Magnus Reier Rushfeldt

Automatic Topic Generation for Broadcasters:

Usable Metadata from Topic Models on Systematically Preprocessed TV Subtitles

Master thesis, Spring 2021

Data and Artificial Intelligence Group
Department of Computer Science
Faculty of Information Technology and Electrical Engineering

Abstract

At Norsk Rikskringkasting (NRK), Norway's public broadcasting corporation, increasing digitization and changes in how people read news, watch TV and listen to radio lead to new challenges. Tax-paying citizens of Norway (and thus "customers" of NRK's services) demand to find all multimedia content in the form of TV programs, news articles, and radio programs relevant to them at NRK's online platforms with as little effort as possible. All the while, journalists, editors, and publishers at NRK struggle to keep track of what content they have and how to make sure what they publish is relevant and connected to all other related media items they publish. NRK has a high demand for more structured metadata on its content, which would help in all these efforts. Still, they cannot manually create metadata for thousands of multimedia files and need some automatic way of generating this from the files. The data they create must also be understandable and useful for employees.

With the development of powerful new Natural Language Processing (NLP) technology in recent years, many previously complex language tasks can be efficiently and accurately solved, although there is a catch. Most of this technology is developed for English or other major world languages. Even though universal multilingual alternatives exist for some algorithms, for many smaller and resource-constrained languages such as Norwegian, the performance is not on the same level as for English. But modern NLP methods that are extended or adapted to work well for Norwegian can help solve these performance gaps and provide NRK with powerful metadata generation tools. In this thesis, I will therefore contribute with an approach to solving their metadata problem: Automatic generation of metadata from files in the form of topics, each represented as a list of related keywords and an implicit topic in the file. This generation will be performed using Topic Modeling, a form of unsupervised learning where hidden topics in text documents are identified and represented using words from the documents. I use NRK's new and tailor-made dataset of NRK's Subtitled TV (NST) subtitle files, and process NST with the topic models Latent Dirichlet Allocation (LDA) and Top2Vec. I investigate the models' ability to create topics that are useful to NRK employees that work with publishing. To improve the models' performances, I also conduct a systematic study of the effects of preprocessing steps on the data and models' results using the Python preprocessing toolkit textPrep with extensions for Norwegian. Finally, to assess whether the topics generated by the topic models are useful for people who work in publishing, I conducted a user study on how interpretable the final topics are considered to be by NRK employees, including journalists and editors.

My contributions include a novel approach to generating structured metadata for employees to use in publishing, an adaptation of the textPrep toolkit for use with Other languages than English, a novel systematic study of the effect of preprocessing steps on language model-based topic models, and a user study of topic interpretability tailored for employees who work in publishing.

Preface

This document is the Master's thesis project report that serves as the delivery of the course TDT4900 Computer Science, Master's Thesis at Norwegian University of Science and Technology (NTNU). The project was written during the Spring of 2022 and is based in part on research and literature review conducted Fall of 2021 in preparation for the Master's thesis, as part of the course TDT4501 Computer Science, Specialization Project. The thesis project report was written by Magnus Reier Rushfeldt, Master student at the Department of Computer Science, Faculty of Information Technology and Electrical Engineering. The project was supervised by Professor Ole Jakob Mengshoel at the Department of Computer Science, with the help of company representatives Egil Ljøstad and Maja Wettmark at NRK.

I want to thank Professor Mengshoel for his contributions in the form of meetings, advice, and feedback, often on very short notice, throughout the project. His patience and constant optimism was extra appreciated as the project was delayed towards the end of the semester. In addition, I would like to thank Egil and Maja at NRK for their many meetings to help me understand the needs at NRK. I would especially like to thank Maja for her tireless efforts and quick replies while assisting me with preparing and conducting a user survey at NRK.

Magnus Reier Rushfeldt
Trondheim, July 1, 2022

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The basis for this document and the project it describes will be presented in this introductory chapter of the thesis. Initially, the reason for why the research of this thesis was done will be outlined in Section 1.1. This background provides the foundation for formulating a research goal and associated research questions in Section 1.2, followed by a plan for how the research will be conducted to address those goals and questions in Section 1.3. Finally, Section 1.4 provides an outline of how the thesis is structured.

1.1 Motivation and Background

Two sets of issues, important to two different groups of stakeholders, motivated the research done in this thesis project. Firstly, the need for improved Natural Language Processing (NLP) resources and techniques for the Norwegian language is considered necessary for several Norwegian companies and research institutions to meet diverse challenges in an increasingly digital future. This demand led to the creation of the research project SCRIBE. Secondly, NRK requires improved structured metadata for solving a diverse set of problems in their organization, which led to the formation of the MEGAS project, a work package (WP) of SCRIBE. Both groups of issues will be further explained in Subsection 1.1.1 and Subsection 1.1.2, respectively.

1.1.1 SCRIBE: Improving NLP for Norwegian

In recent years, innovations in the fields of NLP, Artificial Intelligence (AI), and Machine Learning (ML) have greatly improved the accuracy and quality of results for language processing tasks, such as sentiment analysis, text classification, and Automatic Speech Recognition (ASR). This technology leap has enabled NLP technology, more than ever before, to streamline digitization and information retrieval efforts, simplify user interfaces and provide language aids to people with special needs. An example of the latter is how speech-to-text transcriptions generated from robust new ASR systems can improve access to audio and audiovisual content for hard-of-hearing people. While these innovations have often focused on improving the benchmark for NLP tasks in English, language-specific and language-agnostic research efforts have made this technology available and usable across many languages other than English.

However, the availability of equally powerful resources for the Norwegian language is lacking. Norwegian is a resource-constrained language, meaning that not enough resources are available in terms of different types of Norwegian text corpora. This sparsity is exemplified by the amount of Wikipedia articles that have been written in each language, as a corpus of Wikipedia articles

in a particular language is often an essential text resource for training ML-based NLP methods for that specific language. In English, around 6.4 million articles exist, while there are only about 570K in Norwegian Bokmål, the most used Norwegian written language. This problem is even further complicated due to Norwegian being a multi-faceted language in all the forms of natural language: both spoken and written. In fact, Written Norwegian exists as two Norwegian languages, Bokmål[1] and Nynorsk[2], that have equal status as official languages in Norway. Even though Bokmål is used as the primary written language by 85-90 of the Norwegian population, and Nynorsk is only used by 10-15, there is still a strong political incentive for increasing the availability of resources for both languages. Meanwhile, in the spoken dimension of Norwegian, there exists a myriad of Norwegian dialects that can differ significantly in vocabulary and pronunciation[3]. In short, efforts to improve language technology for Norwegian must not only take into consideration a lack of resources in general, but also the fact that these resources are spread thin over multiple variants of the language in both spoken and written form, and that all these variants and their users need improved language technology. The severity of this resource problem has led to the Norwegian government highlighting the need for improving NLP technology for Norwegian in their AI strategy document from 2020[4].

Several companies and research institutions have recognized the severity of this issue and have come together to focus research efforts on a joint project. These partners are:

- **Norsk rikskringkasting (NRK)**, or the **Norwegian Broadcasting Cooperation:** The public national broadcasting company in Norway. NRK produces and has access to many Norwegian multimedia sources (news articles, TV programs, and radio programs), and wishes to utilize better the data contained in these resources.
- **Telenor:** The largest Norwegian telecommunications company in Norway
- **Nasjonalbiblioteket**, or the **National Library of Norway (NLN):** Norway's only national library responsible for depositing and digitizing documents produced in Norway for future use and study. They also have access to vast corpora of Norwegian multimedia sources, especially audio and text, that the group can use for improving Norwegian language technology
- **Norwegian University of Science and Technology (NTNU):** One of the leading universities for STEM studies in Norway. NTNU has experience in NLP and AI research, and multiple ongoing efforts to improve the state-of-the-art for NLP.
- **Norwegian Open AI Lab (NAIL);** A research hub focusing on AI research, hosted at NTNU. NAIL provides resources and facilities for the joint project.

Together, these partners created the research project SCRIBE: Machine transcription of Norwegian conversational speech. This project has the goal of developing ASR-systems, or speech-to-text transcription systems, for Norwegian speech spoken by multiple users in multiple dialects[5]. This research goal intends to address the issues of spoken Norwegian resource challenges. The complete overview of the research areas in SCRIBE can be seen in Figure 1.1. As the figure shows, one project related to SCRIBE, Metadata extraction, also focuses on how users can utilize the text generated from transcriptions generated by the systems developed in SCRIBE for other purposes. This focus has led to the project MEGAS, which is detailed in Subsection 1.1.2.

[1]Bokmål: https://en.wikipedia.org/wiki/Bokm%C3%A5l
[2]Nynorsk: https://en.wikipedia.org/wiki/Nynorsk
[3]Norwegian dialects: https://en.wikipedia.org/wiki/Norwegian_dialects
[4]Strategy document: https://www.regjeringen.no/en/dokumenter/id2685594/
[5]SCRIBE project website: https://scribe-project.github.io/

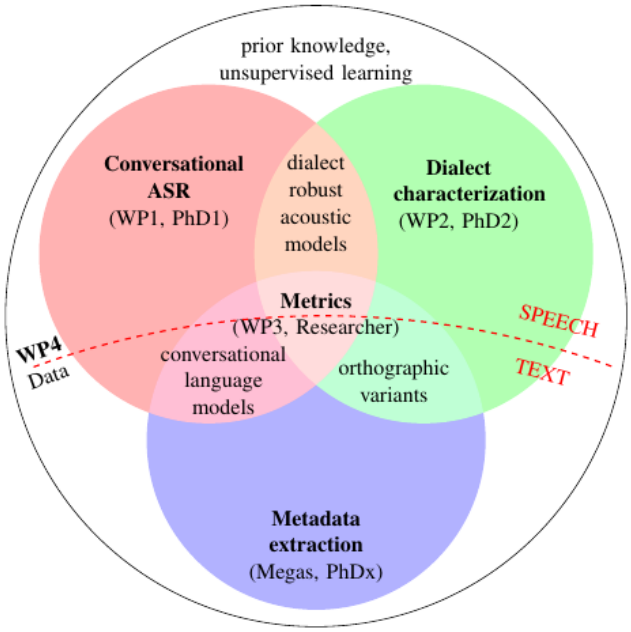


Figure 1.1: Venn diagram showing the different work packages (WPs) of the research plan in the SCRIBE project and where their scopes overlap. The four WPs are Conversational ASR, Dialect characterization, Metrics, and Metadata extraction. The last WP, Metadata extraction, is what is defined as MEGAS and is detailed in Subsection 1.1.2

1.1.2 MEGAS: Generating Structured Metadata for NRK

Because NRK has extra issues they wish to solve other than improved ASR technology, which is the main focus of all the partners in the SCRIBE project, they have also taken the initiative for the project MEGAS: MEtadata Generation of Auto-tranScribed text. As mentioned in Subsection 1.1.1, this WP in SCRIBE aims to explore how users, both employees and the audience of NRK, can use metadata extracted from Norwegian transcribed text and other Norwegian text resources to improve the value of these resources. NRK namely has several issues today that require improved metadata on all their multimedia content across all platforms:

- **Editorial insight:** With an extensive library of multimedia content, it is hard for journalists and editorial staff in NRK to keep track of all the content they have and what topics the content concerns. Content lack structured metadata that provides information about what a specific TV program or news article is about from a glance and how it relates to other media items and their topics. This absence of content-descriptive metadata makes it tedious for editorial staff to keep track of and publish content, so they are sure they cover the stories they need to cover and can find relevant content to publish together.
- **Reporting of content coverage:** NRK has a mandate from the Norwegian government to produce content that follows a set of guidelines, according to the so-called "NRK-plakaten" ("the NRK poster")[6]. These guidelines include providing a balanced and politically neutral viewpoint on cases in news coverage and providing content covering all types of topics and

[6]The NRK poster (only in Norwegian): https://info.nrk.no/vedtekter/?/l3v4qr5cy50

genres that are important and interesting to all parts of the population. To ensure that NRK follows these guidelines, NRK has to report all activities they do to Medietilsynet (the Norwegian Media Authority) in yearly reports[7]. This report has to show through data collection and statistics what types of topics NRK actually do cover. Today, this job is tedious and requires a lot of manual effort to show what content NRK has.

- **User access to content:** NRK has three leading online platforms that are widely used by the Norwegian population today: nrk.no (news platform), NRK TV (video streaming platform), and NRK Radio (radio/podcast streaming platform). Users are offered a comprehensive and extensive range of news articles, TV programs/videos, and radio programs/podcasts on these platforms. However, finding the content the user is looking after is not always easy. NRK has limited use of recommendation of content based on user preferences, and there are limited filtering and search capabilities on the platform for actively finding content of certain themes, topics, and genres.

To help in the research efforts on how to best extract metadata from transcribed text, NRK is providing me with a large dataset which they call **NRK's Subtitled TV (NST) dataset**. This dataset consists of 14,614 subtitle text files, mostly in Norwegian. Each file contains the full set of subtitles belonging to a single TV program in NRK's library of media content. This dataset substitutes actual transcribed text generated with ASR from speech, and allows for testing the results of algorithms found during the research. The NST dataset will be further explained in Section 4.2.

1.1.3 Research Focus: Topic Modeling

The issues highlighted above are the main drivers for NRK creating the MEGAS project and the associated thesis problem description that led to this thesis. Due to the need for all parties of the SCRIBE project to make the content of ASR transcriptions accessible in a meaningful way, and NRK's need for better structured metadata that mainly helps to highlight the themes and topics of their content, the choice was made to focus on how to extract topic information from TV subtitle text data. In order to perform this topic extraction, **Topic Modeling**[8] will be used. Topic modeling methods attempts to find the underlying topics in a text document, that is identifying different persons, places, actions and/or abstract themes that a document entails. I believe that extracting topics from text in this way will help contribute to both the needs of the SCRIBE project in general, and specifically to NRK's need for topical information on their content. A more succinct definition of the research focus will be defined through a research goal and associated set of research questions in Section 1.2.

1.2 Goals and Research Questions

Based on NRK's needs for informative topic metadata, along with all the project stakeholders' need for improved language technology for the Norwegian language, the following research goal has been formulated:

Research goal Investigate different topic modeling methods and their capabilities to perform topic modeling on a group of TV programs based on their Norwegian subtitle text, such

[7]About NRK's mission and reporting of activities: https://www.nrk.no/etikk/slik-defineres-nrks-oppdrag-1.11371666

[8]Topic modeling: https://en.wikipedia.org/wiki/Topic_model

that a non-expert user, specifically a journalist or editor at NRK, can easily interpret the topics outputted and how they relate to the content of the TV programs in question.

There are several aspects of this goal that needs to be examined and addressed. Firstly, a study of existing methods for topic modeling must be performed. These methods must also be compared and contrasted with respect to what input data and parameters they require, what output they produce, what potential algorithm training or learning is necessary, and the quality of their results. As topic modeling methods might not work optimally on unprocessed data, part of this study should also involve how much preprocessing must be performed on data to give usable results for each model, and how varying degrees of preprocessing affect the quality of the results. Then, what extensions are necessary to adapt existing methods so they can perform topic modeling well on Norwegian text must be mapped out, and a set of common parameters and output forms must be found so that the extended models can be compared in the context of the research goal. Finally, the most feasible sets of methods and preprocessing configurations must be experimented with and compared for finding the ideal setup with respect to strength of topic results, according to relevant evaluation metrics, and the usability and interpretability of the method from the perspective of a non-expert user. All these aspects can be formulated as the following list of research questions:

RQ1 What methods for topic modeling exist already and how do these methods work on Norwegian subtitle data?

RQ2 Which preprocessing steps can be used on input data for each model found in **RQ1**, and what effect do these steps have on the intrinsic data quality as well as the results of each topic model?

RQ3 Based on **RQ1** and **RQ2**, what combination of topic modeling method and preprocessing steps give the best results according to journalists and editorial staff in NRK?

In Section 1.3, I will further present how this thesis attempts to answer the research questions above and ultimately contribute towards the research goal.

1.3 Research Plan

The structure of this research will be divided into three parts, coinciding roughly with each research question:

1. First, I will explore and experiment with topic modeling methods. To answer **RQ1**, a brief literature study of existing topic modeling methods will be performed, and interesting candidates will be selected for further use. Then, these methods will be tested on the NST dataset and evaluated.

2. Then, once candidate topic modeling methods have been found, preprocessing methods for topic modeling and necessary extensions for the Norwegian language will be explored. Different combinations of preprocessing methods, along with Norwegian extensions, will be tested with all topic model candidates and evaluated. This step is intended to answer **RQ2**.

3. Finally, after the optimal combinations of topic models and preprocessing steps have been found according to automatic metrics, the results of the optimal setups will be evaluated by real users. I will construct a survey with the results which will be sent to NRK employees,

mainly journalists and editors, where they will be asked to interpret and evaluate the results according to relevant human evaluation metrics. This concludes the research by attempting to answer **RQ3**.

1.4 Thesis Structure

My thesis adheres to the the following structure:

- **Chapter 1 - Introduction:** The current chapter, which introduced the background for the research project, the research goal and research questions to be answered, the research plan for contributing towards said goal and questions, as well as this thesis overview.
- **Chapter 2 - Background Theory and Related Work:** Here, the background material necessary to understand the work done in this thesis is presented. This includes an introduction to different aspects of topic modeling, as well as preprocessing methods and evaluation metrics. The chapter is concluded with a section on related work.
- **Chapter 3 - Methodology and Architecture:** In this chapter, the methodology that will be used for the research, as well as the architecture that implements it, will be presented. Starting with a display of the pipeline that will be used for preprocessing of the NST dataset, the chapter will move on to discuss the setup for the different topic models used, and how the results from the topic models will be presented. Finally, a section is dedicated to the user survey that will be sent to NRK employees.
- **Chapter 4 - Experiments and Results:** After the architecture has been explained, the experimental setup for the different experiments to be performed will be introduced. First, the NST dataset from NRK will be introduced and explored. Then, three experiments and their results will be detailed: A preliminary experiment to assess viability of topic model candidates, a preprocessing experiment to find the optimal combination of preprocessing steps to perform before topic modeling, and a topic interpretation experiment in the shape of the NRK user survey.
- **Chapter 5 - Discussion:** The analysis of the results found in the preceding chapter will here be analyzed, experiment by experiment. This chapter will also address how each research question has been answered, and the limitations of the results found.
- **Chapter 6 - Conclusion and Future Work:** The final chapter of the thesis will summarize the work of the thesis and the contributions found. It will also highlight focus areas for future research.

Chapter 2

Background Theory and Related Work

In this chapter, background literature that is relevant to the project will be discussed. The first section, Section 2.1, will introduce the field of topic modeling, and different methods for topic modeling will be highlighted. The subject of preprocessing methods and how this can affect topic model quality is explored in Section 2.2, before Section 2.3 goes through different approaches and metrics for evaluating topic model quality. Finally, previous research that has explored topics and areas related to my thesis work will be highlighted in Section 2.4.

2.1 Topic Modeling

2.1.1 Definition of Topic Modeling

Topic modeling is the process of finding underlying topics in a text[1]. Different approaches and methods exist for performing this operation, but the basic concept is the same for all topic models: Given a corpus of m documents, $D = \{d_1, ..., d_i, ..., d_m\}$, and a vocabulary of all the n words in D, $V = \{v_1, ..., v_n\} \subset D$, a topic model M should return a set of k topics, $T = \{t_1, ..., t_j, ..., t_k\}$, so that each document has one or more topics with a given *representativeness* to the document:

$$R_{d,M}(d_i, t_j) \tag{2.1}$$

Each topic t_j is represented by a set of u topic words from V, $t_j = \{w_{j,1}, ..., w_{j,c}, ..., w_{j,u}\} \subset V$, so that each topic word has a given representativeness to the topic:

$$R_{t,M}(t_j, w_{jc}) \tag{2.2}$$

Here, *representativeness* refers to a measure of how representative a topic model thinks a word is for a topic, and how representative it considers a topic to be for a document. A high representativeness score for a given topic and document means that the topic could be considered to accurately convey one or more of the topics in the document, while a high representativeness score for a given word and topic means the word should be well within the domain defined by the topic. This representativeness measure is often represented as a probability, but can also be

[1] Topic modeling: https://en.wikipedia.org/wiki/Topic_model

represented in other ways, such as a similarity measure. In sum, a topic model M can be defined as:

$$M(D, V) = T \tag{2.3}$$

2.1.2 Overview of Topic Modeling Approaches

There are several different research areas that are based on different approaches to performing topic modeling. The classical approach, which is also the most used one, is what is known as generative probabilistic algorithms. These attempt to model documents as probability distributions of topics, and topics as probability distributions of words. This approach, as well as examples of methods from this field, will be discussed in Subsection 2.1.3. A more modern approach, which is very novel and barely explored, is that of distributed representation models. These models use joint word and document vector embeddings to represent words and documents, and then use cosine similarity measures and clustering to find topics, topic words and topic-document relations. This approach will be examined in Subsection 2.1.5.

In order to find relevant topic models for performing topic extraction in subtitle text, the topic modeling literature review of Vayansky and Kumar [2020] was consulted. In this review, some of the most cited topic modeling methods, as well as some new ones, are presented and discussed. Each model's strength and weaknesses are highlighted, and the authors suggest which set of problems and associated constraints each model are best suited for with the help of a decision tree. The models that are best suited for the task in this project will be discussed in Subsection 2.1.3.

Although this review provides a comprehensive review of some of the most used topic models, the scope of the review is limited to focus on generative probabilistic topic models alone. Therefore, other literature outside of Vayansky and Kumar [2020] was reviewed to find other approaches. One of the more promising approaches was found in the distributed representation models, which is represented by the Top2Vec model of Angelov [2020]. This model will be discussed in Subsection 2.1.5.

2.1.3 Generative Probabilistic Topic Models

In the generative probabilistic topic modeling algorithms, topics and topic words are modeled as latent variables[2] in the documents and topics, respectively. These latent variables, if known, would generate the probability distribution over a set of text that best represent another text. But since the latent variables are unknown, they have to be learned/estimated in some way, which is what the topic model attempts to do.

In this approach, there are usually two sets of distributions used. Based on the notation defined in Subsection 2.1.1, the first type is a distribution for each topic, t_j over all the words, V, representing the probability that a word is in t_j. This means that the representativeness measure for topics and words here is represented as a conditional probability of the word appearing given the topic. Using the notation of Equation 2.2, we thus have:

$$R_t(t_j, w_{j,c}) := P(w_{j,c}|t_j) \tag{2.4}$$

The other type of distribution is per document, d_i, over all the topics, T, representing the probability that a word is in d_i. Similarly as for topics, the representativeness measure for documents and topics is thus represented as the conditional probability of the topic appearing given the document, Using the notation of Equation 2.1, we thus have:

[2]Latent variables: https://en.wikipedia.org/wiki/Latent_variable

$$R_d(d_i, t_j) := P(t_j|d_i) \tag{2.5}$$

In addition to the inputs of documents D and vocabulary V from the documents, these algorithms also require the number of topics, k, to be specified as an input. Thus, modifying Equation 2.3, we get the following equation for LDA:

$$M(D, V, k) = T \tag{2.6}$$

This tells the algorithm how many topics the algorithm should attempt to find, which is necessary in order to model a probability distribution over topics for each document. Changes in the value of k given to the model can greatly affect the quality of topics, and so must be taken into consideration in order to optimize the results given by the model. Although there exists approaches to iteratively and automatically estimate the optimal k to give to the model, as explained in Vayansky and Kumar [2020] these methods are often computationally expensive, cumbersome and does not necessarily find the optimal k in all situations. Therefore, these methods will not be further explored in this thesis.

Generative probabilistic topic models often require some amount of text preprocessing to give the wanted results. As these models simply try to find the most likely words to appear in a document, topic modeling on unprocessed text may quickly lead to frequent, but semantically uninteresting, words such as "the", "and", and "of" being chosen as topic words in topics. In addition, words that are inherently meaningful, but prevalent across all the documents in a corpus, such as the words "machine" and "learning" in a corpus about ML methods, may be chosen as topic words even though they do not provide any useful information as the common topic of the corpus is obvious. Both these categories of words could be considered "stop words", and must often be removed to get the intended results. Removing stop words, as well as other preprocessing methods that are useful for topic modeling, are discussed further in Section 2.2.

The point of this approach is to find the words that most accurately represent each document through the combination of topic-word and document-topic distributions. One of the first and most successful methods in the generative probabilistic field used only this simple approach, a method called Latent Dirichlet Allocation (LDA), which will be discussed in Subsection 2.1.4. Based on the NST dataset properties discussed in Section 4.2, as well as the research goal and questions formulated in Section 1.2, there are several from Vayansky and Kumar [2020] candidates that could be considered relevant: LDA, the Correlated Topic Model, the Pachinko Allocation Model, the Dynamic Topic Model, and the Topics Over Time model. All the methods work well for moderately long and long documents (meaning documents that contain more than 50 words), which account for almost all the NST subtitle files. However, due to time constraints and to reduce the complexity and number of experiments that would be needed to perform across different variants, I decided to focus on only one algorithm from this group: LDA. LDA has achieved good results across many different tasks and problems, and has been adapted and extended to many different use cases. Besides, the article that introduced LDA, [Blei et al., 2003], is by far the most cited of the algorithms, with over 40,000 citations. Thus, this algorithm would provide the best baseline for the later experiments as a representative for this category of topic models. And by focusing on only one algorithm from this group, more focus could be devoted to algorithms from other research areas using other approaches.

