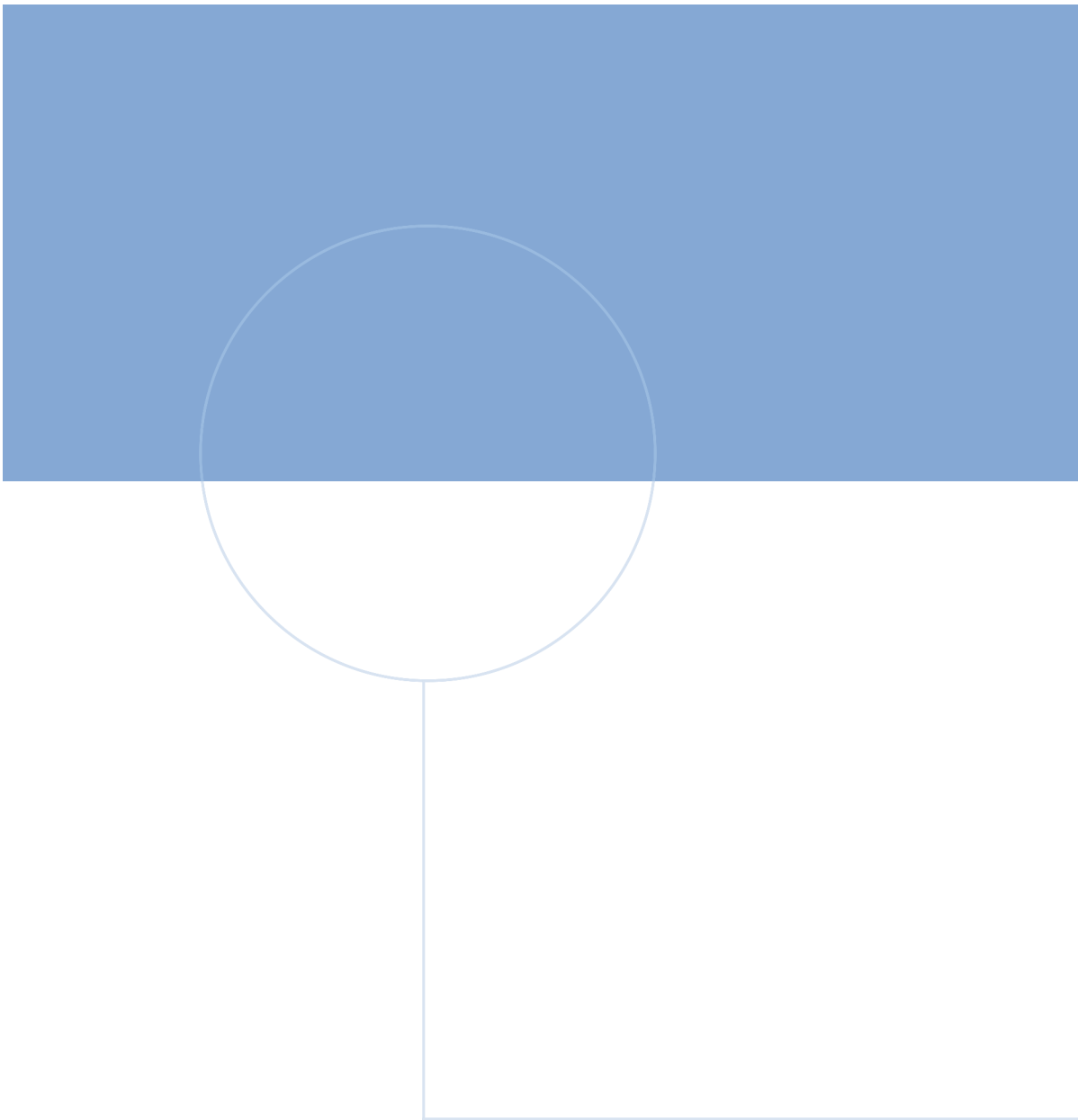


Figure B.15: Additional wordclouds



NTNU

Norwegian University of
Science and Technology