2.1.4 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) was the first successful generative probabilistic algorithm to be used. Introduced in Blei et al. [2003], this algorithm started as a type of Bayesian network

that was developed specifically for topic modeling. It follows the core approach of generative probabilistic algorithms as described in Subsection 2.1.3. For probability distribution modeling, LDA models documents using latent Dirichlet stochastic variables which lead to a probability distribution over all k topics. Then, for each topic, latent multinomial stochastic variables model the probability distribution over all words. The model then iteratively tries to find the optimal Dirichlet variables for the documents, and the multinomial variables for the topics, so that given a document, the distributions of topics for the documents, and the distributions of words for those topics, has the highest probability of recreating the document (that is, the words present in the document).

2.1.5 Distributed Representation Topic Models

A novel approach to topic modeling is the use of joint word-document vector embeddings to represent words and documents, and the use of clustering and similarity measures to find topics. In this approach, the ability of vector embedding representations of words and documents to model semantic relations between each other is leveraged. The underlying assumption of this approach is that, once words and documents are embedded as vectors, words that appear close together in the vector space can be considered to have similar semantic meaning, while words that appear close to a certain document should be semantically related to something the document is about. Based on these assumptions, clusters of words and documents appearing together indicates that this area in the vector space encodes an underlying topic of the documents, and that the words of the topic can be represented using the words of this cluster.

More precisely, using the notation defined in Subsection 2.1.1, the documents D and words of V are all embedded as vectors in a vector space of some dimensionality. Only D is needed, as Top2Vec automatically learns V from the documents while processing them. Then, k topics are defined by the number of distinct clusters of word and document vectors that can be found. Each topic $t_j = \{w_{j,1}, ..., w_{j,c}, ..., w_{j,u}\}$ belongs to and is defined by one of these clusters, where t_j is itself also a vector in the cluster. A topic word is a subset of the topic, $w_{j,c} \subset t_j$, if the word is in the cluster, which means the cosine similarity of the word to the topic is high enough. The cosine similarity also defines the representativeness of the given word to the topic. Using the notation of Equation 2.2, we have:

$$R_t(t_j, w_{j,c}) := \cos(t_j, w_{j,c}) \tag{2.7}$$

A document d_i belongs to a topic if its vector belongs to the same cluster as the topic vector, meaning that the cosine similarity of the topic to the document is high enough, Thus, the representativeness of a given topic to the document is defined by the cosine similarity between them. Using the notation of Equation 2.1, we have:

$$R_d(d_i, t_j) := \cos(d_i, t_j) \tag{2.8}$$

As a document can only belong to at most one cluster, this means that each document can only be represented by one topic. In addition, as some documents might be too far away from any cluster to be assigned to a cluster, it could end up not being represented by any topics. Finally, the Top2Vec model, only needing the documents as input, has the following equation modified from Equation 2.3:

$$M(D) = T \tag{2.9}$$

These topic models have the advantage of being intended to use directly on mostly unprocessed text, so good results might be achieved even without any text preprocessing. At the same

time, a disadvantage is that each document can only be represented by one topic, even though the document might in reality have been more accurately represented by multiple topics. The only known method to use the distributed representation approach at the time of writing of this thesis, is Top2Vec, which is the method of this approach that will be discussed here. See Subsection 2.1.6 for an exploration of that method.

2.1.6 Top2Vec

The Top2Vec method was introduced in Angelov [2020]. Top2Vec creates a joint word-document embedding as explained above, and performs dimensionality reduction and clustering, before finally finding topic vectors for each cluster to render the topics. For the embedding, Top2Vec offers the choice of using either Doc2Vec, Sentence-BERT or Universal Sentence Encoder. The vectors that are created using the embedding will tend to have a very high dimensionality, so document and word vectors are often far apart (with low cosine similarity). Thus, before any clustering can be found, dimensionality reduction must be performed. This is done using a method called Uniform Manifold Approximation and Projection (UMAP), which projects all the vectors into a space with much lower dimensionality in the way that maximizes the density of clusters of word and document vectors. Now, clustering is performed using the method Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), which groups together all documents and words that are found in dense clusters, and ignores outlier words and documents. Then, a topic vector is defined for each cluster by calculating the centroid vector of each cluster. This is the vector that best represents the geometric, and thus ideally the semantic, center of the cluster. Finally, a given number of the word vectors closest to the topic vector are selected, and these vectors represent the topic words of the topic vector, and thus represent the actual topic.

In Angelov [2020], Top2Vec is compared to LDA and gives better results. As an embedding-based model, it also has the advantage of being able to perform topic modeling on text data that is unprocessed, does not require the number of topics k to be manually selected, can group topics hierarchically and can find topics in shorter texts (such as social media posts). Although Top2Vec is a novel method of a largely unexplored topic modeling approach, it has been used successfully in other independent research. In Hodgson et al. [2021], Top2Vec was used to find important topics in the research field of intelligent tutoring systems during a 20-year period. Meanwhile, Ma et al. [2021] examined topics in tweets about COVID vaccines using both Top2Vec and LDA models, and found that the Top2Vec topics were more relevant. Based on these findings, Top2Vec is a promising candidate to perform topic modeling along with LDA.

2.2 Text Preprocessing Methods

In order to get good quality results from NLP algorithms on texts, one often needs to prepare the text using some level of text preprocessing. Text preprocessing is related to the principle of GIGO: "Garbage In, Garbage Out". This principle states that good results from an algorithm can only be achieved if the data input is of good enough quality. Text preprocessing involves scanning through the raw text data that is to be used for a certain algorithm, and performing one or more operations on the text to either remove or change parts of the text. The operations can be done on the character level, word level or sentence level. This is done to shape and adapt the text so it better fits the assumptions and input requirements of the given algorithm that will process the text later, which will ideally improve the results of the algorithm.

When it comes to topic modeling, many methods often require a certain level of preprocessing on the input text to return good quality topics. This is mentioned briefly in subsection 1.5.5

of Boyd-Graber et al. [2017], and properly explored in Churchill and Singh [2021b], the latter article of which is further discussed in Subsection 2.4.1. Such preprocessing can involve steps such as removing certain characters, tokenizing sentences, normalizing word case and removing stop words. This is especially true for generative probabilistic topic models, which is briefly touched upon in Subsection 2.1.3. All these steps, as well as other steps that can be relevant for topic model preprocessing, will be briefly outlined in the following subsections.

2.2.1 Tokenization

The minimum requirement for many topic models, including generative probabilistic ones, is that documents are fed into the topic models as list of words. In order to get a list of words from a continuous text string (which is usually how raw text documents are represented), tokenization must be performed. This involves splitting the sentence into tokens, which means separating words, numbers and special characters from each other, so they can be individually listed. The most straightforward way to do this involves simply splitting a sentence on every occurrence of whitespace and special characters, although there are more sophisticated approaches for this if it is important to preserve certain chains of tokens, for example tokens that are connected to each other by hyphens (such as "10-year-old").

2.2.2 Character Removal

Oftentimes, certain special characters in text are noise or clutter for a topic model. This usually means punctuation, and other characters that are not alphabetical or numerical. Sometimes, one might even want to remove numbers, so that only alphabetical characters are preserved. In social media texts, more complicated groups of characters may be unwanted, such as user tags and hyperlinks. Removal of punctuation is a very common basic preprocessing step for most topic models.

2.2.3 Normalization

A word that appears multiple times with different capitalization and word cases across a document (such as "normalization" vs "Normalization") is often unwanted, as this may lead to the same word being counted as different words when frequency of words is to be considered. Therefore, normalization forces all words to the same word case (usually lower case) to ensure different capitalizations of the same word are ignored. In addition, normalization can also mean numbers written using digits are converted to alphabetical spellings, so that all occurrences of the same number are consolidated into the same canonical representation.

2.2.4 Lemmatization

A more sophisticated form of normalization, lemmatization is the process of rewriting all inflections of a word into the word's lemma, that is the base form of the word. Thus, words such as "be", "is", and "are", which are inflections of the lemma "be", are all converted into "be". This process can significantly increase the frequency of a word that mostly occurs in different inflected words across documents. However, lemmatization can be hard to perform perfectly, and it can often be computationally expensive.

2.2.5 Stop Word Removal

Stop words are words that have adds little semantic meaning to a sentence, which involves words such as "the", "and", and "of". As they carry almost no meaning, they are not very useful as topic words for representing topics. However, as they are often quite frequent in normal text, they can quickly dominate the list of topic words in a topic. Thus, they must often be removed to get meaningful topics. As these words belong mainly to closed word classes, meaning word classes that contain a low and finite number of words, most stop words can be filtered out using predefined lists of stop words.

2.2.6 Term Frequency Removal

Sometimes stop words can also be words that are meaningful in general, but not meaningful in a given context because they are too abundant across all documents. Instead of manually the most abundant terms to a custom stop word list to remove them, one might remove words such as these by simply counting the frequency of all terms that appear in the corpus, and remove the terms that have a frequency above a certain threshold.

2.2.7 TF-IDF Removal

Not all words that could be considered stop words are frequent enough to be filtered out simply using term frequency. For example in a corpus of text documents about machine learning, one would expect to find the terms "machine" and "learning" in most documents. Although both terms have semantic value in a given sentence, they do not help in identifying topics that are specific only to certain subgroups of documents in the corpus. In this context, "machine" and "learning" could thus be considered stop words. However, they might only appear a few times in each document, and might not appear at the top of list of most frequent terms. This is where Term Frequency, Inverse Document Frequency (TF-IDF) comes in. By counting the number of documents a term occurs in as well (regardless of how many times in each document), and dividing the term frequency by this number, a score is returned that indicates how unique a term is to a subset of documents in the corpus. A term with a high TF-IDF score likely occurs often in only a few documents, meaning these terms are potentially valuable for identifying unique/diverse topics. Conversely, a term with a low TF-IDF score is probably less interesting as it either appears very little, or appears across most documents. Thus, by removing terms with a TF-IDF score below a certain threshold, terms such as "machine" and "learning" in a corpus with documents about machine learning are likely to be removed.

2.2.8 Part-of-Speech Removal

Part-of-Speech (PoS) tagging is the process of identifying what word class each word in a sentence belong to. This also involves resolving ambiguity when two different words from different word classes and with different meanings have the same spelling, such as the noun "model" vs the verb "(to) model". This tagging is useful if one only wishes to retain certain classes of words, and remove others. For example, one might only want to keep nouns to use as topic words in a topic model. In this case, after a document has been PoS-tagged, all words that are not tagged as nouns can simply be dropped, and only the nouns of each document will remain. PoS-tagging often requires knowledge of the language to be performed well.

2.3 Topic Model Evaluation

Good evaluation metrics are important to ensure the quality of topic model results and to reasonably compare the results with that of other methods. As topic modeling is often performed as a form of unsupervised learning, meaning there is no ground truth to compare topic model results to, it is often necessary to have evaluation metrics that can be used directly without ground truths. A good overview of widely used evaluation methods for topic models is found in Churchill and Singh [2021a]. Topic modeling evaluation metrics generally revolve around three areas of assessment: coverage (Subsection 2.3.1), coherence and diversity (Subsection 2.3.2) and qualitative evaluation (Subsection 2.3.3).

2.3.1 Topic Coverage

This type of evaluation metric tries to measure either how well a predefined set of topics or the documents in the corpus are covered by the topics produced by the topic model. There are various metrics used here. Topic recall measures how many of the original ground-truth topics were discovered. Accuracy meanwhile measures how many of the documents are accurately covered by the topics found, based on ground-truth topics. Purity is an accuracy measure not requiring ground truth, while KL-divergence allows comparing coverage of a given model to that of a baseline model. Finally, perplexity is a measure of how well a trained topic model can predict the topics of an unseen document, although this measure is shown not to correlate well with human evaluations. Coverage as a measure will not be focused on in this study, as there are no ground truths of topics present, and coverage metrics that do not use ground truths are limited in how well they can gauge the coverage of topics.

2.3.2 Topic Coherence and Diversity

While coverage assess the topic result set as a whole, coherence evaluates the quality of individual topics, by measuring how consistent and related the topic words of a topic are to each other. A classic approach here is topic precision, measuring what fraction of words in the found topics match that of a similar ground truth topic. However, the most widely used approach is that of Pointwise Mutual Information (PMI). This metric does not require ground truths, and measures coherence by calculating relative co-frequencies of words. There exists several variants of this measure, including a normalized variant called NPMI.

Related to the measure of topic coherence, is that of topic diversity. This metric measures how much words of topic overlap, represented as the fraction of topic words that are unique in the set of top n words from each topic. This measure, similarly to PMI, does also not require ground truths. If many of the top topic words appear across multiple topics, it means the topic set as a whole is less unique. Thus, a coherent topic set result require both that individual topics are coherent, and that all the topics are diverse.

In addition, Churchill and Singh mentions the measure topic intrusion introduced by Chang et al., which will be further explored in Subsection 2.4.2. This metric tries to assess how coherent a topic is considered by a human, by measuring the fraction of human subjects who can correctly select the word that does not belong in a topic, when a number of words belonging to a topic in addition to a word from another topic is placed together.

2.3.3 Qualitative Topic Evaluation

Although automatic metrics can provide some information about the quality of topic model results, they struggle to represent how well humans can understand and interpret the results.

Thus, qualitative analysis of topics using human subjects can add valuable insight into how useful topics and topic-documents pairings found by a topic model will be in the real world. Qualitative analysis can be used to assess the same aspects as covered in the subsections above, such as coverage, coherence and diversity. The qualitative measures of word intrusion and topic intrusion introduced by Chang et al., and what information they provide, will both be discussed in Subsection 2.4.2.

2.4 Related Work

Performing topic modeling on subtitles in Norwegian, with a focus on the effects of preprocessing on the results and user interpretation of the results, is a relatively novel study with little to no work found doing a similar approach. Still, there are some related work that are similar to part of the work I will be doing in this thesis. Below, an article on preprocessing for topic models (Subsection 2.4.1), as well as an article on how humans interpret topic model results (Subsection 2.4.2), will be examined and compared to my approach.

2.4.1 Preprocessing for Topic Modeling

According to Churchill and Singh [2021b], the effects of preprocessing on topic model results are severely under-explored in the literature relating to topic modeling. The authors found almost no literature that systematically examined how these steps might affect the topics that are produced from a given topic model. To contribute in this area, Churchill and Singh formalized a preprocessing rule methodology, as well as classes of rules, and normalized the inputs and outputs of each of these rules so it would be possible to stack rules in a modular fashion in a preprocessing pipeline.

In order to demonstrate this framework and systematically explore the effects of preprocessing, the authors created a Python toolkit called textPrep that implemented their preprocessing methodology, and conducted experiments using the toolkit. They tested a number of different rule stacks, or pipelines, on three different topic models: LDA, the Dirichlet Multinomial Model (DMM), and the GPU-based DMM (GPUDMM). Using three different social media datasets from Twitter, Reddit and Hacker News, they examined how each pipeline affected statistics of each dataset, and then examined how the results of each topic model result changed after receiving each preprocessed variant of the dataset. The topic model results were evaluated using a topic coherence measure based on NPMI, as well as measuring topic diversity (see Subsection 2.3.2). They found that different pipelines had different effects on each model, and that different pipeline-model configurations would be optimal on each dataset. However, the most important finding was that significant improvements in topic coherence and diversity could be found for most models, as long as the right combination of preprocessing rules were used in the pipeline.

This article is interesting as it shows the potential that preprocessing can have on topic model results, and presents a way to examine this in a structured fashion, which lays the groundwork for the work I do in this thesis. Besides, as the textPrep toolkit is available as an open-source Python library, I am able to efficiently explore preprocessing configurations of my own.

There are some areas of this research which could be expanded and improved upon. First of all, the models considered in this study does not take into account some newer model concepts, for example models based on the distributed representation algorithms. Although some of these models should not need preprocessing of the input text before topic modeling, what effects preprocessing of text can have on these models should be properly examined. Further on, Churchill and Singh [2021b] only focuses on social media datasets. Social media posts are usually quite

short documents, and finding topics in these using models such as LDA might not be the same as finding topics in longer documents. Finally, the textPrep toolkit is only made for use on English texts. The toolkit might need to be adapted in order for certain preprocessing methods to work on texts in other languages, and this should be studied further. In Section 4.4, suggestions for extensions to this research are suggested.

2.4.2 Human Interpretation of Topic Model Results

A well-known problem when it comes to evaluating the results of topic models, as discussed in Section 2.3, is that results that are considered good according to automatic metrics might not be good or useful to a human. Indeed, in a study on how humans interpret topic model results (Chang et al. [2009]), it was found that human interpretability of topics was actually anti-correlated with predictive likelihood, an automatic metric that is often used to evaluate topic model results.

The authors present two user tests to gauge human interpretability: word intrusion, assessing the topic coherence, and topic intrusion, a test measuring the link between documents and topics. In the first test, users are presented with sets of six words. Each set of words contain five words from the same topic and one intruder word that does not belong to the topic. The user is then asked to pick the word that they believe is the intruder. The model precision, that is the fraction of users selecting the correct intruder word, gives an indication of the human-interpreted coherence of the topic. The latter test, topic intrusion, presents the user with a similar challenge: A document is presented to the user along with four topics and their topic words, three of which have a high probability of modeling the document, and the last one being an intruder topic which has a low probability of modeling the document. Similarly to the word intrusion task, users are asked to pick the topic which they believe is the intruder. The topic log odds, the ratio of correctly picked topics, then models the human-interpreted accuracy of the topics belonging to a given document.

Experiments were then performed on a news article dataset and Wikipedia dataset, using the models of pLSI, LDA and CTM. The models were trained on part of the datasets, and the remaining parts were used to calculate the held-out, or predictive, likelihood of each model. Then, users on a crowd-sourced platform were asked to answer surveys with word and topic intrusion tests, containing the results from these models. It was found that model precision (the word intrusion metric) showed negative correlation with predictive likelihood for all models across both dataset, whereas topic log odds (the topic intrusion metric) showed no or negative correlation with predictive likelihood across both datasets, depending on model.

This article shows how some automatic metrics should be used with caution in establishing the quality of results, and that it is important to include real-world measurements such as human interpretability tests to really gauge how good topic model results are. This study therefore motivates the use of similar user surveys in the research. In Section 4.5, the modification of parts used in this survey are discussed.

Chapter 3

Methodology and Architecture

This chapter deals with the methodology that will be used for experiments, and the architecture that implements this. A high-level flowchart diagram displaying the full process is shown in Figure 3.1, and the sections in this chapter roughly follow each step of the flowchart from left to right. Each section will first present the methods used, and then the implementation of said methodology. The preprocessing pipelines, topic model processing and creation of visual topic representations will all be implemented in Python[1], and the full implementation can be found in a GitHub repository[2]. In Section 3.1, the pipeline setup to preprocess the subtitle files will be outlined, before Section 3.2 focuses more on the topic models to be used and their parameters. Then, Section 3.3 explains how the topic outputs are to be handled and presented, before Section 3.4 concludes with presenting the survey that will be conducted to test interpretability of the topics.

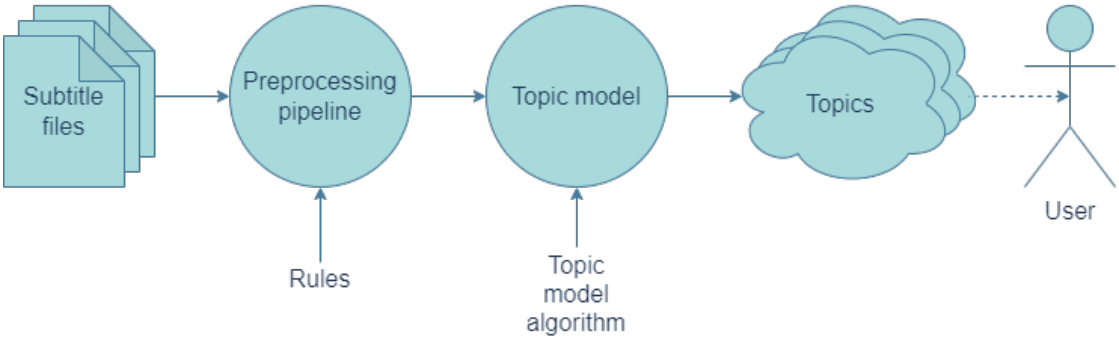


Figure 3.1: High-level flowchart diagram showing how subtitle files first go through a preprocessing pipeline, then a topic model, before they finally produce topics. These can then be evaluated by a user. It is also possible to extend the system in the future with other preprocessing rules for the preprocessing pipeline and other topic modeling algorithms to use as topic model

[1]Python website: https://www.python.org/
[2]Metadata extraction pipeline: https://github.com/magnurr/Master-Thesis-Project

3.1 NST Data Pipeline

As will be discussed in Section 4.2, the parts of the NST dataset that will be focused on in this study is purely the text part of the subtitle files, without the metadata. After extracting only the text part of the subtitle, the subtitle will then be sent through a preprocessing pipeline in the same style as the one used in Churchill and Singh [2021b]. Here, a stack of preprocessing rules will be applied to each subtitle text, before the preprocessed files are sent to topic models. Depending on the experiment, several variations of this pipeline, with different rules in each, will be used in parallel to assess the effects of preprocessing. This will yield multiple variations of the NST dataset to be further processed by the different topic models. In these experiments, token statistics of the different preprocessed datasets will also be recorded and assessed.

Using Python and the data science library Pandas[3], the CSV subtitle files will be modified using Pandas' `DataFrame` in Python. Next, the subtitle texts will be sent to the preprocessing pipelines. These pipelines will be implemented using the Python textPrep library[4] of Churchill and Singh, in order to take advantage of the methodology they introduce. Different pipeline stacks will be constructed using existing rules, Norwegian extensions to rules as discussed in Section 4.2 as well as some new rules added to textPrep. The extensions that are performed as part of this thesis can be found in on GitHub[5]. The preprocessed subtitle texts that each pipeline produces will each be written to a separate CSV file for later topic model processing. In addition, preprocessed dataset statistics will be generated and recorded using textPrep's built in statistics function. Below is a list of how each rule, as introduced in Section 2.2, is created as a new rule or adapted from an existing rule in textPrep. My adapted version of textPrep can be found here:

3.1.1 Character Removal: Punctuation, Number and Metadata

Based on the concept introduced in Subsection 2.2.2, all punctuation, numbers and subtitle metadata artifacts are removed. Punctuation removal is already a rule in textPrep that can be used as is, while a new rule had to be made for removing numbers. For removing subtitle metadata, a new rule also has to be created. Based on the description of subtitle metadata in Subsection 4.2.1, alignment tags (`\{\anX\}`), line-break characters (`\n` and `\r\n`), and the "Recording of live-captioning" ("Opptak av simultanteksting") phrases will be identified and filtered out.

3.1.2 Stop Word Removal: Norwegian List

Proper stop word removal (Subsection 2.2.5) for the NST dataset requires a proper stop word list for Norwegian. For this, Birkenes from NLN recommended to use a basic Norwegian stop word list from NLN covering the most important Norwegian stop words[6]. This can be expanded upon if other words are discovered that should be filtered out as stop words. The effects of stop word removal in this project is studied by adapting textPrep's existing stop word removal rule to use the Norwegian stop word list above.

[3]Pandas website: https://pandas.pydata.org/
[4]The textPrep GitHub repository: https://github.com/GU-DataLab/topic-modeling-textPrep
[5]textPrep extension for this thesis: https://github.com/magnurr/topic-modeling-textPrep-Norwegian-subtitles
[6]NLN's Norwegian stop word list: https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-70/

3.1.3 Lemmatization and PoS Removal: Norwegian Language Model

As lemmatization (Subsection 2.2.4) and Part-of-Speech-tagging (Subsection 2.2.8) both require knowledge of the language being processed, a suitable lemmatizer and PoS-tagger for Norwegian must be found. Once again, based on input from Birkenes [2022], the OBT-tagger of University of Oslo (UiO) was recommended as the best lemmatizer for Norwegian. However, as this tagger requires a Docker image to run, and I will use Python for text preprocessing, this tagger would require a lot of work to integrate with Python. Luckily, a Python-friendly alternative lemmatizer for Norwegian exist through the Python NLP library SpaCy and their Norwegian language model (including lemmatizer), `nb_core_news_lg`[7]. This model is not as good as the OBT-tagger, but has a decent lemmatizer. Another advantage of using the language model from SpaCy, is that this model also features a tagger for PoS-tagging in Norwegian. The existing rules in textPrep for lemmatization and PoS-tagging will both be adapted to use `nb_core_news_lg` as language model, and then lemmatization and removal of words with certain PoS-tags is tested.

3.1.4 Normalization, Term Frequency and TF-IDF: No Changes

Normalization (Subsection 2.2.3), as well as term frequency and TF-IDF removal (Subsection 2.2.6 and Subsection 2.2.7), will be performed using textPrep's existing rules. As numbers will be removed during character removal in this project, normalization of numbers will not be performed. However, normalization of word case will be performed. Term frequency and TF-IDF removal is done using appropriate thresholds.

3.2 Topic Model Configurations and Parameters

In this next part of the pipeline, the preprocessed subtitle files will be sent to the different topic models for generating topics. As discussed in Section 2.1, the topic models chosen for this study are LDA and Top2Vec. Both models will receive the preprocessed subtitle files as input. LDA will also need the parameter k to be given. After the topic models have generated the topics, both will produce a number of topics represented by words and their relative representativeness to their topic, $R_{t,M}(t_j, w_{j,c})$. Both models will also be able to return which topics belong to a specific document once given a document as input, as well as the topic-document representativeness for each, $R_{d,M}(d_i, t_j)$.

After topics have been generated, they must also be evaluated with automatic metrics. Based on the investigation done in Section 2.3, two of the best suited metrics for automatic evaluation without ground truths are topic coherence based on NPMI scores, as well as topic diversity. These are also the same metrics used in Churchill and Singh [2021b], which the preprocessing experiment will be based on. So each model generated from each pair of topic modeling method and preprocessed dataset will be evaluated according to these metrics.

The LDA model `LdaMultiCore` from the Python data science library Gensim[8] will be used to implement the LDA model.

Meanwhile, the Top2Vec implementation, `Top2Vec`, comes from the Top2Vec GitHub repository[9] of Angelov.

Finally, for topic evaluation, the textPrep package already provide the topic coherence and topic diversity metrics out of the box. Thus, textPrep will be used for calculating these.

[7]SpaCy Norwegian Bokmål language model: https://spacy.io/models/nb
[8]Gensim LDA model: https://radimrehurek.com/gensim/models/ldamulticore.html
[9]Top2Vec GitHub repo: https://github.com/ddangelov/Top2Vec

3.3 Topic Presentation

After getting the topics out from each topic model, the question is how to best present the topics so that they are understandable and easy to interpret by non-expert users. There are many possible ways to display topics from topic models, each with their advantages and disadvantages, and some of these are presented in Boyd-Graber et al. [2017]. One of the simplest forms are just using the list of words as outputted by the model, sorted according to the words with the highest representativeness to the topic. However, this representation is visually bland and it may take some time for the viewer to grasp what is the essence of the topic. Another alternative is using word clouds. Here, each word is displayed in a random area inside a canvas, so that the size of the word is proportional to its topic representativeness. The advantage of this representation is that there is an intuitive connection between the words that are first seen and which words are the most important, so a viewer can quickly grasp the essence of the topic. On the flip side, they can also confuse a viewer with being messy and chaotic to read, especially if words are displayed at different angles, and viewers might incorrectly assume words closer to each other are more semantically related. Finally, an improvement of the word cloud can use word associations to place words close to each other if they are related, for example by the use of co-occurrence. However this requires extra data to be generated and may end up being equally messy as a word cloud.

Based on the need to have a visually interesting display format, that also makes the actual topic represented quick to grasp, while only using the topic data given from the topic models, word cloud was chosen as presentation format. To create word cloud representations of the topic model topics, the Python WordCloud library[10] was used.

3.4 User Survey

In order to best evaluate how humans perceive the topic model results, which is essential to gauge how useful the topics would be to journalists in NRK, a user survey will be conducted containing questions related to topics of selected topic models. This survey will be intended for employees at NRK, especially those who work with publishing. The questions in the survey will be partially based on questions used in the survey of Chang et al. [2009]. See Section 4.5 for further detail on how our surveys differ and why.

The first part of the survey will deal with individual topics and how they are interpreted. Each topic will have a separate page, where the word cloud representation of the topic will be presented, and the following questions will be asked about the topic:

1. What name would you give to the topic? (a topic labeling task)
2. Which of the topic words belong to the topic? (a coherence task, inspired by word intrusion)
3. Does the topic describe any of the two following TV-programs well? (a document-topic association interpretation task)
4. On a scale from 1 to 5, how useful is this topic for describing the topic of a TV-program? (rating of the perceived coherence and topic interpretability)

Ideally, the combination of these questions should provide means of evaluating both how coherent the topic words are and how well topics are accurately assigned to programs. To study the diversity of topics and accuracy of assignments to programs even further, the second part

[10] WordCloud documentation: https://amueller.github.io/word_cloud/

of the study will provide two additional sets of questions: First, a proper topic intrusion task using the LDA topics, and then a modified topic intrusion task where only one topic is correctly assigned using the Top2Vec topics.

In addition, on each page and a separate final page there will be open questions about what participants think of the specific topics, and how the topics should be modeled in general. These questions are not explicitly part of the human evaluation part of the study, but are added to provide extra meaningful information to be used for discussion and future work.

To implement this study, a Google Forms survey[11] will be created. The topics will be selected from the LDA and Top2Vec models that get the best results in the preprocessing experiment.

[11] Google Forms: https://www.google.com/forms/about/

Chapter 4

Experiments and Results

This chapter deals with the three experiments that were planned to answer the research questions, as well as the results of these. The experimental plan will again be briefly outlined in Section 4.1, before the dataset to be used and each of the experiments are detailed in Section 4.2, Section 4.3, Section 4.4, and Section 4.5, respectively. Finally, the results of the experiments will be presented in Section 4.6.

4.1 Experimental Plan

To answer the research questions, three sets of experiments will be performed:

1. Preliminary model experiments to see if the topic models work on the Norwegian text in the subtitle files of the NST dataset, as well as examining topic model configuration parameters. This experiment is detailed in Section 4.3.

2. Preprocessing experiments to assess the effect of preprocessing rules on NST dataset and topic model quality. These experiments will be discussed in Section 4.4.

3. A topic interpretation experiment conducted as a user survey, to assess the human interpretability of the topic models, as well as identifying focus areas for future improvements. This survey is detailed in Section 4.5.

The first two experiments will be performed using the NST dataset, to be explored in Section 4.2. The final experiment will use the results of the two previous experiments, and gather data from respondents to the survey.

4.2 NRK's Subtitled TV (NST) Dataset

To assess whether theoretical approaches to metadata extraction on transcription text can work in reality, it is necessary to test approaches on real data. Although actual ASR-generated speech-to-text transcriptions would provide the most real examples of data to test on, there are some issues in ASR-generated transcriptions that make this data complicated to test on. Artifacts such as misinterpreted words and disfluencies (disruptions in the sentence flow in natural speech) might cause too much noise for the metadata extraction algorithms to properly work, and filtering out this noise might prove to be tedious. Besides, this part of transcription analysis is outside the

scope of the MEGAS project, and rather belongs to the scope of other sub-projects in SCRIBE. Therefore, a natural substitute to real speech-to-text transcriptions, that provide data close to real transcriptions which is also clean, is TV program subtitles. This is what NRK provides in the NRK's Subtitled TV (NST) dataset.

4.2.1 About the NST Dataset

The subtitles in the NST dataset are transcriptions of speech, but compared to ASR-generated ones, subtitles are most often manually created by humans. A human writing a subtitle can filter out artifacts that ASR systems might not be able to, and focus on reproducing the speech as cleaner and better flowing sentences (without disfluencies). Clean sentences that are grammatically and syntactically correct, as well as semantically meaningful, can therefore be analyzed using NLP algorithms designed for written text. Therefore, the NST dataset should provide a useful compromise between authenticity of the data and simplicity of processing, which allows this thesis to focus more on the metadata extraction part of the process.

The NST dataset contains 14,614 subtitle files, each being the full set of subtitles to an NRK TV program. There are an additional 2,7K subtitle files in Danish, but these will be ignored for this study. The subtitle files are represented in the proprietary file format PAC (file extension `.pac`), which cannot be read and edited by normal text file editors. However, the open-source program Subtitle Edit[1] can be used to open, read and edit the files. The program can also convert the PAC files to widely used open-source file formats. As it is unpractical to work with the data in the proprietary PAC format, they are converted to Comma-Separated Values (CSV) files to be used in this study.

As soon as the subtitle files are converted to CSV files, they can more easily be interpreted. Each line in the subtitle file contains four fields, represented as columns in the CSV file: "Number", "Start time in milliseconds", "End time in milliseconds", and "Text". The "Number" field is a sequence number, indicating which position the subtitle has in the sequence of subtitles. The "Start time in milliseconds" and "End time in milliseconds" fields indicate when each subtitle should appear on the screen and when it should disappear, respectively. These time values are given in number of milliseconds since the program started. Finally, the "Text" field contains the actual subtitle (line of dialogue) to be displayed on screen. Each subtitle is a complete sentence or sub-sentence, that either fully or partially represents a phrase uttered by someone in the program.

The first line of each subtitle file is special. This line is reserved for technical metadata about the subtitles that the file contains, and the program they belong to. This subtitle will always have the sequence number 0 and start time 0, with the end time being a few hundred milliseconds after program start. Although this is in itself a valid subtitle, it will never be visible as a subtitle to the viewer, and is only meant for internal use. Although most of the fields contain information that is not useful for a text analysis, there are two fields that are potentially important: "Variant" and a release date field. The "Variant" field contains a three-letter code indicating which Norwegian language is used in the subtitle: `BOK` for Bokmål and `NYN` for Nynorsk. This is an important field if the NLP algorithm being used needs information about which language is used in a given subtitle file. In addition, an unlabeled release date field (sometimes labeled "Levering") contains information about the release date of the program (sometimes multiple dates if the program has been broadcast multiple times). This field could be useful if temporal analysis of the data is to be performed, for example to see how trends in words and topics change over time.

[1]Subtitle Edit exists both as an online service in the browser, and as a downloadable program for Windows. Both versions can be found here: https://nikse.dk/SubtitleEdit/

Even though the subtitle text itself is mostly clean text that can be directly used for text preprocessing, there are some additional technical metadata artifacts of the "Text" field that must be considered first. Firstly, Most lines of dialog contain a so-called alignment tag. This tag has the format `\{\anX\}`, where X is a number between 1 and 9. This tag is only used by the program reading the subtitles to know which part of the screen to display the subtitle on, and not shown on-screen. Secondly, to break and correctly wrap the subtitle lines when displaying them on screen, many lines contain `\n` ("new line") or `\r\n` ("carriage return, new line") characters. These are also not shown on-screen. Finally, some programs might have been subtitled live when they were created (called "live-captioning"), meaning a person or ASR system created the subtitles live while the program was being recorded. In these cases, the first line of the subtitle file after the initial metadata line might be a subtitle text saying "Opptak av simultanteksting" ("Recording of live-captioning"). This line is displayed on-screen for the viewer to see. Although there might be slight nuances in quality and form of sentences in the programs that have been live-captioned versus programs that have been subtitled "offline", this line does not provide other meaningful information about the program and associated subtitles in question.

4.2.2 NST Dataset Statistics

A brief statistical analysis of the NST dataset shows some interesting properties of the dataset, as seen in the first row of Table 4.2. Note that only 14,550 files is represented in this analysis, as there were parse errors when trying to read the remaining 64 files. First, we see that the average length of a subtitle file is about 204 lines. The smallest file has only 2 lines of subtitles, while the largest file has 2146 lines. As seen in the left histogram of Figure 4.1, most files are between 0 and 200 lines of length, while a large fraction of files have between 200 and 600 lines of subtitles. A very small fraction of files in the dataset have more than 600 lines.

It is also interesting to look at some statistics about the data on the token level. In this analysis, only the "Text" field of the subtitle files are considered, minus the first metadata line in each file. To get tokens from the subtitle text, tokenization is performed (see Subsection 2.2.1). After tokenizing the subtitles, noise from the subtitle metadata in the "Text" field must be taken into consideration, as discussed in Subsection 4.2.1. Because this metadata would clutter the analysis, it is removed (see Subsection 3.1.1 for how this is done). After extracting only the pure text part of the subtitles (what would be displayed on-screen), textPrep is used to generate statistics for the tokens in all of the subtitles. From the first row of Table 4.1, we can see that there are around 27.2 million tokens in the dataset. The size of the vocabulary, meaning the number of unique words, is 872,561. This means that the average terms in the vocabulary shows up about 31 times across the dataset. We can also see that the average number of tokens in each document is about 1870. Since the average line number of each subtitle file is 204 lines, this means that there are about 9 terms in each line. Finally, we can also see that the average number of stop words in each file is 898, which is 48% of all the tokens in each file/line. How textPrep finds Norwegian stop words in the text is discussed in Subsection 3.1.2.

In order to make the subtitles ready for the experiments, the pure subtitle lines of the "Text" field are extracted. Using Pandas, the CSV subtitle files are first inserted into DataFrames. This allows for easily extracting only the "Text" column, and removing the first line in each file that contains the metadata part. After extracting the relevant columns and rows, these modified DataFrames will be written to new CSV subtitle files, this time containing only the subtitle text. In addition, the subtitle metadata in the subtitle lines were cleaned, as mentioned in the last paragraph. These pure subtitle text versions of the NST dataset files are then used in the experiments of Subsection 4.6.1 and Section 4.4.

Dataset	Dataset size	Vocab size	Number tokens	Average token freq	Number tokens /file	Number stop words /file
Normal	14 550	872 561	27 211 612	31.19	1 870.21	897.91
Preproc	14 550	821 625	27 142 971	33.04	1 865.50	897.44

Table 4.1: Token statistics of the subtitle files in the full NST dataset ("Normal"), compared with statistics of the dataset that has been preprocessed to remove the subtitle metadata ("Preproc"). Although most numbers are relatively unchanged after preprocessing, the vocabulary is reduced by a noticeable amount (almost 6% in size), which accounts for alignment tags and line break characters being stripped off of the words they were previously connected to.

4.3 Preliminary NST Modeling Experiment

4.3.1 Creating an NST Sample

To examine if LDA and Top2Vec could give usable results on the NST dataset, a preliminary experiment testing out the models and their parameters is conducted. In order to limit processing time, a random sample of the dataset is extracted. A sample of 1000[2] subtitle files, almost 7% of the full NST dataset, seems to be a reasonable size, as the distribution of line numbers and the token average statistics are almost the same. This can be seen in Table 4.2, Figure 4.1 and Table 4.3.

In Table 4.3, it should be noted that the average token frequency has drastically changed, from 33.04 to 12.17. This is due to the vocabulary not shrinking at the same rate as the total number of tokens when a smaller sample of files is selected. This is expected, as less and less new tokens can be introduced the larger a corpus grows, as more and more tokens found in a new document will already have been introduced to the vocabulary by other documents. As preprocessing steps are performed in the preprocessing pipeline, the relative change in average token frequency will be more interesting than the absolute change. Thus, the changed average token frequency does not mean that this sample is not representative for the NST dataset. Henceforth, this sample with size 1000 of the NST dataset will be referred to simply as the *NST sample*.

Dataset	Size	Average	Median	Min	Max
Full	14 550	203.9	155	2	2146
Sample	1 000	199.6	149	8	1814

Table 4.2: Statistics of the subtitle files for the full NST dataset and the sample of NST. Each row shows the size of the dataset, as well as the average, median, min and max number of file lines

4.3.2 LDA Experiment

The NST sample is not given directly as input to the LDA model, `LdaMultiCore`. Instead of taking the subtitle files as input directly, a vocabulary of the tokens from all the subtitle files must first be generated. This corresponds to V as defined in Subsection 2.1.4, and is done using

[2]Due to an error extracting the sample, only 999 files were extracted in the sample used. As this error was not noticed until sometime later in the experiments and the margin of error is so negligible, the error was ignored in experiments, and the sample is for convenience considered to be of size 1000 in this text.

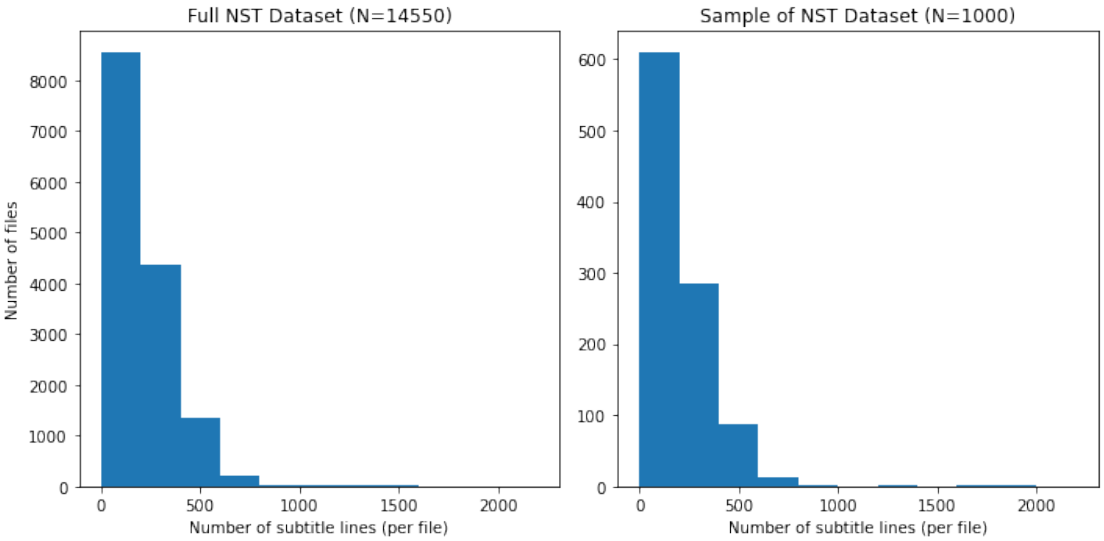


Figure 4.1: Histograms of number of lines in each subtitle file in the full NST dataset (left) and the sample of the NST dataset (right). Note that the scale of the Y-axis is different in the two figures. This is intentional to show the similarity between the datasets in terms of the distribution of relative file lengths.

Dataset	**Dataset size**	**Vocab size**	**Number tokens**	**Average token freq**	**Number tokens / file**	**Number stopwords / file**
NST	14 550	821 625	27 142 971	33.04	1 865.50	897.44
Sample	1 000	149 975	1 825 168	12.17	1 826.99	880.99

Table 4.3: Token statistics of the subtitle files in the full NST dataset, compared to the statistics of the sample from NST. Both datasets have been preprocessed to remove subtitle metadata.

the `Dictionary` class of the `corpora` module in Gensim. Then the vocabulary is used to convert the subtitle files into Bag-of-Words (BoW) format[3] using the `doc2bow` function of `Dictionary`. When a subtitle file is turned into a BoW, the original ordered lists of the tokens in the subtitles are given IDs and counted. Then, only a list of token IDs and their frequency is kept while the whole original tokenized subtitle list is discarded. For the purpose of topic modeling with a generative probabilistic model, though, using BoW-formatted documents is equivalent to using the original documents and will give the same results. The vocabulary and BoW-formatted subtitles are given as parameters to the LDA model, along with the number of topics k to be created. This corresponds to the inputs of Equation 2.6.

The process of selecting the optimal k for LDA is a non-trivial process as mentioned in Subsection 2.1.3, often using computationally expensive and time-consuming estimation methods. Due to time and scope constraints, use of this kind of estimation is not done. Besides, a few test runs with different size of k, where the value ranges from 2 up to 50, shows that there is very little change in the topics created and what words they contain. So, instead of focusing capacity on estimation of k, k is selected through a much simpler approach: Using the number of topics found by Top2Vec. As Top2Vec automatically finds the appropriate number of topics, based on the natural clusters it finds after having learned embeddings, it can be assumed that this number of topics is a good approximation to the number of topics that would be optimal for LDA too. Besides, this allows for a more direct comparison of the results, as the number of topics found by both models will be the same. As can be seen in Subsection 4.3.3, Top2Vec finds 8 topics when run with the chosen parameters on the NST sample. Therfore, $k = 8$ is given as the number-of-topics parameter to `LdaMultiCore`.

After `LdaMultiCore` has been called with the parameters described above, the topics are returned. These are returned as tuples of the shape (topic ID, topic words). The topic ID is just a sequence number in the list of topics, while the topic words are themselves represented as a linear combination of the products of each topic word together with their probability for appearing in the topic, that is $P(w_{j,c}|t_j)$.

4.3.3 Top2Vec Experiment

The model of `Top2Vec` has two required parameters: the collection of documents, `documents`, and the embedding model to use, `embedding_model`. The NST sample is given as argument for the `documentsparameter`. Each subtitle file is given as one, long continuous string, as Top2Vec works on full-text documents. For the `embedding_model` parameter, there are four choices:

1. `doc2vec`
2. `universal-sentence-encoder`
3. `universal-sentence-encoder-multilingual`
4. `distiluse-base-multilingual-cased`

One model already sticks out above the others before any experiments have been run: the `distiluse-base-multilingual-cased` model. This model was recommended by Birkenes [2022] as NLN have performed experiments with the model on Norwegian texts in the past and achieved good results. However, to be sure each model is tested to see what results they produce.

In addition to these two required parameters, there are a number of optional parameters that can be tweaked. Due to time constraints, most of these were not explored. Their default

[3]Bag-of-Words model: https://en.wikipedia.org/wiki/Bag-of-words_model

values mostly seemed reasonable, and changing these parameters was either not relevant to these experiments or did not seem to be critical for model performance. However, two optional arguments were changed: `min_count` and `workers`. The first argument specifies the minimum count a word must have for it to be included in the embedding, and this argument is set to 50 by default. However, the `Top2Vec` API recommends setting `min_count` to a lower value for smaller corpora, so it is set to 5 here. Meanwhile, the `workers` allows for selecting how many parallel threads that the `Top2Vec` will run in when training the model. This argument does not affect the results of the model, but can speed up the training significantly if set to a high enough number. Thus, `workers` is set to a number 2 less than the number of CPUs on the computer, to maximize speedup while avoiding consuming all available resources on the computer.

The `Top2Vecmodel`, similarly to LDA, returns topics as tuples of the shape (topic ID, topic words). Here, topic words are listed with their cosine similarity to the topic vector of the topic they're in, that is $\cos(t_j, w_{j,c})$.

4.3.4 Creating Word Clouds from Topics

After getting the topic lists as results from both LDA and Top2Vec, they must be rendered as the word clouds mentioned in Section 3.3. Here the `WordCloud` class of the WordCloud library is used. Each topic found is fed into the word cloud generator with the 10-20 words with the highest topic representativeness and associated representativeness scores, so that each word can be printed with size proportional to its topic representativeness. In order to reduce the amount of visual clutter, topic words are only printed horizontally, and are all printed in the same color. Using only one color per word cloud has the added benefit of making word clouds of two different topics visually distinguishable by printing them in two different colors.

4.4 NST Preprocessing Experiment

After the initial experiments with LDA and Top2Vec that confirms that the topic models work on the NST dataset, the NST preprocessing experiment begins. Here, the preprocessing pipelines briefly mentioned in Section 3.1 are implemented and used to preprocess the datasets, so that both intrinsic and extrinsic data quality can be measured and compared across datasets, and to see if better performance can be achieved. Even though it would be natural to start using the full NST dataset in this experiment, due to limited by computational resources and time constraints, the dataset used here is still the NST sample.

4.4.1 Extension to textPrep

As discussed in Subsection 2.4.1, the analysis of the effects of preprocessing methods on data used for topic modeling is a largely unexplored research topic in the field of topic modeling. Therefore, to improve the results of later experiments and provide a contribution to this sub-field of topic modeling, experiments on the effects of preprocessing will be performed as part of this research project. My main contributions will be to examine some areas which Churchill and Singh did not properly examine:

- The effects of preprocessing on distributed representation models, with Top2Vec as test subject
- If preprocessing rules have different effects on model results when the documents are longer and with more verbal language than social media texts

- To see if the textPrep toolkit can easily be extended for languages other than English

4.4.2 The Preprocessing Pipelines

In these experiments, the NST dataset will be run through a set of different preprocessing pipelines, each consisting of a set of modular preprocessing rules, and the resulting dataset will be assessed both for intrinsic data quality and for extrinsic quality, that is the effects on the quality of topic model results. The measures used here will be the same as in textPrep: intrinsic quality is measured with token statistics and extrinsic quality is measured with topic coherence and topic diversity on both the LDA and Top2Vec results for each dataset variation. A combination of existing rules, modified rules and new rules are used to build the pipelines, as outlined in Section 3.1. Six pipelines are made, all of which will be listed and described below, in rising order of amount of preprocessing steps. Unless otherwise specified, all the pipelines are stacked on top of each other, so the next pipeline takes the preprocessed data of the last pipeline as input, and so on (see also Table 4.4 for mapping between rules and pipelines):

1. **Raw pipeline:** This pipeline takes the NST sample as input, and performs only punctuation removal (as well as the subtitle metadata removal that was already done to the NST sample). This could be considered the absolute minimum preprocessing that is done on the dataset in a generative probabilistic topic modeling context, so it is as good as raw. This mirrors the "Baseline 1" pipeline of Churchill and Singh [2021b], and is a baseline dataset in this experiment along with the unprocessed NST sample.

2. **Basic pipeline:** Here, the capitalization normalization, number removal and stop word removal is added to the raw pipeline. This corresponds closely to the "Lightweight" configuration in Churchill and Singh [2021b], except that this pipeline does removes numbers and subtitle metadata instead of URLs.

3. **Lemmatized pipeline:** Lemmatization is added as a step in this pipeline. This is studied as a separate pipeline to see if lemmatization alone drastically alters the average frequency of tokens.

4. **Lemmatized + TF/TF-IDF-cleaned pipeline:** Term frequency (TF) and TF-IDF-cleaning is added in this pipeline. Terms with $TF > 1000$ (the top 20 most common terms in the lemmatized version of the NST sample have $TF > 1000$) and $TF-IDF < 0.5$ (terms that appear a little in many documents or rarely appear at all) are removed. This cleaning is inspired by the TF-IDF-cleaning experiment done in Churchill and Singh [2021b].

5. **PoS-cleaned (verbs) pipeline:** Here, Part-of-Speech-tagging is performed on the dataset before further preprocessing. Then, all terms that are tagged with the `VERB` tag are removed. This is based on the hypothesis that verbs will appear as topic verbs too often over nouns, even though nouns are thought to be more valuable due to more semantic meaning.

6. **PoS-cleaned (only keeping nouns) pipeline:** Similarly as in the previous pipeline, PoS-tagging is performed here. But instead of only removing verbs, this pipeline removes all terms that are not tagged as `NOUN` (common nouns) or `PROPN` (proper nouns). Again, this is based on the hypothesis that nouns will be most useful as topic words, as they are the words that carry the most semantic meaning. So if only nouns are left, the assumption is that this dataset must provide the best word clouds.

4.4.3 Pipeline Evaluation

After a pipeline has created a preprocessed version of the NST sample, token statistics will be generated from the dataset. These measures are the same as used in Table 4.3. Then, their datasets are sent to the topic models. Most of these pipelines and the datasets they produce are thought and intended to improve the results for LDA, as this model often require some preprocessing to be effective. These datasets are not expected to improve upon Top2Vec, as this model is made for full sentences and might degrade as soon as sentence structure is lost through heavy preprocessing. However, Top2Vec will also be run with these pipelines to verify if this hypothesis is right. Both LDA and Top2Vec will be run with the parameters found to be fitting in Section 4.3.

After the models have been run on all the datasets, topic coherence and diversity will be calculated for each model-dataset pair. Topic coherence is as used in Churchill and Singh [2021b], the average of Normalized Pointwise Mutual Information (NPMI) score for all topics in the model. Thus, coherence of a model M, $C(M)$ is defined as follows:

$$C(M) = C(T) = \frac{\sum_{t_j \in T} NPMI(t_j)}{|T|} \tag{4.1}$$

with NPMI being defined as:

$$NPMI(t_j) = \frac{\sum_{x,y \in t_j} \frac{\log \frac{P(x,y)}{P(x)P(y)}}{-\log P(x,y)}}{\binom{|t_j|}{2}} \tag{4.2}$$

NPMI can be between minimum -1 and maximum 1 in score, that is $-1 \leq NPMI(t_j) \leq 1$. A score of -1 means words never appear together in document, they only appear separately from another. Meanwhile a score of 1 means each word always appear together with all the other words, when one of them appear. As coherence is a normal average of NPMIs, we also have $-1 \leq C(M) \leq 1$.

Topic diversity is also the same measure as used in Churchill and Singh [2021b]. That is, the topic diversity of a model M for a given top c topic words, $D(M, c)$, defined as the fraction of unique words among the top c topic words, $W_{top}(T, c)$. In other words, the diversity is the cardinality of the intersection of the top c topic words, divided by the cardinality of the union of the same set:

$$D(M, c) = D(T, c) = \frac{|\bigcap_{w \in T'} w|}{|\bigcup_{w \in T'} w|} \tag{4.3}$$

with $T' = \{t'_1, ..., t'_j, ..., t'_k\}$ where t'_j is defined as the top c words in each topic t_j. Diversity is expressed as a fraction between 0 and 1, that is $0 \leq D(M, c) \leq 1$. A diversity score of 0 means no diversity - all top c topic words in each topic are shared across all other topics, and a score of 1 means full diversity - no words among the top c ever occurs as a top c word in another topic.

4.5 NST Topic Interpretation Experiment

After the NST preprocessing experiment of Section 4.4 is completed, a topic interpretation experiment will be conducted to validate the results of the best LDA and Top2Vec models from the preprocessing experiment, to see if they really are usable by humans. This will also serve to provide more nuanced evaluation of the topics, as Chang et al. [2009] showed that human evaluation metrics can prove to give quite different results than automatic metrics. The models

Rule	**About rule**	**Raw**	**Basic**	**Lem**	**Lem + TF**	**PoS verbs**	**PoS nouns**
Metadata	3.1.1	X	X	X	X	X	X
Punctuation	3.1.1	X	X	X	X	X	X
Numbers	3.1.1		X	X	X	X	X
Normalization	3.1.4		X	X	X	X	X
Stop words	3.1.2		X	X	X	X	X
Lemmatization	3.1.3			X	X	X	X
TF	3.1.4				X	X	X
TF-IDF	3.1.4				X	X	X
PoS: rem. verbs	3.1.3					X	X
PoS: keep nouns	3.1.3						X

Table 4.4: Preprocessing rules (rows) and which pipelines each rule are used in (columns).

to be used in this survey will be picked based on what LDA and Top2Vec models have the best combined topic coherence and diversity score. More precisely, the models with the best harmonic mean of coherence and diversity (with coherence normalized) $H(M)$:

$$H(M) = H(T) = \frac{\frac{1+C(T)}{2} + D(T)}{2} \tag{4.4}$$

Once the topic models are chosen, the combined set of topics from the two models are integrated into the survey. The structure of the survey is as described in Section 3.4. The tests used in my survey will however be altered. For example, the word intrusion test is dropped, as this test seems to be too trivial in some cases to provide accurate topic coherence measures. Instead, interpreted topic coherence is measured by asking users to select all topic words out of a selection of the highest probability words that they feel belong to the topic. By seeing how large the fractions of selected words are, it is possible to infer human-interpreted topic coherence in a more nuanced way. The topic intrusion test however seems more meaningful, as it is often not trivial, and achieving high human-interpretability here should be more of a challenge requiring high accuracy, as interpreted by a human, on the topic-document assignments. However, as this test is only adapted for mixture models such as LDA, that assign multiple topics to a document, a different approach must be used for singular models such as Top2Vec, that only assigns one topic per document. An inverse variant, where the user is accepted with four topics and a document, and only one of the topics belong to the document while the rest are intruders, is a viable option.

The topics are chosen at random to be included in each part of the survey, to minimize the risk of bias with regard to which topics are used in which questions. The word clouds of both the LDA and Top2Vec models are also mixed together and put in a random order (with an equal number of word clouds from both), so the user is unaware that the word clouds are generated by two different models. This is done to ensure unbiased evaluation of both sets of word clouds with regards to where they originate from, so it is possible to do a fair comparison of their results afterwards. Similarly, TV programs to be used for the different tasks are chosen at random from the set of programs that have been assigned each topic. Then, for each question, the word cloud representation of the topics are inserted as images. The order of alternatives in the topic intrusion task is ordered randomly for each participant, to minimize the risk of bias in the ordering of correct and wrong answers. Finally, once the survey has been created, it is sent to NRK employees to answer. The different sections of the survey will be presented below.

The survey is in Norwegian, but brief summaries and translations will be given in English. The survey in its entirety as presented to users can be found in Section B.

4.5.1 Survey Introduction

First, the survey and its purpose is introduced to the user, as can be seen in Figure 4.2. After a short statement about NRK wanting to improve their structured metadata, the survey goes on to present presents the topics from the project as word clouds in order to avoid confusion with existing topic definitions in NRK. It is however made clear that these word clouds aim to describe topics in TV programs. An example is shown further down on the page. Then, the potential future applications of the topics/word clouds in a publishing context is expressed, and that these topics must be evaluated to assess what work needs to be done to get there. Finally, the introduction explains that there are a number of word clouds that have been made from a sample of 1000 subtitle files of a wide range of TV programs, and that the word clouds try to find the words most representative of the topics, with bigger words being more important.

Before the user starts the word cloud survey, they are then asked a question about what type of role/position they have in NRK (Figure 4.3. The responses to this question will be used to see how employees with different roles in a broadcasting corporation might interpret and evaluate topics differently.

4.5.2 Part 1: Individual Word Cloud Interpretation

The first part of the survey, as briefly discussed in Section 3.4, presents one word cloud at a time to the user, and asks them to answer several questions related to this word cloud. There are a number of pages in this part with one wordcloud each, that all follow the same format presented here. The same number of word clouds are taken from LDA, as there are from Top2Vec. All word clouds are sampled randomly from the selected topic models, and presented in a random order in the first part of the survey, although all respondents will see the same order. Each page starts with presenting the word cloud is, after which users are asked to give a name to the word cloud. An example word cloud along with this question is seen in Figure 4.4. The answers to this question should help to highlight topic coherence, as name suggestions should be more similar to each other if topics are more coherent, and vice versa.

The next question, which can be seen in Figure 4.5 asks the user to select which of the topic words from the word cloud they feel belong to the topic. based on the topic they could interpret from the word cloud. This question too should highlight how coherent the topic is, as a more coherent topic should have more topic words that are easy for the user to relate to the underlying topic, and the words should match well across different responses. If the topic is very incoherent, however, this should be indicated by the user not being able to find any words that relate to the unclear topic, or might select random words. In this sense, this question touches upon the same interpretability aspects as the word intrusion test in Chang et al. [2009]. Both tests asks users to select what words belong or not, with some words having a higher topic representativeness than others, and should have similar divergence in responses the less coherent a topic is.

Then, the user is presented with two NRK TV programs connected to the topic, and asked to select if one, both or none of the programs are accurately represented by the topic of the word cloud. This question is seen in Figure 4.6(a). Whereas the previous questions examined the coherence between topic and topic words, this question is meant to examine the relation between topics and programs. If a topic is coherent with the programs it is strongly representative of, the user should be able to easily interpret that and select programs that have a high topic-program representativeness score more often than those that do not.

Spørreundersøkelse om automatisk genererte ordskyer fra TV-programmer

Mitt navn er Magnus og jeg skriver nå en masteroppgave i datateknologi ved NTNU i samarbeid med NRK. Oppgaven handler om hvordan man kan tilføre NRKs TV-programmer god og strukturert metadata ved hjelp av maskinlæring.

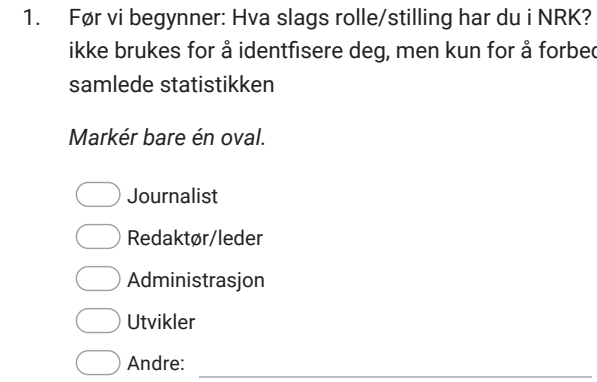
I den anledning ønsker jeg å svare på om et system jeg har laget klarer å hente ut meningsfulle og presise "temabeskrivelser" om programmene. Disse "temabeskrivelsene" har form som ordskyer med emneord som er hentet fra TV-programmenes undertekst, der ordskyene forsøker å finne et eller flere temaer i programmet. Se bildet under for eksempel. NB: Disse ordskyene kan skille seg noe fra eksisterende definisjoner og bruk av "tema", "saksunivers", "kategorier", osv. internt i NRK.

Tanken er at disse ordskyene i framtiden skal kunne brukes av journalister og andre som jobber redaksjonelt i NRK som et hjelpeverktøy i publisering, slik at de kan skaffe seg bedre oversikt over tema som går igjen på tvers av innhold og for å lettere finne TV-programmer, radioprogrammer og/eller nyhetsartikler som henger sammen med hverandre tematisk. Men for å kartlegge om disse ordskyene er verdifulle, så trenger jeg tilbakemelding fra relevante brukere på hvor presise og meningsfulle ordskyene er som temabeskrivelser for innholdet de omfatter.

I denne undersøkelsen har systemet mitt generert 14 ordskyer ut fra en samling på undertekster fra 1000 TV-programmer i NRK TV. Programmene dekker et vidt spekter av tema og sjangre, og inneholder alt fra nyhetsprogrammer til underholdningsprogrammer. Ordskyene som systemet mitt lager har ikke navn, men består alle av et antall emneord som systemet har plukket ut for å forsøke å beskrive helheten av temaet. Et emneord i ordskyen har større skriftstørrelse jo mer representativt systemet tror det er for temaet. Med fire av ordskyene følger det noen spørsmål som ønskes besvart, og til slutt er det en seksjon hvor de resterende ordskyene og TV-programmer skal matches. Det er fint om du svarer på alle spørsmålene etter beste evne, men du kan også la spørsmål stå ubesvart om du ikke vet hva du skal svare.

Eksempel på ordsky:

Figure 4.2: Introduction to the user survey.



1. Før vi begynner: Hva slags rolle/stilling har du i NRK? Denne informasjonen vil ikke brukes for å identfisere deg, men kun for å forbedre informasjonen i den samlede statistikken *

Markér bare én oval.

- Journalist
- Redaktør/leder
- Administrasjon
- Utvikler
- Andre: _____

Figure 4.3: Question about role of user in NRK. The options are "Journalist", "Editor/manager", "Administration", "Developer" and "Other", the latter category of which allows the user to write their own role.

Finally, each page asks how useful the user thinks the word cloud is for describing topics in programs, and if they have any comments. This can be seen in Figure 4.7. The usefulness question is meant to provide some conscious and direct feedback on the interpretability of the word clouds, to add to the results of the other questions. The free text part will not be a central part of the analysis in the discussion, but responses here will be highlighted as potential focus areas for future work.

Ordskyens navn

Systemet mitt kan ikke automatisk gi navn til ordskyene den finner. Det kan kun bruke ulike ord den finner i underteksten på tilhørende TV-programmer (emneordene som er vist i ordskyen) som et forsøk på å beskrive temaet. Hvilket navn tenker du er dekkende for ordskyen?

Ordet/-ene du bruker kan komme fra ordskyen over, men trenger ikke å være det. Dersom du ikke klarer å finne noe passende navn for ordskyen, la feltet stå blankt.

2. Hvilket navn vil du gi ordskyen?

Figure 4.4: Question about the name of the topic. The user is instructed to give a name that they believe is accurate and covers the topic well. They are also told that they can choose to use terms from the word cloud or not, and asked to leave the field blank if they struggle to find a name.

Ord som hører hjemme i ordskyen

Dersom det er tydelig for deg hvilket tema det er som ordskyen prøver å beskrive, og det var mulig for deg å sette et navn på ordskyen, hvilke emneord mener du passer inn i ordskyen? Kryss av for alle emneord du synes hører hjemme under ordskyen.

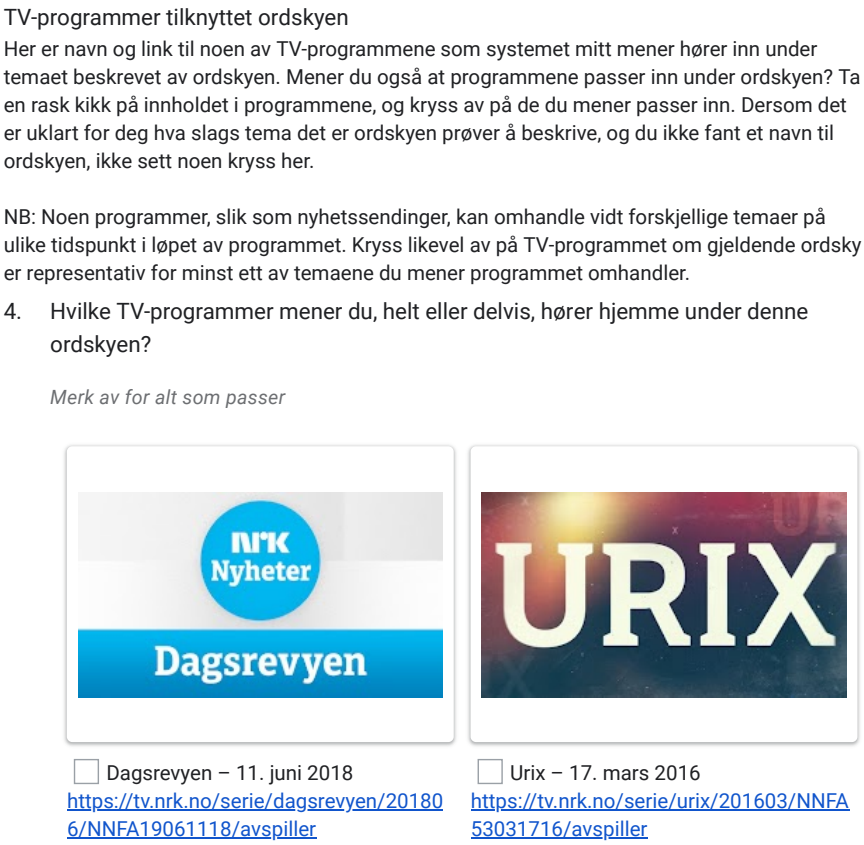
Hvis det derimot er uklart for deg hva slags tema det er ordskyen prøver å beskrive, og du ikke fant et navn til ordskyen, ikke sett noen kryss her.

3. Hvilke ord hører hjemme i ordskyen?

Merk av for alt som passer

- ☐ syriske
- ☐ syrisk
- ☐ erdogans
- ☐ siri
- ☐ erdogan
- ☐ kurdiske
- ☐ tyrkere
- ☐ islamistiske
- ☐ kurdisk
- ☐ afghanske
- ☐ irakiske
- ☐ taliban
- ☐ tyrkerne
- ☐ migranter
- ☐ migrantene
- ☐ kurdernes
- ☐ istanbul
- ☐ tyrkiske
- ☐ kurderne
- ☐ gaddafi

Figure 4.5: Question about which topic words belong in the word cloud (topic). The user is instructed to choose one or more words from the word cloud that they perceive as coherent with the interpreted topic, also based on the name they gave the topic. If the user struggles to interpret the topic is, and could not find a name for the topic, they are asked to leave all words unmarked.



(a) Question about which programs are well-represented by the word cloud. They are given a title, image and link to the NRK TV website where the specific programs can be opened and played. Users are asked to quickly skim through the video to get an idea of the topics that could be present in the program, and then asked to come back to the survey and reply to the question. They are also informed about the possibility that some programs, especially news broadcasts, have many different topics that change throughout the duration of the program. If the user finds any part of the program to be about the topic they interpret from the word cloud, they should mark the program as relevant either way. They are also asked to leave the question unanswered if they have not been able to interpret the topic of the word cloud.

5. Hvor nyttig synes du denne ordskyen er for å beskrive et tema i et program? *

   Markér bare én oval.

	1	2	3	4	5	
Svært lite nyttig	◯	◯	◯	◯	◯	Svært nyttig

6. Har du noen kommentarer om ordskyen?

Figure 4.7: Question about the usefulness of the word cloud to describe a topic in a program (linear scale from 1="Not useful at all" to 5="Very useful"), as well as free text question for commenting on the word cloud.

4.5.3 Part 2: Word Cloud Intrusion Tests

Once the user has gone through some individual word clouds and responded to questions about these in part 1, they will now meet multiple word clouds in each question in what is called "word cloud intrusion tests". These are partially or fully inspired by the topic intrusion tests of Chang et al. [2009], depending on which model the question is about. Even though there are questions with both LDA and Top2Vec here, the models cannot be mixed in the tasks here, and each topic model has slightly different, but related tasks the user must perform. There are two pages here, and the first has questions about LDA, while the second has questions about Top2Vec.

The first page deals with a word intrusion test that is almost identical to the topic intrusion test of Chang et al. [2009]. The introduction to this part can be seen in Figure 4.8. An example question from this page is presented in figure Figure 4.9. As the word intrusion test seemed to work well for identifying interpretability of topics, I include the same test here. In addition to being a good measure for topic-document coherence, The test should also be a good measure for topic diversity, as less diverse topics will be harder to separate from each other and from the program. So the more diverse the topics are, the easier it should be for the user to identify the correct intruder.

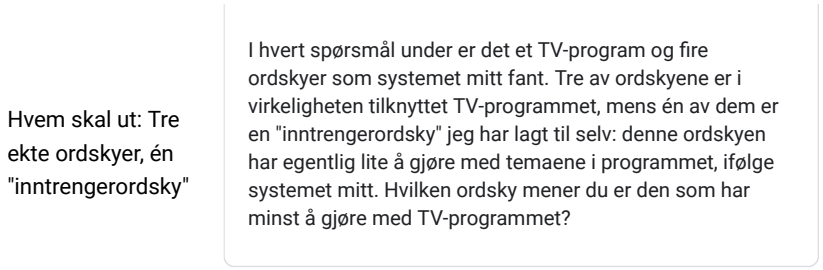


Figure 4.8: The introduction to the first word cloud intrusion test of part 2, which unbeknownst to the user includes only the LDA topics. The user is told that they will be presented with four word clouds and a TV program. Three of these are representative of the topic according to the LDA model, while the last one is an intruder word cloud that is not representative and linked to the program by me. The user is asked to pick the word cloud that they think is the "intruder", the one who does not belong to the program. This is very similar to the topic intrusion test of Chang et al. [2009].

The second page then deals with a different variant of the topic intrusion test. An example question from this page is presented in figure Figure 4.10. As this test is meant to test the topics of the Top2Vec model, and the Top2Vec model only has one topic assigned to each program, a different approach must be used. Therefore, the intrusion test is inverted: Instead of one word cloud being the "intruder", three of the word clouds are "intruders", and the remaining one word cloud is the only one Top2Vec has found to be representative of the program. Even though the task is inverted, it should still be a good measure for topic-document coherence and topic diversity: If topics are similar to each other and the correct topic is not very coherent with the program, the user should still struggle to find the right topic.

Figure 4.9: The user is presented with the title and an image of a TV-program, along with an NRK TV link to the program. Below are four word clouds, three of which belong to the program and one who does not. In a similar manner to the questions in part 1, they are asked to skim through the TV program, identify topics and deduce which of the word clouds are not representative - the "intruder".

Figure 4.10: The user is presented with the title and an image of a TV-program, along with an NRK TV link to the program. Below are four word clouds, only one of which belongs to the program and three who do not. Similarly as in the other word cloud intrusion task, they are asked to skim through the TV program, identify topics and deduce which of the word clouds are not representative - the "intruders".

4.6 Results

Here, the results from all the three experiments discussed in the previous sections will be presented. The discussion of said results can be found in Chapter 5.

4.6.1 Preliminary NST Modeling Experiment Results

After running `LdaMultiCore` with the NST sample and $k = 8$ as described in Subsection 4.3.2, the 8 topics (represented as word clouds with top 10 topic words of each topic) found can be seen in Figure 4.11.

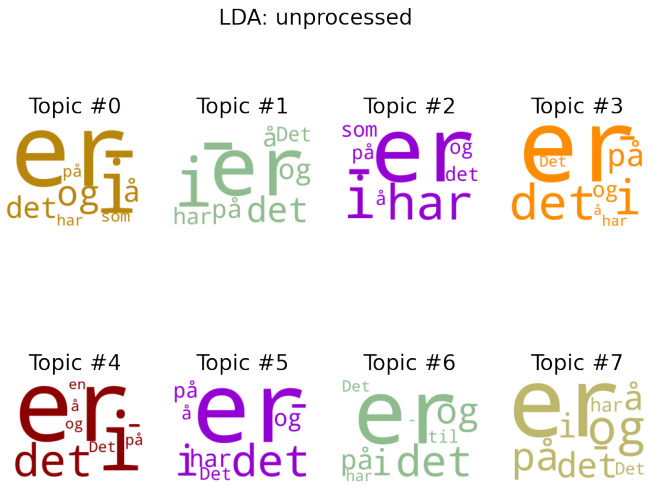
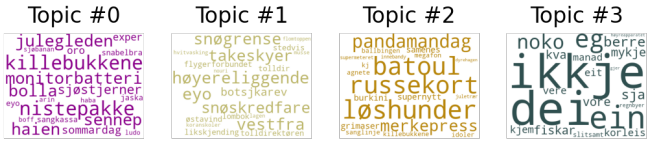


Figure 4.11: The 8 topics found with LDA using the (unprocessed) NST dataset and $k = 8$. Each topic word cloud displays the top 10 topic words in each topic, with size proportional to the probability that the word appears in the topic.

Meanwhile, Top2Vec had to be tested with the different embedding models mentioned in Subsection 4.3.3. First, `doc2vec` was tried (see Figure 4.12). Next, `universal-sentence-encoder` and `universal-sentence-encoder-multilingual` was tried out. These were not expected to work, as neither of them are trained for Norwegian. The `universal-sentence-encoder` did surprisingly work, and the results can be seen in Figure 4.12. On the other side, the `universal-sentence-encoder-multilingual` model did not work. Lastly, it was time to test the `distiluse-base-multilingual-cased` model. The resulting topics of that model can be seen in Figure 4.13.

The discussion about the results of these preliminary experiments and the consequences they have for the later experiments is found in Section 5.1. For convenience while reading this chapter, the result of the discussion there is that both methods work and show promise, so they are expanded upon in the next experiments. LDA with the same parameter of $k = 8$ and Top2Vec with the embedding model `distiluse-base-multilingual-cased` are the configurations that are used in the rest of the experiments.



(a) Top2Vec found 4 topics when the embedding model `doc2vec` was used.

(b) Top2Vec found only 1 topic when the embedding model `universal-sentence-encoder` was used.

Figure 4.12

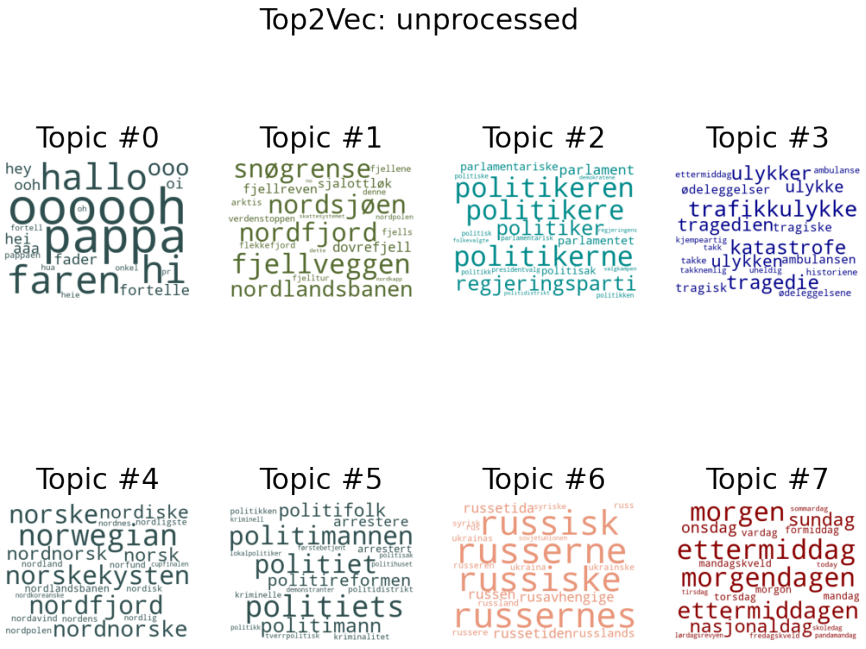


Figure 4.13: Top2Vec found 8 topics when the embedding model of `distiluse-base-multilingual-cased` was used. This is the embedding model that is used henceforth, which is why LDA has 8 topics: 8 topics was automatically found by this Top2Vec variant, and using the same number of topics thus gives the most fair comparison of the two models.

4.6.2 NST Preprocessing Experiment Results

After the results of preliminary experiment as discussed in the section above, and the discussion of the results in Section 5.1, LDA and Top2Vec are both used for the NST Preprocessing Experiment. In this experiment, all the pipelines of Subsection 4.4.2 are run on the NST sample using the textPrep toolkit, and the preprocessed datasets are written to files. Then, dataset token statistics are run on all the datasets using textPrep. The results of that statistical analysis of the dataset can be seen in Table 4.5. This analysis gives an indication of the intrinsic data quality that the datasets in themselves have.

Dataset	**Vocab size**	**Number tokens**	**Average token freq**	**Number tokens /file**	**Number stop words /file**
unprocessed	149 975	1 825 168	12.17	1 826.99	880.99
raw	95 757	1 777 144	18.56	1 778.92	971.93
basic	81 649	604 211	7.40	604.82	0.00
lemmatized	64 426	610 326	9.47	610.94	0.00
lem_tfidf	17 784	498 286	**28.02**	498.78	0.00
pos_verb	16 296	395 655	24.28	396.05	0.00
pos_noun	13 753	288 623	20.99	288.91	0.00

Table 4.5: Token statistics (see Subsection 4.2.2 for more details on what each stat means). As is to be expected, vocabulary size, total number of tokens, average number of tokens per file and average number of stop words per file shrink the more heavy the preprocessing is done in the pipelines. Also, stopwords are completely gone from every pipeline after the basic one, because this one removes all stop words. One interesting statistic is average token frequency. This is a statistic that is often maximized in topic modeling, as it means more words show up often which will be easier to identify as words central to the topic of one or more documents. In this regard, the lemmatized+TF+TF-IDF dataset has the best properties for topic modeling.

Dataset	**Coherence**		**Diversity**	
	LDA	**Top2Vec**	**LDA**	**Top2Vec**
unprocessed	0.434	0.088	0.163	0.957
raw	0.443	0.228	0.212	0.944
basic	0.806	0.563	0.300	**1.000**
lemmatized	0.712	**0.618**	0.225	**1.000**
lem_tfidf	0.909	0.523	**0.512**	0.929
pos_verb	0.913	0.336	0.400	0.917
pos_noun	**0.951**	−0.252	0.375	0.650

Table 4.6: The topic coherence and topic diversity of each model-dataset pair. The score of the dataset with the best metric for each metric and dataset, is highlighted in **bold**.

After that analysis, LDA and Top2Vec performs topic modeling on all of the datasets in turn. Note that both models keep their configurations fixed throughout all the preprocessing experiments (see Section 5.1 for the configurations selected). This is to ensure the changes seen in the models from pipeline to pipelines are only caused by the differences in preprocessing steps, and not by internal model factors. This also means that the number of topics for LDA are kept constant, at $k = 8$, throughout all the experiments, even though this might lead to the number

of topics found by either model to diverge. However, the main focus on comparison of the two models lies in the Section 4.3 and the Section 4.5, whereas this experiment will only focus on how each model individually is affected by the preprocessing steps.

The topic models that are generated for for each preprocessed dataset, and the topics they produce, are in turn used to generate collections of word clouds as was done in Subsection 4.6.1. The collections of word clouds generated from each dataset-model combo can be found in the appendix Section A. The raw topic sets generated by each model are also used to calculate topic coherence and topic diversity, as was discussed in Section 4.4. This provides a measure of extrinsic data quality, in addition to the intrinsic measure. The topic coherence along with diversity of the models have been plotted in Figure 4.14. As discussed in Section 4.5, the LDA and Top2Vec model with the highest harmonic mean, $H(M)$, among its peers would be selected to use in the topic interpretation experiment of Section 4.5. This will be further discussed in Section 5.2, but for convenience for the reader, the conclusion is that the LDA-lem_tfidf model and the Top2Vec-basic models were chosen for use in the survey.

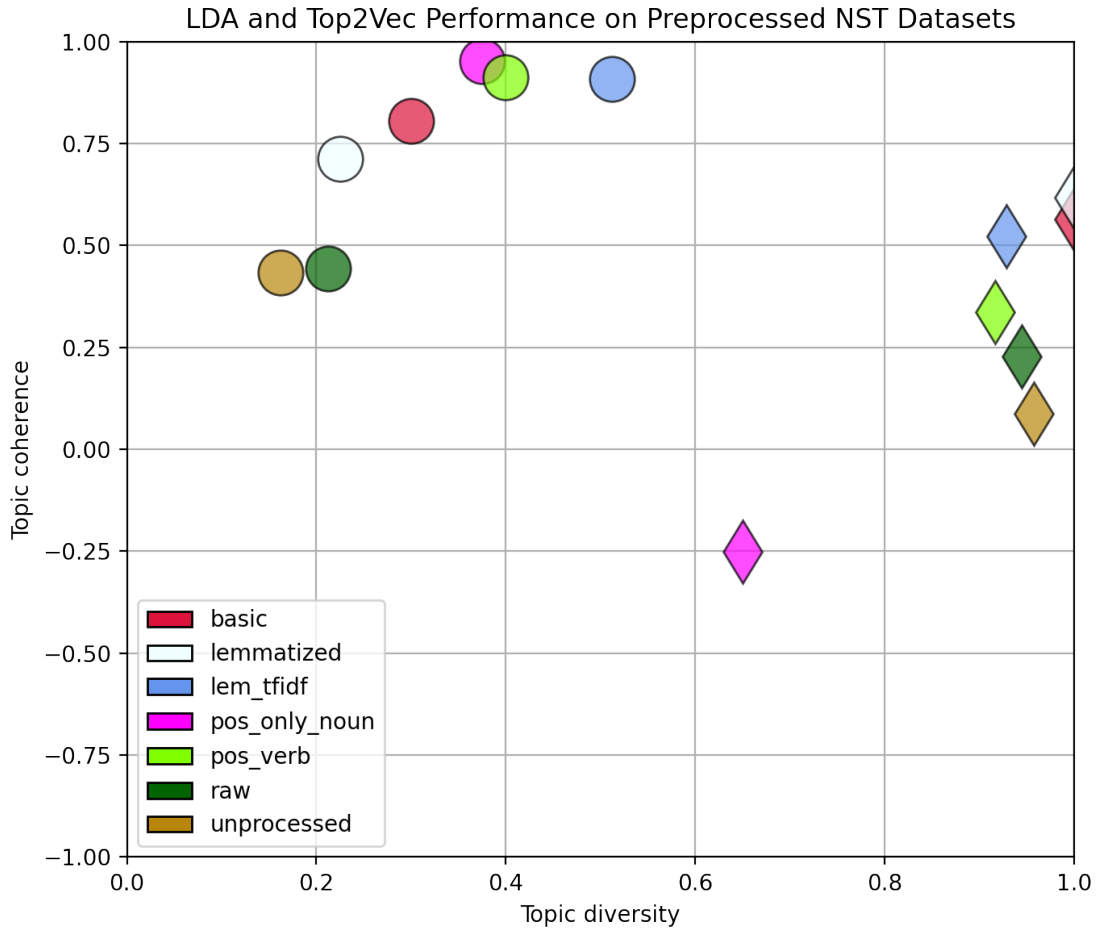


Figure 4.14: Plot of topic diversity (X-axis) and topic coherence (Y-axis) for each of the topic model and preprocessed dataset pairs. Circles represent LDA models, while diamonds represent Top2Vec models. The closer a data point is to the upper right, the better the total model performance is.

4.6.3 NST Topic Interpretation Experiment Results

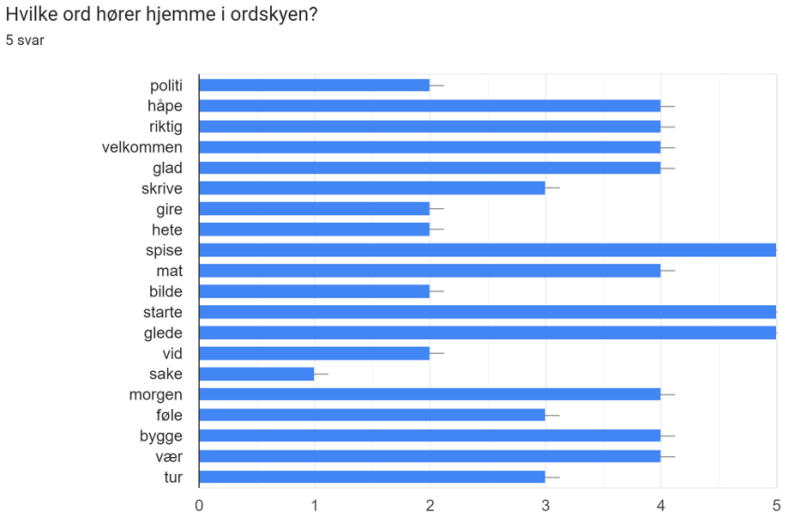
Based on the discussion in Subsection 5.2.3, the LDA-lem_tfidf model and the Top2Vec-basic models are selected for further use in the user study that was sent to NRK employees. Due to time constraints, the survey was only open for two days, which limited response time for NRK. 10 responses to the survey were collected. Statistics of the responses are gathered below.

First, the responses to the question about occupation/role in NRK showed that 6 out of 10 respondents are journalists, editors or managers (with editorial responsibility), as shown in Figure 4.15. This means that a reasonable fraction of the respondents are in the intended target group of employees in NRK who work with publishing.

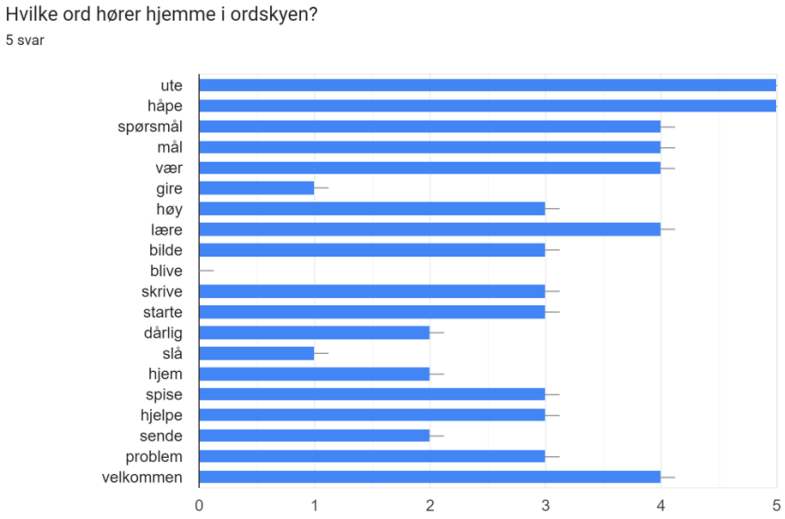
Figure 4.15: Responses to the question about occupation/role in NRK. 6 out of 10 are journalists/editors/managers, while the remaining 4 have other occupations. The specific occupation of the respondents have been generalized to preserve anonymity.

The results from part 1 of the study are presented next. This is the part about individual word clouds and their properties. This part had four word clouds, two of which came from LDA-lem_tfidf and two of which came from Top2Vec-basic. The specific word clouds were topic #4 and #5 of Top2Vec-basic (see Figure 10), and word clouds #0 and #3 of LDA-lem_tfidf (see Figure 5). The statistics of each word cloud below are presented so they are grouped according to topic model for clarity, not necessarily in the same order that they were in the survey. The results for the LDA word clouds are presented together first, then the results for the two Top2Vec word clouds, for each question.

With that, the presentation of results from part 1 of the survey is concluded. There was also a free text field on each page, but the responses from these will be collectively assessed and discussed in the discussion. Now, part 2 will be presented. Here, the two tasks of this part were already split between the word clouds of LDA-lem_tfidf and Top2Vec-basic. The results from the first task and first page of part 2, which was the word cloud intrusion task where only one word cloud was an intruder, deal with word clouds belonging to LDA-lem_tfidf. The results are shown below:

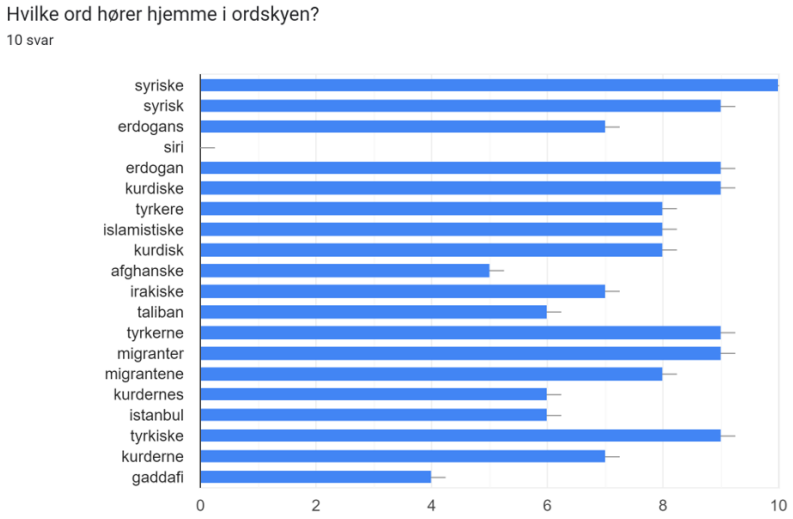


(a) This bar chart shows how many users selected each of the terms to be a part of the word cloud #2 (topic #3 in LDA-lem_tfidf). 5 out of 10 users responded.

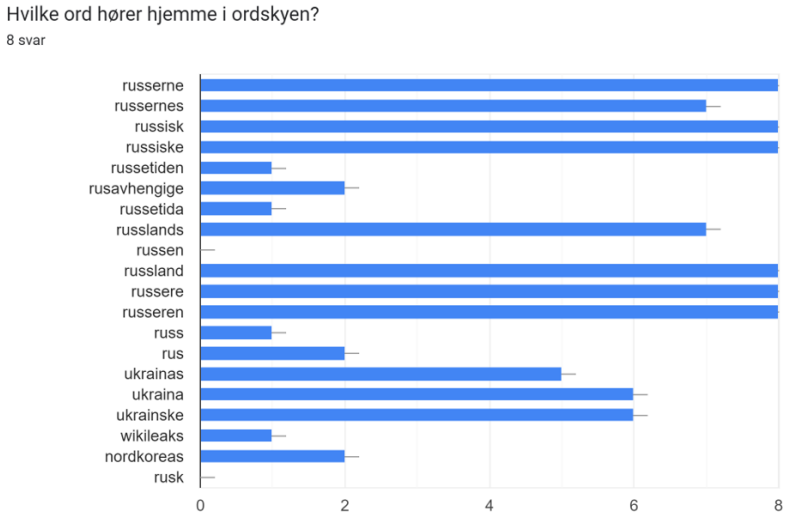


(b) This bar chart shows how many users selected each of the terms to be a part of the word cloud #4 (topic #0 in LDA-lem_tfidf). 5 out of 10 users responded.

Figure 4.16

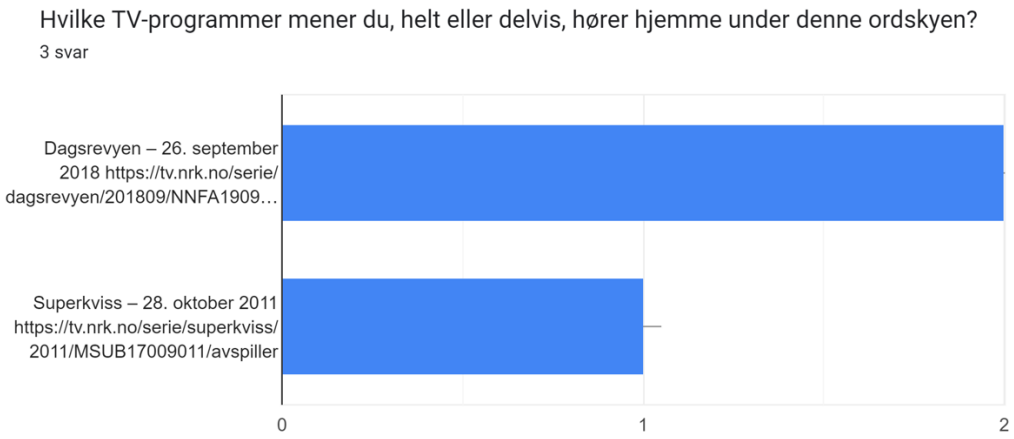


(a) This bar chart shows how many users selected each of the terms to be a part of the word cloud #1 (topic #5 in Top2Vec-basic). All 10 users responded.

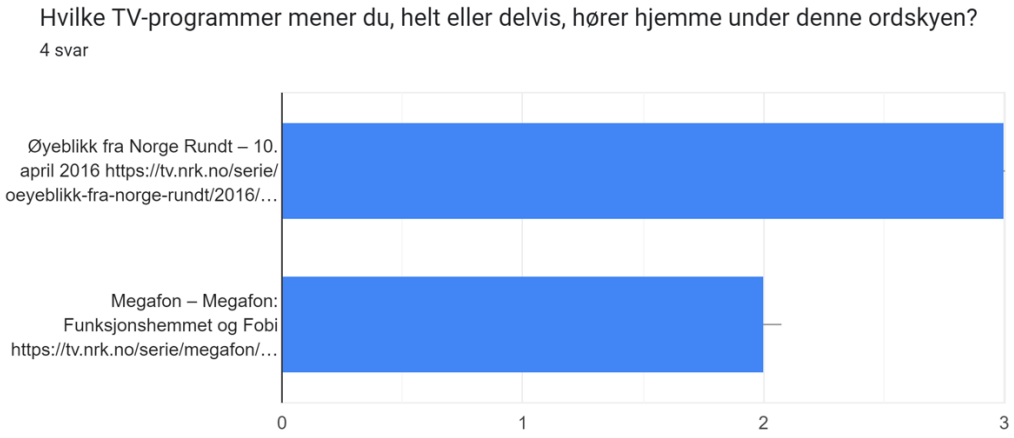


(b) This bar chart shows how many users selected each of the terms to be a part of the word cloud #3 (topic #4 in Top2Vec-basic). 8 out of 10 users responded.

Figure 4.17

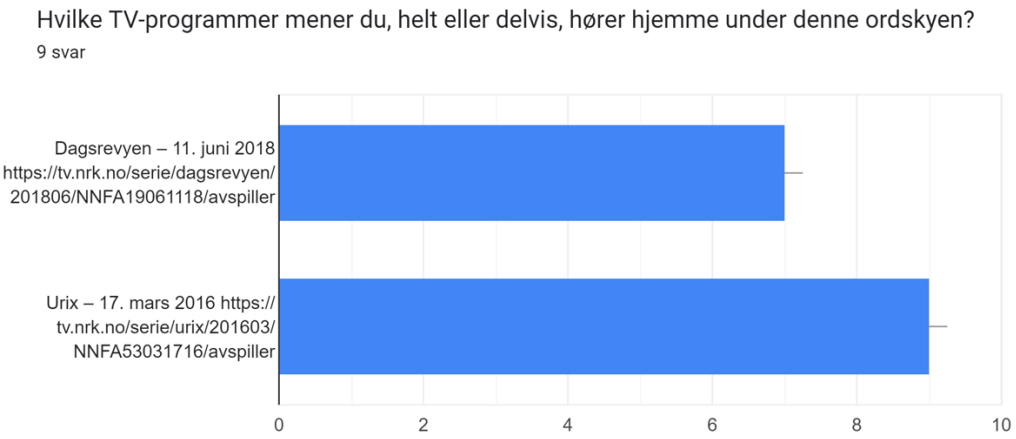


(a) This bar chart shows how many users found the word cloud #2 to be representative for each of the programs (topic #3 in LDA-lem_tfidf). Only 3 out of 10 users responded.

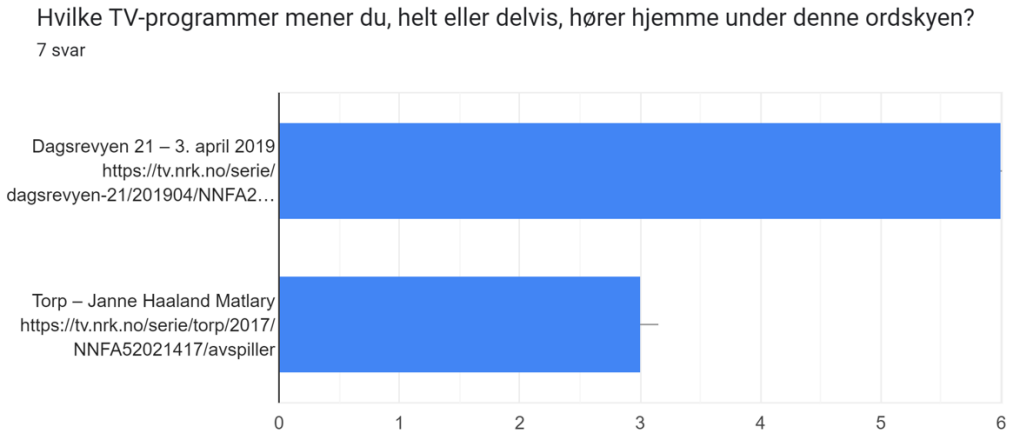


(b) This bar chart shows how many users found the word cloud #4 to be representative for each of the programs (topic #0 in LDA-lem_tfidf). Only 4 out of 10 users responded.

Figure 4.18

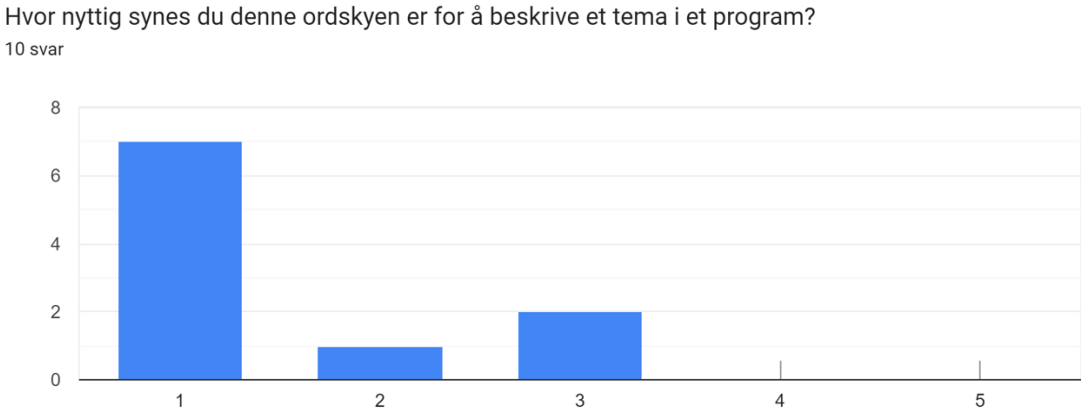


(a) This bar chart shows how many users found the word cloud #1 to be representative for each of the programs (topic #5 in Top2Vec-basic). 9 out of 10 users responded.

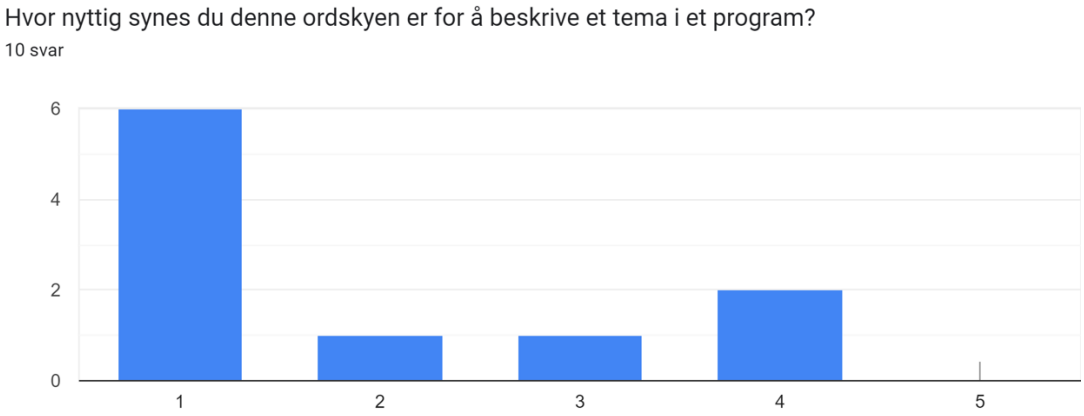


(b) This bar chart shows how many users found the word cloud #3 to be representative for each of the programs (topic #4 in Top2Vec-basic). 7 out of 10 users responded.

Figure 4.19

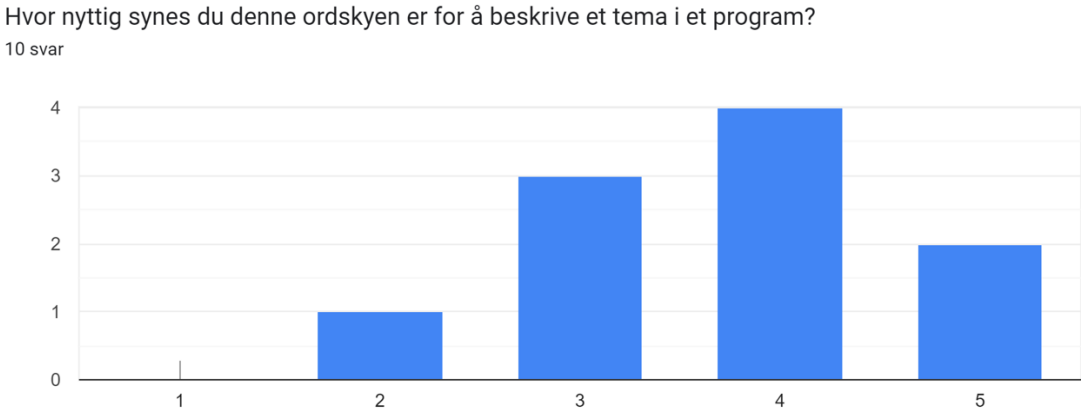


(a) This bar chart shows how many users found the word cloud #2 to be useful for describing the topic in a TV program (topic #3 in LDA-lem_tfidf). All 10 users responded.

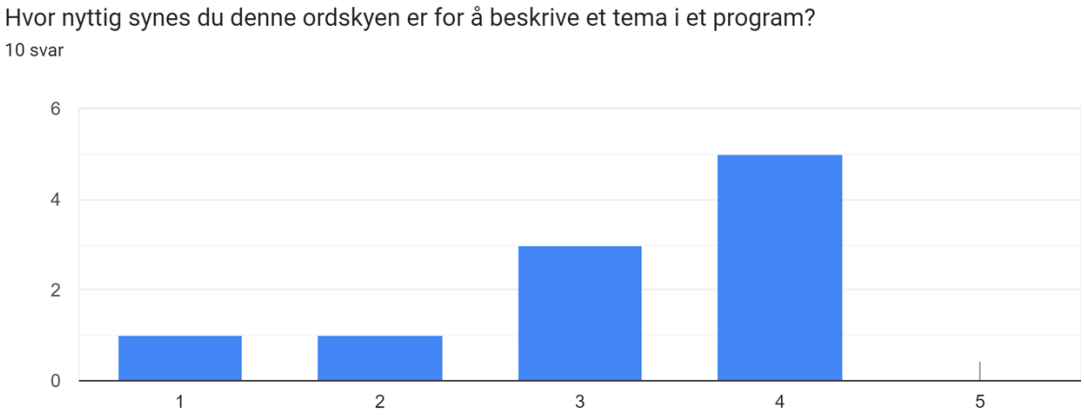


(b) This bar chart shows how many users found the word cloud #4 to be useful for describing the topic in a TV program (topic #0 in LDA-lem_tfidf). All 10 users responded.

Figure 4.20

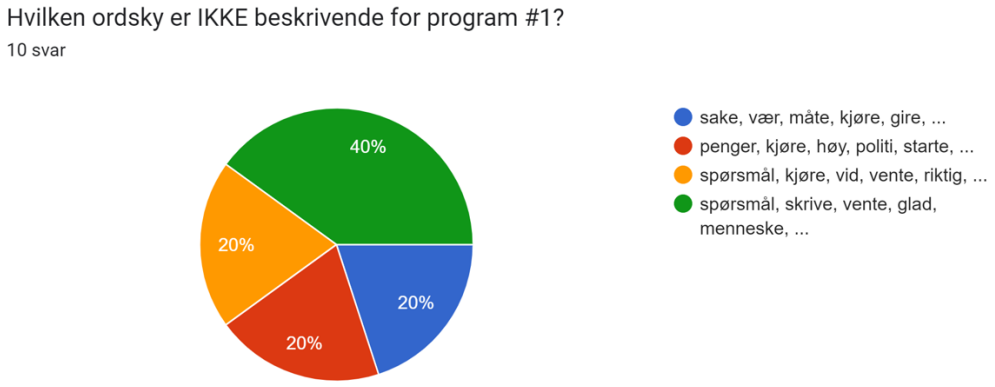


(a) This bar chart shows how many users found the word cloud #1 to be useful for describing the topic in a TV program (topic #5 in Top2Vec-basic). All 10 users responded.

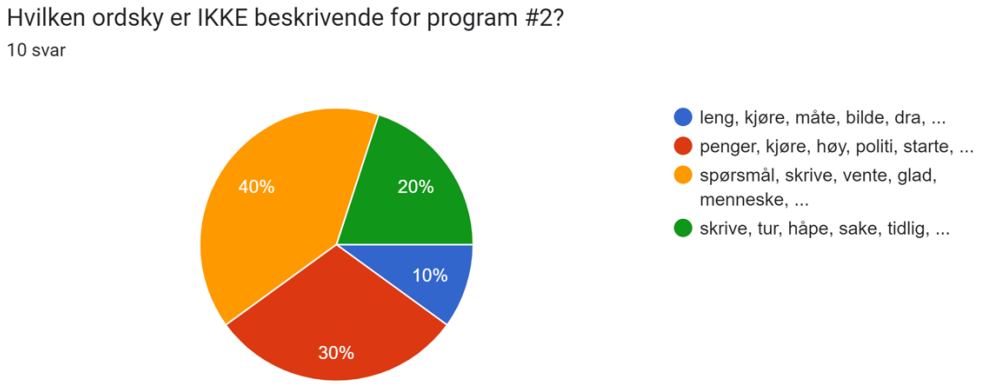


(b) This bar chart shows how many users found the word cloud #3 to be useful for describing the topic in a TV program (topic #4 in Top2Vec-basic). All 10 users responded.

Figure 4.21

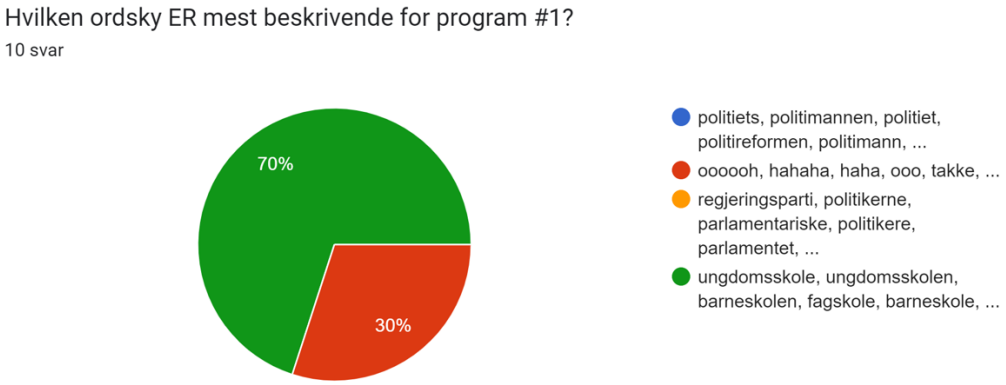


(a) This diagram shows how different users answered on what word cloud they thought were the intruder. The correct answer was topic #7 of LDA-lem_tfidf, which is the word cloud with the green sector: "spørsmål, skrive, vente, glad, menneske,...". All 10 users responded.

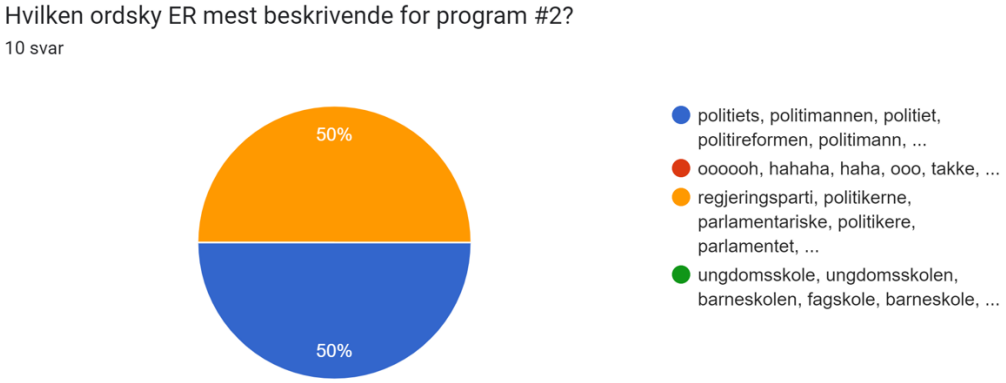


(b) This diagram shows how different users answered on what word cloud they thought were the intruder. The correct answer was topic #4 of LDA-lem_tfidf, which is the word cloud with the green sector: "skrive, tur, håpe, sake, tidlig,...". All 10 users responded.

Figure 4.22

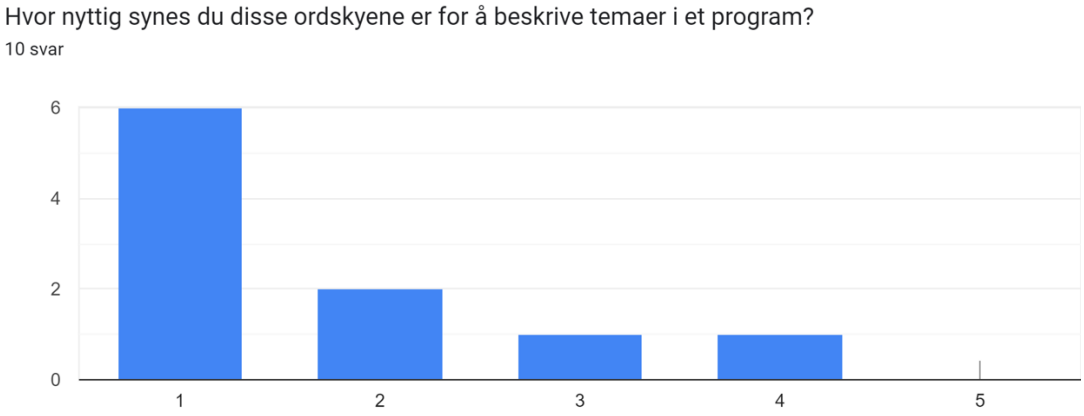


(a) This diagram shows how different users answered on what word cloud they thought was not the intruder. The correct answer was topic #3 of Top2Vec-basic, which is the word cloud with the green sector: "ungdomsskole, ungdomsskolen, barneskolen, fagskole, barneskole,...". All 10 users responded.

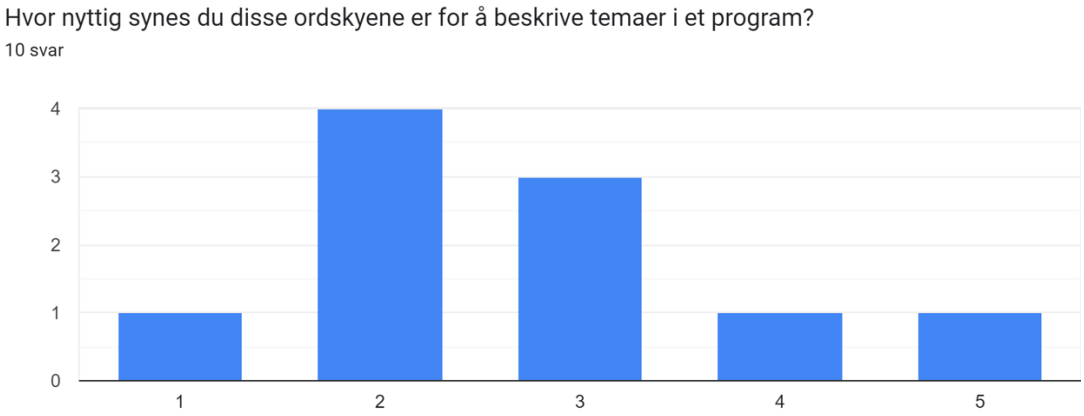


(b) This diagram shows how different users answered on what word cloud they thought were the intruder. The correct answer was topic #1 of Top2Vec-basic, which is the word cloud with the blue sector: "politiets, politimannen, politiet, politireformen, politimann,...". All 10 users responded.

Figure 4.23



(a) This bar chart shows how many users found the word clouds in the LDA word intrusion task to be useful for describing the topic in a TV program. All 10 users responded.



(b) This bar chart shows how many users found the word clouds in the Top2Vec word intrusion task to be useful for describing the topic in a TV program. All 10 users responded.

Figure 4.24

Model	**Survey ID**	**Topic ID**	**# answers**	**Name suggestions (# duplicates)**
LDA	2	3	6/10	Livsstil, Positivt politi, Frokost, Norge i dag, Norgesglasset, Støy
	4	0	6/10	Været, Uteliv, Der ingen skulle tru at nokon kunne bu, Håp utendørs, Støy, Barn i skolealder
Top2Vec	1	5	9/10	Midtøsten (2), Syria-krigen, Den tyrkisk-kurdiske konflikten, Syria, Kurdistan, Urix, Syria - Tyrkia, Syriske migranter til Tyrkia
	3	4	8/10	Konflikt, Kveldsnytt, Rusproblematikk i Russland, Russere, Russland (3), Russland og Ukraina

Table 4.7: The name suggestions for the LDA-lem_tfidf and Top2Vec-basic word clouds.

In addition to all the questions where users were asked to select options or name a topic, users were also asked to voluntarily leave feedback on each page of the survey on how well they thought the word cloud(s) worked as means to describe topics in a TV program. In addition, the last page of the survey asked users what they thought a useful word cloud looked like, and what would be important to take into consideration when creating such a word cloud. The replies to both questions can be seen in Table 4.8.

Model	Survey ID	Comment
LDA	2	"Really wondering what this really is."
		"Very difficult to imagine what a common name for this word cloud might be. Was mostly confused."
		"Confusing"
		"Impossible to describe - the word cloud makes no sense, has no intuitive common denominators"
		"This word cloud seems to not have managed to extract the meaningful parts in the text file"
	4	"Confusing and a bit bland"
		"I was onto something, but did not quite hit."
		"Impossible to name"
		"Not very meaningful words for retrieval in TV shows"
		"Do not understand this"
	WCI1	"Difficult to see the connection to the content"
		"The words "blive" and "sake" are repeated in several places. These seem to me like artifacts? Not words that are actually used that much."
		"I find this very fun!"
		"Do not see any of the matches. Unable to identify the intruder"
Top2Vec	1	"Two clear themes (Syria and Turkey) that go a little in different directions. It's the less important words that provide context about what the topic really could be."
		"Fancy!"
	3	"Thought this was a news broadcast from when Russia invaded Ukraine, but the Torp broadcast with an International Relations researcher is more in line, yes."
		"Impossible to name"
		"Mixes "rus", "russ" and "russers". Very useless"
	WCI2	"They only describe parts of the content of the program, not the program concept"
		"Composite programs with several elements. The word clouds were similar for the two programs, which do not make sense if three supposedly are constructed."
		"I think I nailed it"
		"The word cloud partly fits one of the elements of the program, but would have been more useful with one subject word per program element"

Table 4.8: The free text comments that users gave in the survey for the LDA-lem_tfidf and Top2Vec-basic word clouds. Note that the comments have been translated to English from Norwegian. "Survey ID" refers to the ID used for the individual word clouds used in the survey, while WCI = Word Cloud Intrusion, and refers to the free text field that asked about all the word clouds that were used in the word cloud intrusion tasks.

What is necessary to get a useful word cloud?
"Nouns and proper names"
"Thematic categorization."
"Variety! Not just adjectives etc, but all possible word classes (except articles, prepositions etc - they must be meaningful). About 15-20 words. Like the visual as it was now. But maybe some more parameters reflected by colors can be introduced? If it is uttered by a woman, it is red, if a man, it is green, etc. Or, I don't know, but some additional factor that makes it even more useful?"
"I think the concept of word clouds for programs is very good, but it may seem difficult to get a system to choose the right words to describe programs. Maybe nouns will work best. It is better with too many than too few words, but the balance before it becomes chaotic and only noise is subtle. I think some of these word clouds had a bit too many words, but like the design visually if the size of the word is related to how relevant it is to the content of the program."
"Be able to search for the "big" words. There are too many words in the cloud, feels very repetitive but not very informative. I think extracting names, places and one topic per program is more useful. Would also like to have the opportunity to search the entire corpus of text, not just what has been extracted."

Table 4.9: A longer question at the end of the survey asked what would be required to have a useful word cloud, and these are the 5 comments that were written. Note that the comments have been translated to English from Norwegian.

Chapter 5

Discussion

Here the results of all the three experiments and their implications will be discussed. First, the results of each experiment are addressed and discussed, before a discussion is done on whether or not the experiment and findings answers the research questions defined in Section 1.2. This will be done experiment for experiment in Section 5.1, Section 5.2 and Section 5.3. Finally, the limitations of the findings of this research is addressed in Section 5.4.

5.1 Preliminary NST Modeling Experiment Discussion

Here, the results and learnings found in the Preliminary NST Modeling Experiment of Section 4.3 will be discussed. The results of this experiment are found in Subsection 4.6.1. There were mixed results with regards to the performance of LDA and Top2Vec on the NST sample, but there were still promising aspects, and some alternative paths forward were highlighted.

5.1.1 Preliminary LDA Results

The 8 topics created by `LdaMulticore` are shown in Subsection 5.1.1. It is clear from these results, even before calculating any metrics or statistics for evaluating the model, that the topics are completely useless when LDA performs topic modeling directly on the NST sample. The word clouds are filled with semantically meaningless stop words such as "er", "i" and "og". Even the hyphen "-" character has managed to sneak in, which is probably due to hyphen often being used at the start of each new line of dialogue in subtitles. The diversity is also clearly horrible, as almost all the top topic words in each topic are top words in the other topics as well. When topic diversity and coherence is calculated this is also confirmed, with $C(T) = 0.434$ and $D(T) = 0.163$, where T is the set of the 8 topics found. Still, it is well-known that LDA needs some amount of preprocessing to give satisfactory results, which Churchill and Singh also verified more precisely. Thus, even though the results LDA return now are not at all usable, it is still included in the further experiments. This is both to use as a baseline for Top2Vec, but also to study effects of other preprocessing steps on different data than what was done in Churchill and Singh [2021b], in order to extend that research.

5.1.2 Preliminary Top2Vec Results

When it came to Top2Vec, there was some uncertainty from the beginning if it would work on Norwegian text. Not all the embedding models that Top2Vec support are trained for or handle

well Norwegian. However, due to the recommendation of Birkenes to use `distiluse-base-multilingual-cased` for training Norwegian language models, a model of which Top2Vec supports, there was more reason to believe there would be a suitable Top2Vec configuration that could work well on Norwegian text. All embedding models have been tested, as can be seen in Subsection 4.6.1.

The first one out was Doc2Vec. When Top2Vec is run on the NST sample with Doc2Vec as embedding-model, it returned four topics, which can be seen in Figure 4.12. The fact that this model worked out of the box makes sense, as it is language-agnostic and does not require pre-training for a language it has not seen before in order to start training on it. However, the topics were not very good. Although diverse, they did not seem to be coherent at all, with so many different words appearing together in the topic that rarely appears together, if at all. Next, `universal-sentence-encoder` and the multilingual variant, `universal-sentence-encoder-multilingual` was tried.

There was some error with the multilingual variant, so it was not possible to test it. Even though the normal encoder works, the result is even more nonsensical than that of `doc2vec`. Top2Vec only manages to find one topic in the whole NST sample when it uses `universal-sentence-encoder` as embedding model, which is certainly highly inaccurate. Besides, the topic words are also very bland with the exception of some words related to Denmark/Danish and Norway/Norwegian. However, the main words of the topic are stop words.

Finally, once topic modeling is attempted on NST using the `distiluse-base-multilingual-cased` embedding model, the results are drastically different. With this embedding model, Top2Vec finds 8 topics, the word clouds of which are seen in Figure 4.13. It is immediately clear that topics are relatively coherent here, which the reader should be able to testament to, with for topic #2 being about politics and topic #6 about Russia or Russians. They are also clearly distinct, with the only immediately visible overlap being some potential overlap in Norwegian/Nordic/coastal aspects between topic #2 and topic #5. Calculating metrics, however, gives an unexpected result: Although the model's diversity is very high, $D(T) = 0.957$, the coherence turns out to be very low, $C(T) = 0.088$. This means that this Top2Vec model has much lower coherence than the LDA model discussed in Subsection 5.1.1, even though the Top2Vec results for most humans should clearly be more coherent.

Although surprising, there is a clear hypothesis for how the Top2Vec model can get such a low coherence score. As can be seen in the equation for coherence, Equation 4.1, the measure is based on co-occurrence frequency for calculating the relative probability of the terms co-occurring in the same document. However, terms that have similar or related meanings might not show up in the same documents at all. For example, two TV-programs that are both about the COVID-19 pandemic might use distinct terms to describe the infection, with speakers in one program consequently calling the infection "Corona-infection", while speakers in the other program only use the term "COVID-infection". Many words might be synonyms or closely related in meaning, but are used by different speakers in different settings. Top2Vec attempts to utilize the semantic modeling of word embeddings to capture this, and seem to find many synonyms and related words among the topic words it find. However, the coherence metric ignores these relations and is completely dependent on the assumption that words that are related in meaning appear often together. Although this assumption has some truth to it, it is vulnerable to the cases of synonyms and related words being used exclusively of one another to describe the same thing. This further incentivizes the need for a more detailed study of effects of preprocessing, as well as comparing the results of the automatic metrics to human evaluations.

5.1.3 Preliminary Experiment and Answering of Research Questions

This preliminary experiment was intended to answer my first research question, **RQ1**: **What methods for topic modeling exist already and how do these methods work on Norwegian subtitle data?**

After this preliminary experiment, the two existing methods of LDA and Top2Vec are identified as viable for processing Norwegian subtitle data, through the successful use on the NST sample. Top2Vec already produces a result out of the box that is relatively interpretable and could perhaps even be used as is without the need for any extension. In any case, with the promising results that Top2Vec got from the NST sample when using `universal-sentence-encoder-multilingual` as embedding model, means that Top2Vec with this configuration is the most viable candidate model to move forward with in the next experiments.

Although LDA is not currently performing well on NST-like Norwegian data, this is expected to improve significantly once more preprocessing has been done on the dataset, as is seen with most applications of LDA models. As the model is expected to work well with enough preprocessing, LDA also satisfy as an answer to **RQ1**. However, it should be noted that many other types of topic modeling methods exist, including variations of LDA and other types of generative probabilistic methods. Other methods have not been closely examined, even though they might be more suitable for this task than LDA. At the same, LDA is well-known and successfully used in many contexts. LDA is also a good baseline and stepping stone for future topic modeling experiments and research with that can go further into more well-suited and tailored methods. On the other hand, Top2Vec is a quite new method that has not been used in much literature so far, which will allow for this research project to do something novel with respect to the application areas of recently developed models. So LDA and Top2Vec together provides a decent representation of the breadth of methodology in the topic model field, with one representing the tried-and-true and the other representing the novel methods. With this, yes, existing methods have been found that already works well or have potential to work well for Norwegian subtitle data, so **RQ1** can be considered answered as a result of this experiment.

5.2 NST Preprocessing Experiment Discussion

The preprocessing pipelines of the NST Preprocessing Experiment (Section 4.4) and results thereof (Subsection 4.6.2 show that preprocessing certainly affects the model quality of both LDA and Top2Vec, for better or worse.

5.2.1 Effects on Intrinsic Data Quality

First of all, it is clear from Table 4.5 that each pipeline removes considerable amounts of tokens from the NST sample, and thus should provide some interesting contrasts in the results of the models. There are specifically two aspects of the token statistics that should be noted. First of all, the basic pipeline is perhaps the most aggressive in terms of ratio of tokens removed. From the raw dataset to the basic dataset, the average number of tokens in files have fell to almost a third of the original size. Not surprisingly, stop words are completely gone, as this is the first pipeline to do stop word removal. This also seems to be the main reason behind the considerable drop in number of tokens, as this accounts for more than 80% of the fall in average token number. The other interesting aspect is how drastically the average token frequency increases with the steps of the lemmatized+TF/TF-IDF-cleaned pipeline (lem_tfidf). The increase also leads to this dataset having the highest average token frequency of all the datasets, which is why this number is highlighted. According to Churchill and Singh [2021b], a high average token frequency can

be a good indication that the dataset has less noise and infrequent tokens, which provide better conditions for generative probabilistic models in using less resources and finding better topic words. This big increase in average token frequency is coupled with a large drop in vocabulary size, the second largest of the table, which is likely due to the the TF-IDF-cleaning, where all tokens with a frequency under a certain threshold are removed. These two pipelines, the basic one and lem_tfidf one, that change the dataset in more dramatic ways than the others, will be discussed again in the next subsection when choice of models is done.

5.2.2 Effects on Extrinsic Data Quality

After evaluating the effect a diverse range of preprocessing steps has on the NST sample itself, the extrinsic data quality can be studied. The topics that were generated by the Top2Vec and LDA models from all the preprocessed datasets can be found in word cloud format in Section A. Topic coverage and topic diversity measures have also been computed for each dataset-model pair, and are found in Table 4.6 and Figure 4.14. From the table, it is clear what the effects are on the metrics of the topic models once their datasets are churned through stricter and stricter preprocessing filters. For LDA, coherence increases almost monotonically with heavier preprocessing (with the exception of the drop in score from basic to the lemmatized pipeline). Thus, the dataset that gives LDA the highest coherence is the pos_noun, that is the dataset that has been preprocessed the heaviest. Meanwhile, for Top2Vec, coherence increases up to the point of the lemmatized dataset, after which it starts degrading again. It even has the only negative coherence score in the dataset, which comes from the same dataset that gives LDA the highest coherence. Meanwhile, for topic diversity, there is no clear trend that comes with increasing preprocessing steps, for LDA at least. Here, it is the lem_tfidf dataset that scores the best diversity metric for LDA. Still, the greatest diversity achieved for LDA is $D(T) = 0.512$, which means that still more than half the top topic words found by LDA are not unique and found as a high-ranking word in one or more other topics. Lastly, it seems that the diversity of Top2Vec seem to be largely unaffected by the preprocessing steps, except for pos_noun, which significantly shifts its diversity down. It makes sense that Top2Vec consistently has very high diversity. The way topics are found through the use of clustering of words in Top2Vec, words of one topic are naturally quite distant from other topics, and it is rare and unlikely that words are in the middle of two or more clusters so they are chosen to be included as topic words in multiple topics.

It is clear from this that preprocessing has very different effect on different models, especially when they have two very different approaches to how they do modeling. As expected, LDA, a traditional model that often require quite a bit of preprocessing in other applications, generally gets the best metric results when significant preprocessing has been performed. However, too much preprocessing seem to start to negatively affect diversity, so one might have to perform a trade-off of less diversity once increased preprocessing to raise coherence reaches a certain level. Meanwhile, for Top2Vec, preprocessing must mainly be performed to increase topic coherence, and only up to a certain point.

The plot of coherence vs. diversity scores for all of the pipeline-model pairs in Figure 4.14 reveals additional patterns. Namely, LDA models (circles) seem to mostly cluster around an area of the coherence and diversity domain space that has high coherence, but lower diversity. Meanwhile, Top2Vec models (diamonds) are consistently on high or maximum diversity levels, but struggle to get as high coherence as any of the LDA models with increased preprocessing. The Top2Vec model that achieves the worst performance is very visible here, that being the model with the heaviest preprocessing pipeline, pos_verb. In total, the effects of preprocessing on the models are mostly as expected, with LDA often benefitting from heavier and heavier

preprocessing, and Top2Vec eventually falling apart with too much preprocessing. It is however interesting to see that Top2Vec does gain some performance increase with lighter preprocessing, as it was expected that Top2Vec would prefer complete and unaltered sentences for the language modeling/embedding part of it's process.

5.2.3 Selection of Best Models for Final Experiment

As explained in Section 4.5, the best LDA and Top2Vec models are chosen from all the models in the Preprocessing Experiment based on the harmonic mean of coherence and diversity, as defined by Equation 4.4. This is equivalent to picking the models closest to the upper right corner in the plot. The LDA model with the highest harmonic mean is the lem_tfidf with $H(M_{LDA,lem+tfidf}) = 0.733$. It is not surprising that this is the best LDA model, as the lem_tfidf dataset is the dataset with the highest average token frequency. As discussed in Subsection 5.2.1, a high average token frequency was a quality Churchill and Singh considered central to good performance for generative probabilistic models, which LDA is.

Meanwhile, the Top2Vec model with the highest harmonic mean is the lemmatized model with $H(M_{Top2Vec,lemmatized}) = 0.905$. However, the lemmatized Top2Vec model has only two topics (see Figure 11). As this would be too little data to perform any of the tests that were planned for the user survey, it was decided to discard this model and use the second best Top2Vec model, which is the basic variant. This model has harmonic mean of $H(M_{Top2Vec,basic}) = 0.891$, which is very close to that of the lemmatized model. However, the basic variant has 6 topics, which is a lot more usable in a user survey where multiple instances of topics need to be assessed to give a reasonable idea of how the model is evaluated as a whole. Again, that a model based on the basic pipeline would be one of the best is not unexpected. As this pipeline was the first to remove stop words, it is also the first pipeline to see the benefits of removing large numbers of unnecessary words, but at the same avoid potential drawbacks of two heavy preprocessing filters. Thus, the LDA-lem_tfidf and Top2Vec-basic models are the ones that will be used for the survey. The word cloud representations of the topics in each model is presented below. These are also the word clouds the respondents will see in the user survey.

5.2.4 NST Preprocessing Experiment and and Answering of Research Questions

The research question that the NST Preprocessing Experiment was intended to answer was the second one, **RQ2**: **Which preprocessing steps can be used on input data for each model found in RQ1, and what effect do these steps have on the intrinsic data quality as well as the results of each topic model?**

Having studied the preprocessing steps presented in Churchill and Singh [2021b], as well as having studied some of my own that are relevant for text preprocessing, as discussed in Section 3.1, the possible steps, or rules, that can be used for topic modeling preprocessing has been thoroughly outlined. By creating many separate preprocessing pipelines, the effect of most of the individual rules could be clearly seen. Performing experiments where rules who were not stacked alone (as the only extra rule) on a pipeline could be tested individually, and different combinations of pipelines where rules were dropped in addition to being stacked, should ideally also have been explored. Alternatively, the rules and the stacking of them could have been organized as an optimization/search problem for finding the best stack of rules efficiently. However, even though there should have been more systematic studies of the different possibilites for stacking rules, the six pipelines gave a good idea of the effect most of the rules had on the data. The effect on both the intrinsic and extrinsic quality has however been well documented. So this

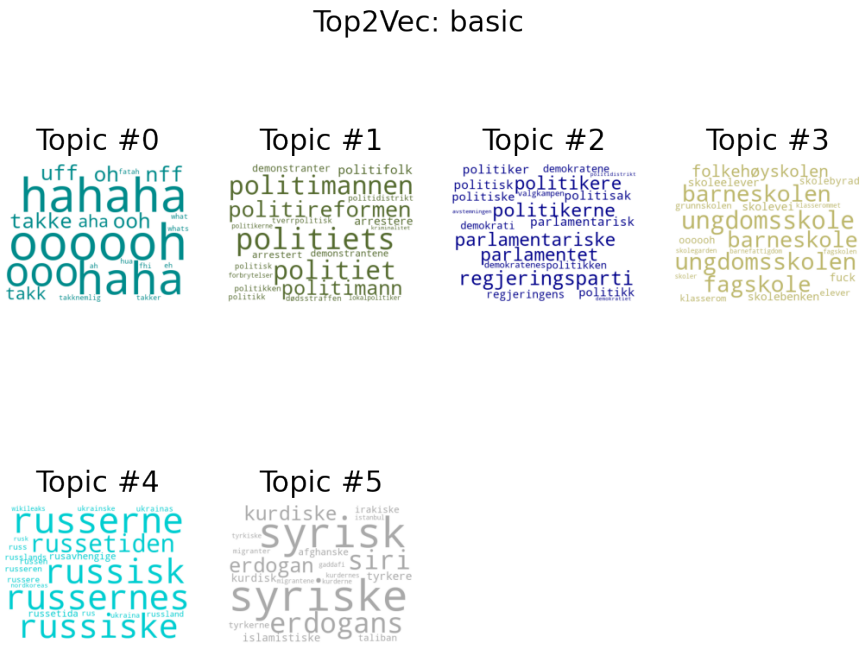


Figure 5.1: Top2Vec model generated from the basic dataset.

Figure 5.2: LDA model generated from the lemmatized + TF + TF-IDF-cleaned dataset.

research question has been answered, but there are potentially other preprocessing pipelines that would have improved results even more.

5.3 NST Topic Interpretation Experiment Discussion

With the responses presented in Subsection 4.6.3, there are multiple aspects that can be discussed, but there is especially one observation that is important: Top2Vec topics outperformed LDA topics in all of the questions, tasks and tests. The concrete differences are presented in the list below:

- **Higher response rate:** All questions in the individual word cloud tasks (part 1 of the survey) were voluntary to answer (with the exception of the question asking the user to grade the usefulness of the word cloud). The user was encouraged to leave questions blank if they felt like they did not understand enough about the topic in the word cloud to properly answer the questions. The questions relating to LDA word clouds were consistently responded to less than those of Top2Vec word clouds. All Top2Vec questions were responded to by at least 7 out of 10 respondents, and often more. Meanwhile, no voluntary LDA question was responded to by more than 6 out of 10 respondents at most, and often less.

- **More consistency in naming:** The naming task had more consistency in what names were given to Top2Vec word clouds than to LDA ones (see Table 4.7. In both Top2Vec questions there were many similar names that are semantically close to each other, and there were even duplicate names (two or more users writing the exact same name) in both questions. In the LDA questions, however, the names are much more disparate.

- **Higher percentage of correct answers:** In the word cloud intrusion tests, a higher percentage of people correctly found the right answer (see Figure 4.22 and Figure 4.23). There were also less spread in the responses, as only one of two of the options were selected by users, even though four options were possible. It must of course be noted that the Top2Vec word cloud intrusion test was not identical to the one used for LDA word clouds. However, this should not greatly affect the difficulty of the tests.

- **Higher self-reported usefulness:** In both the Top2Vec word clouds, at least 50% of users reported that they found the word clouds to be useful (rating of 4 or 5), while more than 50% of users reported that they did not find the LDA word clouds to be useful at all (rating of 1). These self-reports are found in Figure 4.21 and Figure 4.20, respectively. Even though less users find the word clouds for Top2Vec useful in the word cloud intrusion test (only 20% find these word clouds useful, while 30% are indifferent or unsure), this distribution is still better than the responses for the word cloud intrusion test for LDA. This can be seen in Figure 4.24.

- **Higher self-reported confusion:** In the free text fields of the individual word cloud pages, users can leave comments about the specific word cloud they answered questions about. In the fields for the individual word clouds, there were multiple users reporting being confused on the LDA word cloud pages, but not nearly as many on the Top2Vec comments. This can be seen in Table 4.9.

Even though this result is understandable based on a quick glance on the concrete topics that are found in each of the topic sets of LDA-lem_tfidf and Top2Vec-basic, it is still surprising to

see that the LDA topics could have such a high topic coherence score compared to the Top2Vec topics. Yet, based on the user survey, coherence as judged by humans is much higher in the Top2Vec topics. This adds to the hypothesis that the topic coherence measure has some limits that are extra apparent as soon as it is used to measure coherence on an word embedding-based topic model, due to the fundamental differences in how the topics are constructed.

Another interesting, and related hypothesis, is that diversity could be a better heuristic for human interpretability than coherence. Top2Vec-basic has 100% diversity, meaning no top-c words occur among other topics' top words. Meanwhile LDA-lem_tfidf has 52% diversity, meaning on average, almost half the top-c words in each topic occur among the top words of the other topics. This is visually evident in the word clouds of LDA-lem_tfidf topics (see Figure 5.2). For example the word "kjøre" appears in almost all the topics. Could it be that the topics become bland by having low enough diversity? In any case, how coherence and diversity correlates or anti-correlates with human interpretability should be further explored, in a similar manner as Chang et al. did.

It must also be noted that the survey only had 10 participants, and as such, it's hard to draw any statistically significant conclusion from the data material. However, it does give an indication that there might be underlying patters here that should be further explored.

5.3.1 Topic Interpretation Experiment and Answering of Research Questions

The last and final experiment, this Topic Interpretation Experiment, was conducted to answer the last research question, **RQ3**: **Based on RQ1 and RQ2, what combination of topic modeling method and preprocessing steps give the best results according to journalists and editorial staff in NRK?**

LDA-lem_tfidf and Top2Vec-Basic are found to be the models with the best preprocessing steps from each group of models, based on the automatic metrics coherence and diversity. It is also clear based on the user survey that Top2Vec-basic is not only the best model according to NRK staff, but also significantly more usable than LDA-lem_tfidf. The user survey was done with 60% being journalists or editor in NRK (see Figure 4.15), and some of them did seem to find value in the topics and their connection to TV-programs, especially Top2Vec. If a system was to be developed and used immediately, based on the survey conducted with mostly NRK employees who work as journalists or editors, Top2Vec would without a doubt be the best candidate for such a system, along with a preprocessing pipeline similar to the basic pipeline. So the final research question of the study has been answered. Looking at the research goal as well, the goal is partly realized. The first part of the goal has been achieved, as performing topic modeling on Norwegian subtitle data has been proven feasible. The last part, creating topics that the user can easily understand and relate to programs, probably require further work.

5.4 Limitations

Here, some limiting factors for the work is presented, as well as some aspects that could be threats to validity. First of all, there were limited computational resources and time to build use large datasets, build large models and train deep. I had a laptop device with more power than a usual student laptop thanks to NRK, but I still had to limit resources. If there had been more resources available, all experiments might have been possible to do with the full NST dataset, which might have given room for more nuanced themes, due to more terms being present. However, this seemed very infeasible, as the calculation of co-frequency of terms in a

dataset seemed to be very taxing on resources. I even had to rewrite the functionality to use parallel processing (see project code), in order for the co_frequencies to actually be calculated in a feasible time frame. In addition, it could have turned out that the `doc2vec` embedding model might have given better results if allowed to train with maximal training time. However, Angelov said that this takes a lot of time to train, so it seemed more feasible to continue simply with the `distiluse-base-multilingual-cased` embedding model, as it was possible to use the pre-trained model for this.

It would also have been tempting to explore using an estimator function for setting the k-parameter, as mentioned before, and how this alone could affect the results of the LDA model. This would have made sense to explore in the Section 4.3). In addition for the preprocessing pipelines, it would have been interesting to see if all rules could be freely stacked in pipelines, and then tried every combination of rules to see what combination would be optimal, perhaps using a search algorithm to avoid having to exhaust all possible combinations.

When it comes to potential threats to validity, there might be some. First of all, with regards to internal validity, I realized very recently (after having performed much of experiments and thesis work) that the output of `Top2Vec` is not fully deterministic. That means that it might find 8 topics the first time it is run, 6 topics the next time it is run on the same dataset, and so on. The topic words might also have changed from run to run. Now whether these results also change the coherence and diversity metric for the topics between two runs on the same dataset is unclear. If so, the actual ordering and thus result for the Top2Vec model might not be valid because he results are not the same when one tries to reproduce them.the models are created (for example, this time when run, the Top2Vec-pos_verb model might get the best harmonic mean of coherence and diversity, and thus get the spot in the survey. And when it comes to the survey itself, the number of respondent might be too low (20 people)to say that we know with certainty that Top2Vec is the model with the best interpretability.

Chapter 6

Conclusion and Future Work

In this final chapter, the work of this thesis is concluded in Section 6.1, while proposals for future work is suggested in Section 6.2.

6.1 Conclusion

In this thesis, we have seen how there is a deficit in NLP resources for Norwegian today, and how this is important moving forward to enable technology and services for the whole population. We have also seen how NRK has an increased need for structured metadata, with increasing digitization and challenges such as editorial insight over content, accurate and efficient reporting of activities as well as improving user access to content. Through the MEGAS project, this thesis was to find a way of generating structured metadata that could be useful to NRK. I then narrowed the focus of this thesis to topic modeling.

With the formulated research goal and research questions of Section 1.2, we have seen how these questions have been answered through the research plan of three experiments: to assess the possibilities of using topic modeling on subtitles in the first place, to see what effect preprocessing of the subtitle can have on performance of the topic models, and finally to involve relevant end users in NRK by performing a user survey

First Chapter 2 introduced the existing literature that was important to establish and explore for this thesis. The field of topic modeling was introduced, the topic models of LDA and Top2Vec were discussed, and why these two methods were chosen to be used for this project was explained this. Then, appropriate text preprocessing methods and evaluation metrics for topic models were introduced. This laid the foundation for discussing the related work most central to this study: the preprocessing experiments and framework of Churchill and Singh [2021b], as well as the user interpretation focus and specific human evaluation metrics of Chang et al. [2009].

With the foundation laid, I introduced the methodology, as well as implementation of said methodology. Here, our overall pipeline, from raw data to topics, were shown. The NST dataset that was created specifically for the MEGAS project by NRK was explored here, along with how custom preprocessing rules would be made based on the standard ones in textPrep. The topic models and their parameters were discussed further, before a section delved into the overall format of the user survey with NRK was introduced.

Then, the setup and details of all the experiments could be discussed. The NST dataset was discussed in detail here. This was when the plan and setup for the three experiments of this thesis were explored. First, the preliminary experiment was discussed, which is where the NST sample of size $N = 1000$ is created. A few simple experiments are performed with both

LDA and Top2Vec models to find the optimal configurations. Then, secondly, the preprocessing experiment is performed using the ideal configurations of both LDA and Top2Vec, and a set of pipelines that are built with an extended textPrep library. Once these experiments are presented, the final experiment of human interpretation is performed, using the user survey presented there. The results of each experiment are then introduced in the same chapter.

Finally, the discussion go into what the results from each experiment mean, and what consequences they have for the experiment after one another, but also for future experiments.

6.2 Future Work

There are many aspects that would be interesting to look at in further work in this area. First of all, it would be interesting to consider comparing LDA and Top2Vec with other topic models, especially that are outside the domain of generative probabilistic and distributed representation models, and Churchill and Singh [2021a] mentions many other model types that might be worth wile to explore. Besides, there are many variations of generative probabilistic models other than LDA that might get massive improvements in performance over LDA on the NST dataset and for use in broadcasting.

Another area to explore is whether topic coherence using NPMI is a good measure for how coherent a topic set is, especially with regards to models that stray from the classic generative probabilistic approach. As indicated in Section 5.3, there were considerable deviations from interpreted coherence by survey respondents, compared to the topic coherence calculated by NPMI. A future study should examine thoroughly, much in the same way as Chang et al. [2009], how different non-traditional topic models such as Top2Vec perform with regards to popular classic automatic metrics for computing topic coherence, diversity, and properties of topic, topic sets and topic-document mappings. This study should also include human evaluation as Chang et al. did, and whether or not those human-based evaluations correlate or anti-correlate with traditional metrics, and for which kinds of topic models.

Bibliography

Angelov, D. (2020). Top2vec: Distributed representations of topics.

Birkenes, M. B. (2022). private communication.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Boyd-Graber, J., Hu, Y., and Mimno, D. (2017). *Applications of topic models*, volume 11. Now Publishers Inc.

Chang, J., Gerrish, S., Wang, C., Boyd-graber, J., and Blei, D. (2009). Reading tea leaves: How humans interpret topic models. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

Churchill, R. and Singh, L. (2021a). The evolution of topic modeling. *ACM Comput. Surv.* Just Accepted.

Churchill, R. and Singh, L. (2021b). 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*.

Hodgson, R., Cristea, A., Shi, L., and Graham, J. (2021). Wide-scale automatic analysis of 20 years of its research. In Cristea, A. I. and Troussas, C., editors, *Intelligent Tutoring Systems*, pages 8–21, Cham. Springer International Publishing.

Ma, P., Zeng-Treitler, Q., and Nelson, S. J. (2021). Use of two topic modeling methods to investigate covid vaccine hesitancy. In *14th International Conference on ICT, Society, and Human Beings, ICT 2021, 18th International Conference on Web Based Communities and Social Media, WBC 2021 and 13th International Conference on e-Health, EH 2021-Held at the 15th Multi-Conference on Computer Science and Information Systems, MCCSIS 2021*, pages 221–226.

Vayansky, I. and Kumar, S. A. (2020). A review of topic modeling methods. *Information Systems*, 94.

Appendices

A Topic Models from NST Preprocessing Experiment

A.1 LDA Models

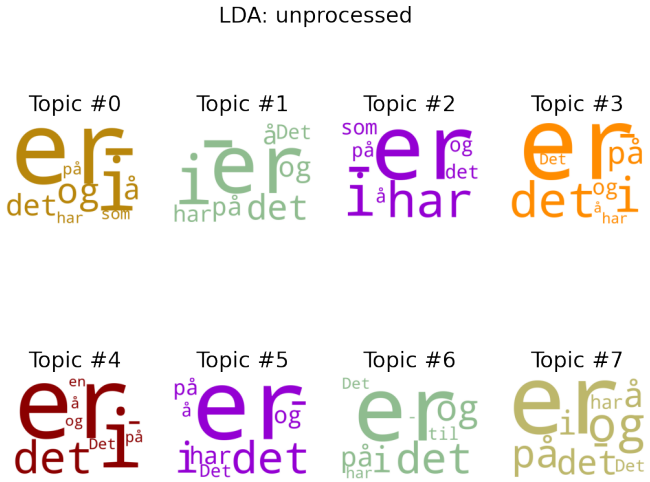


Figure 1: LDA model generated from the unprocessed dataset (NST sample).

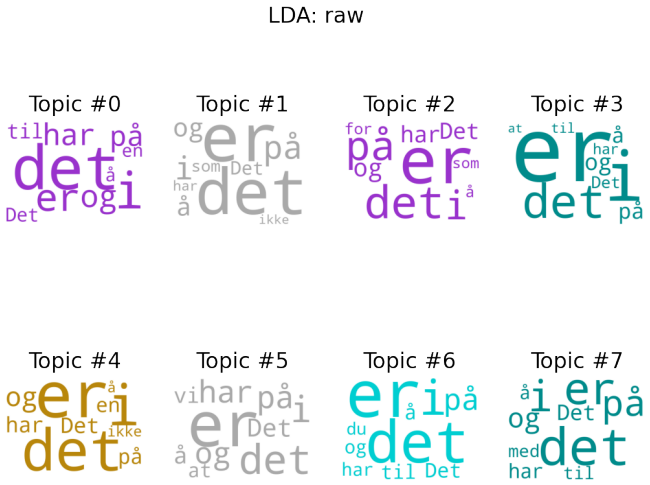


Figure 2: LDA model generated from the raw dataset.

Figure 3: LDA model generated from the basic dataset.

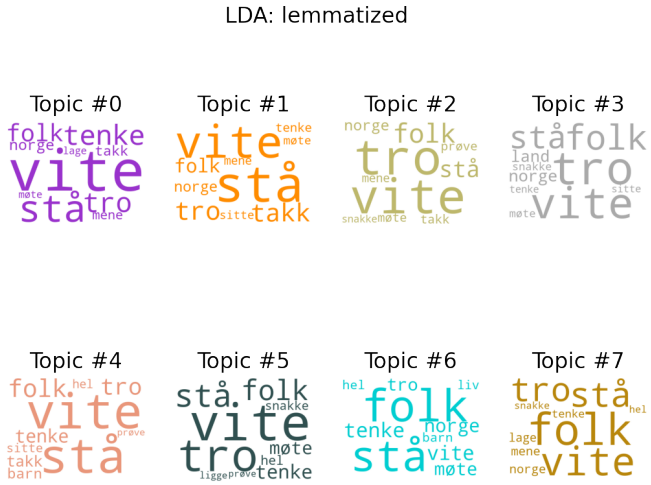


Figure 4: LDA model generated from the lemmatized dataset.

Figure 5: LDA model generated from the lemmatized + TF + TF-IDF-cleaned dataset.

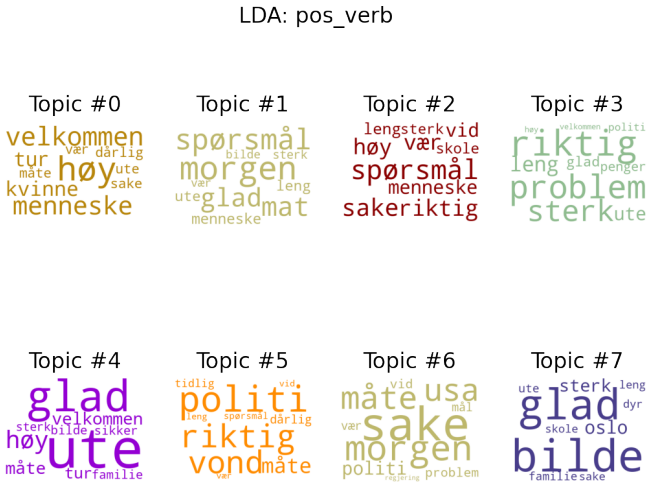


Figure 6: LDA model generated from the PoS-cleaned (verbs removed) dataset.

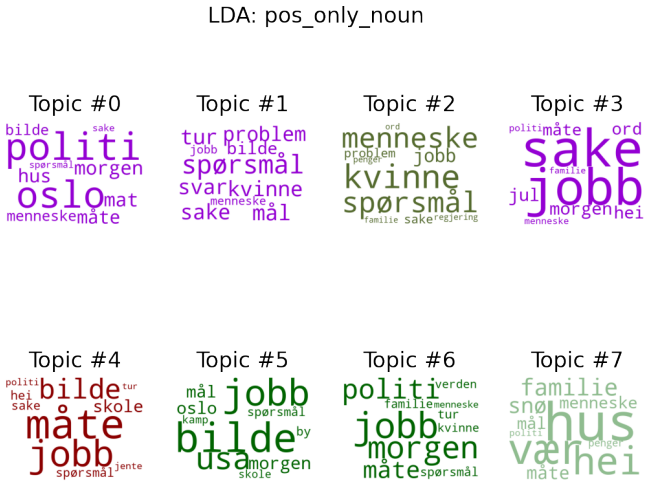


Figure 7: LDA model generated from the PoS-cleaned (only nouns kept) dataset.

A.2 Top2Vec Models

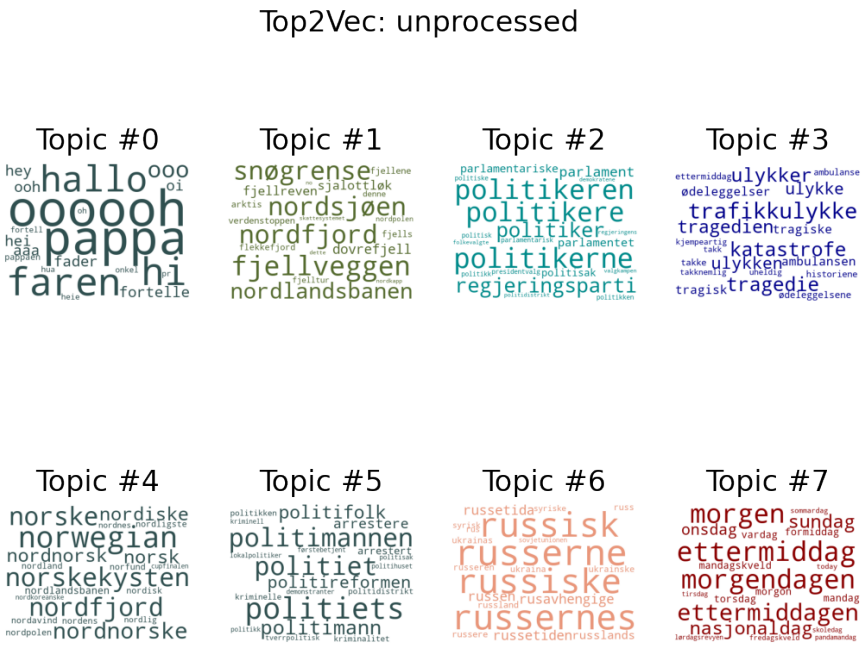


Figure 8: Top2Vec model generated from the unprocessed dataset (NST sample).

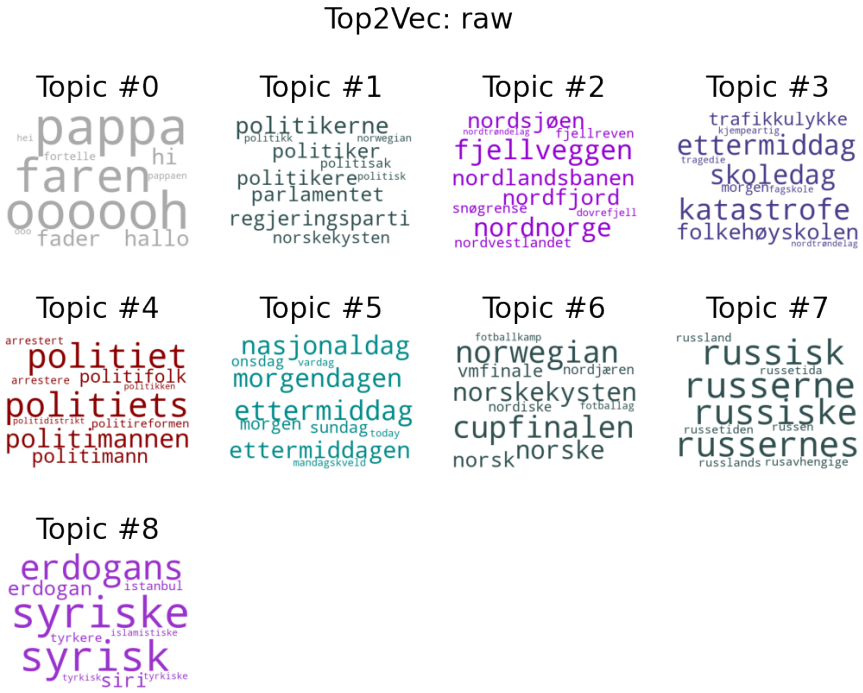


Figure 9: Top2Vec model generated from the raw dataset.

Figure 10: Top2Vec model generated from the basic dataset.

Figure 11: Top2Vec model generated from the lemmatized dataset.

Figure 12: Top2Vec model generated from the lemmatized + TF + TF-IDF-cleaned dataset.

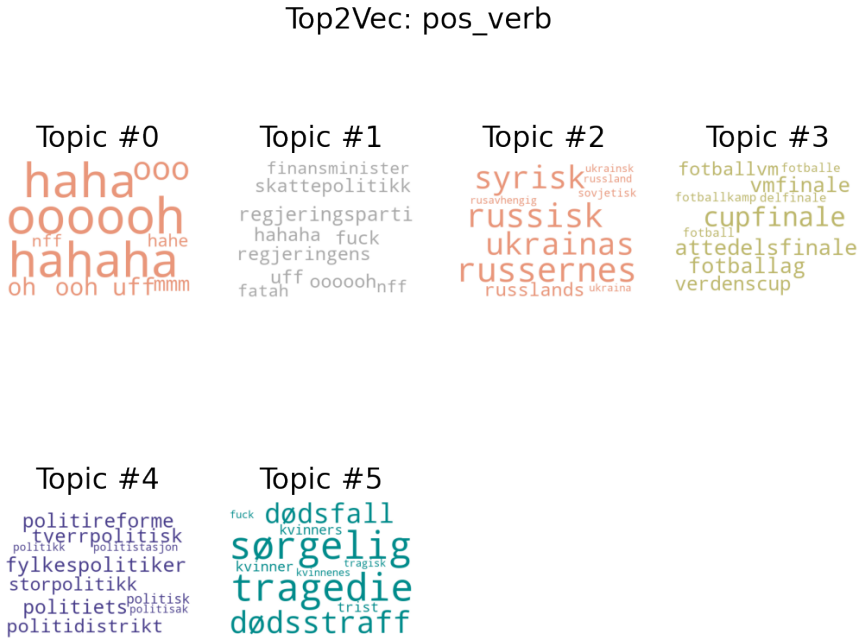


Figure 13: Top2Vec model generated from the PoS-cleaned (verbs removed) dataset.

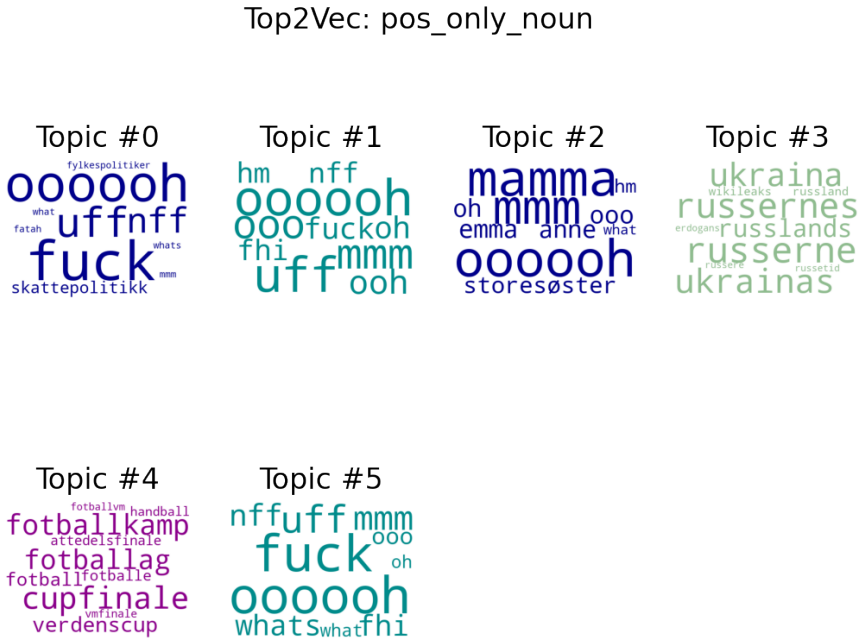


Figure 14: Top2Vec model generated from the PoS-cleaned (only nouns kept) dataset.

B NRK User Survey

Spørreundersøkelse om automatisk genererte ordskyer fra TV-programmer

Mitt navn er Magnus og jeg skriver nå en masteroppgave i datateknologi ved NTNU i samarbeid med NRK. Oppgaven handler om hvordan man kan tilføre NRKs TV-programmer god og strukturert metadata ved hjelp av maskinlæring.

I den anledning ønsker jeg å svare på om et system jeg har laget klarer å hente ut meningsfulle og presise "temabeskrivelser" om programmene. Disse "temabeskrivelsene" har form som ordskyer med emneord som er hentet fra TV-programmenes undertekst, der ordskyene forsøker å finne et eller flere temaer i programmet. Se bildet under for eksempel. NB: Disse ordskyene kan skille seg noe fra eksisterende definisjoner og bruk av "tema", "saksunivers", "kategorier", osv. internt i NRK.

Tanken er at disse ordskyene i framtiden skal kunne brukes av journalister og andre som jobber redaksjonelt i NRK som et hjelpeverktøy i publisering, slik at de kan skaffe seg bedre oversikt over tema som går igjen på tvers av innhold og for å lettere finne TV-programmer, radioprogrammer og/eller nyhetsartikler som henger sammen med hverandre tematisk. Men for å kartlegge om disse ordskyene er verdifulle, så trenger jeg tilbakemelding fra relevante brukere på hvor presise og meningsfulle ordskyene er som temabeskrivelser for innholdet de omfatter.

I denne undersøkelsen har systemet mitt generert 14 ordskyer ut fra en samling på undertekster fra 1000 TV-programmer i NRK TV. Programmene dekker et vidt spekter av tema og sjangre, og inneholder alt fra nyhetsprogrammer til underholdningsprogrammer. Ordskyene som systemet mitt lager har ikke navn, men består alle av et antall emneord som systemet har plukket ut for å forsøke å beskrive helheten av temaet. Et emneord i ordskyen har større skriftstørrelse jo mer representativt systemet tror det er for temaet. Med fire av ordskyene følger det noen spørsmål som ønskes besvart, og til slutt er det en seksjon hvor de resterende ordskyene og TV-programmer skal matches. Det er fint om du svarer på alle spørsmålene etter beste evne, men du kan også la spørsmål stå ubesvart om du ikke vet hva du skal svare.

*Må fylles ut

Eksempel på ordsky:

1. Før vi begynner: Hva slags rolle/stilling har du i NRK? Denne informasjonen vil ikke brukes for å identfisere deg, men kun for å forbedre informasjonen i den samlede statistikken *

Markér bare én oval.

- ◯ Journalist
- ◯ Redaktør/leder
- ◯ Administrasjon
- ◯ Utvikler
- ◯ Andre: _____

Ordsky #1

Ordskyens navn

Systemet mitt kan ikke automatisk gi navn til ordskyene den finner. Det kan kun bruke ulike ord den finner i underteksten på tilhørende TV-programmer (emneordene som er vist i ordskyen) som et forsøk på å beskrive temaet. Hvilket navn tenker du er dekkende for ordskyen?

Ordet/-ene du bruker kan komme fra ordskyen over, men trenger ikke å være det. Dersom du ikke klarer å finne noe passende navn for ordskyen, la feltet stå blankt.

2. Hvilket navn vil du gi ordskyen?

Ord som hører hjemme i ordskyen

Dersom det er tydelig for deg hvilket tema det er som ordskyen prøver å beskrive, og det var mulig for deg å sette et navn på ordskyen, hvilke emneord mener du passer inn i ordskyen? Kryss av for alle emneord du synes hører hjemme under ordskyen.

Hvis det derimot er uklart for deg hva slags tema det er ordskyen prøver å beskrive, og du ikke fant et navn til ordskyen, ikke sett noen kryss her.

3. Hvilke ord hører hjemme i ordskyen?

Merk av for alt som passer

- ☐ syriske
- ☐ syrisk
- ☐ erdogans
- ☐ siri
- ☐ erdogan
- ☐ kurdiske
- ☐ tyrkere
- ☐ islamistiske
- ☐ kurdisk
- ☐ afghanske
- ☐ irakiske
- ☐ taliban
- ☐ tyrkerne
- ☐ migranter
- ☐ migrantene
- ☐ kurdernes
- ☐ istanbul
- ☐ tyrkiske
- ☐ kurderne
- ☐ gaddafi

TV-programmer tilknyttet ordskyen

Her er navn og link til noen av TV-programmene som systemet mitt mener hører inn under temaet beskrevet av ordskyen. Mener du også at programmene passer inn under ordskyen? Ta en rask kikk på innholdet i programmene, og kryss av på de du mener passer inn. Dersom det er uklart for deg hva slags tema det er ordskyen prøver å beskrive, og du ikke fant et navn til ordskyen, ikke sett noen kryss her.

NB: Noen programmer, slik som nyhetssendinger, kan omhandle vidt forskjellige temaer på ulike tidspunkt i løpet av programmet. Kryss likevel av på TV-programmet om gjeldende ordsky er representativ for minst ett av temaene du mener programmet omhandler.

4. Hvilke TV-programmer mener du, helt eller delvis, hører hjemme under denne ordskyen?

Merk av for alt som passer

☐ Dagsrevyen – 11. juni 2018
https://tv.nrk.no/serie/dagsrevyen/201806/NNFA19061118/avspiller

☐ Urix – 17. mars 2016
https://tv.nrk.no/serie/urix/201603/NNFA53031716/avspiller

5. Hvor nyttig synes du denne ordskyen er for å beskrive et tema i et program? *

Markér bare én oval.

	1	2	3	4	5	
Svært lite nyttig	◯	◯	◯	◯	◯	Svært nyttig

6. Har du noen kommentarer om ordskyen?

Ordsky #2

7. Hvilket navn vil du gi ordskyen?

8. Hvilke ord hører hjemme i ordskyen?

Merk av for alt som passer

- ☐ politi
- ☐ håpe
- ☐ riktig
- ☐ velkommen
- ☐ glad
- ☐ skrive
- ☐ gire
- ☐ hete
- ☐ spise
- ☐ mat
- ☐ bilde
- ☐ starte
- ☐ glede
- ☐ vid
- ☐ sake
- ☐ morgen
- ☐ føle
- ☐ bygge
- ☐ vær
- ☐ tur

9. Hvilke TV-programmer mener du, helt eller delvis, hører hjemme under denne ordskyen?

Merk av for alt som passer

☐ Dagsrevyen – 26. september 2018
https://tv.nrk.no/serie/dagsrevyen/20180 9/NNFA19092618/avspiller

☐ Superkviss – 28. oktober 2011
https://tv.nrk.no/serie/superkviss/2011/M SUB17009011/avspiller

10. Hvor nyttig synes du denne ordskyen er for å beskrive et tema i et program? *

Markér bare én oval.

	1	2	3	4	5	
Svært lite nyttig	⬭	⬭	⬭	⬭	⬭	Svært nyttig

11. Har du noen kommentarer om ordskyen?

Ordsky #3

12. Hvilket navn vil du gi ordskyen?

13. Hvilke ord hører hjemme i ordskyen?

Merk av for alt som passer

- [ ] russerne
- [ ] russernes
- [ ] russisk
- [ ] russiske
- [ ] russetiden
- [ ] rusavhengige
- [ ] russetida
- [ ] russlands
- [ ] russen
- [ ] russland
- [ ] russere
- [ ] russeren
- [ ] russ
- [ ] rus
- [ ] ukrainas
- [ ] ukraina
- [ ] ukrainske
- [ ] wikileaks
- [ ] nordkoreas
- [ ] rusk

14. Hvilke TV-programmer mener du, helt eller delvis, hører hjemme under denne ordskyen?

Merk av for alt som passer

☐ Dagsrevyen 21 – 3. april 2019
https://tv.nrk.no/serie/dagsrevyen-21/201904/NNFA21040319/avspiller

☐ Torp – Janne Haaland Matlary
https://tv.nrk.no/serie/torp/2017/NNFA52021417/avspiller

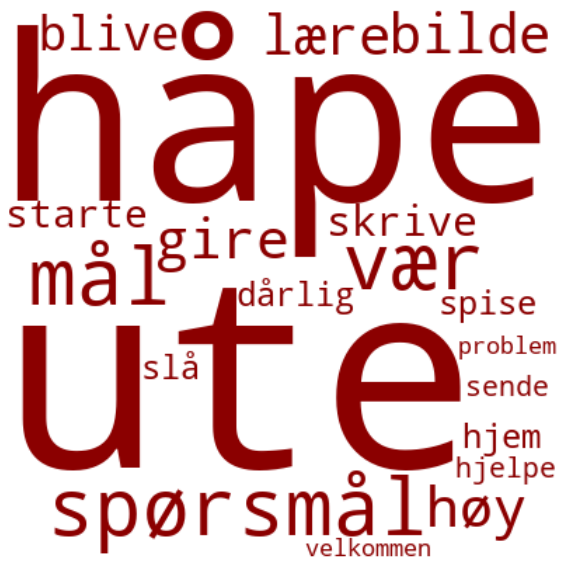
15. Hvor nyttig synes du denne ordskyen er for å beskrive et tema i et program? *

Markér bare én oval.

	1	2	3	4	5	
Svært lite nyttig	⬭	⬭	⬭	⬭	⬭	Svært nyttig

16. Har du noen kommentarer om ordskyen?

Ordsky #4



17. Hvilket navn vil du gi ordskyen?

18. Hvilke ord hører hjemme i ordskyen?

Merk av for alt som passer

- [ ] ute
- [ ] håpe
- [ ] spørsmål
- [ ] mål
- [ ] vær
- [ ] gire
- [ ] høy
- [ ] lære
- [ ] bilde
- [ ] blive
- [ ] skrive
- [ ] starte
- [ ] dårlig
- [ ] slå
- [ ] hjem
- [ ] spise
- [ ] hjelpe
- [ ] sende
- [ ] problem
- [ ] velkommen

19. Hvilke TV-programmer mener du, helt eller delvis, hører hjemme under denne ordskyen?

Merk av for alt som passer

☐ Øyeblikk fra Norge Rundt – 10. april 2016 https://tv.nrk.no/serie/oeyeblikk-fra-norge-rundt/2016/DVNR08000616/avspiller

☐ Megafon – Megafon: Funksjonshemmet og Fobi https://tv.nrk.no/serie/megafon/2012/MSUB07002312/avspiller

20. Hvor nyttig synes du denne ordskyen er for å beskrive et tema i et program? *

Markér bare én oval.

	1	2	3	4	5	
Svært lite nyttig	◯	◯	◯	◯	◯	Svært nyttig

21. Har du noen kommentarer om ordskyen?

Hvem skal ut: Tre ekte ordskyer, én "inntrengerordsky"

I hvert spørsmål under er det et TV-program og fire ordskyer som systemet mitt fant. Tre av ordskyene er i virkeligheten tilknyttet TV-programmet, mens én av dem er en "inntrengerordsky" jeg har lagt til selv: denne ordskyen har egentlig lite å gjøre med temaene i programmet, ifølge systemet mitt. Hvilken ordsky mener du er den som har minst å gjøre med TV-programmet?

Program #1
Link til program: https://tv.nrk.no/serie/tema-psykisk-helse/sesong/5/episode/3/avspiller

Tema - Psykisk helse – 3. Pust (Sesong 5)

22. Hvilken ordsky er IKKE beskrivende for program #1? *

Markér bare én oval.

sake, vær, måte, kjøre, gire, ...

penger, kjøre, høy, politi, starte, ...

spørsmål, kjøre, vid, vente, riktig, ...

spørsmål, skrive, vente, glad, menneske, ...

Program #2

Link til program: https://tv.nrk.no/serie/broedrene-dal-og-professor-droevels-hemmelighet/sesong/1/episode/12/avspiller

Brødrene Dal og professor Drøvels hemmelighet – 12. episode (Sesong 1)

23. Hvilken ordsky er IKKE beskrivende for program #2? *

Markér bare én oval.

leng, kjøre, måte, bilde, dra, ...

penger, kjøre, høy, politi, starte, ...

spørsmål, skrive, vente, glad, menneske, ...

skrive, tur, håpe, sake, tidlig, ...

24. Hvor nyttig synes du disse ordskyene er for å beskrive temaer i et program? *

Markér bare én oval.

	1	2	3	4	5	
Svært lite nyttig	◯	◯	◯	◯	◯	Svært nyttig

25. Har du noen kommentarer om ordskyene?

Hvem skal ut: Én ekte ordsky, tre "inntrengerordskyer"	I hvert spørsmål under er det et TV-program og fire ordskyer som systemet mitt fant. Her er kun ÉN av ordskyene i virkeligheten tilknyttet TV-programmet, mens de resterende tre er "inntrengerordskyer" jeg har lagt til. Hvilken ordsky mener du er den som har mest å gjøre med TV-programmet?

Program #1
Link til program: https://tv.nrk.no/serie/supernytt/201903/MSUB02005319/avspiller

Supernytt – 18. mars 2019

26. Hvilken ordsky ER mest beskrivende for program #1? *

Markér bare én oval.

politiets, politimannen, politiet, politireformen, politimann, ...

oooooh, hahaha, haha, ooo, takke, ...

regjeringsparti, politikerne, parlamentariske, politikere, parlamentet, ...

ungdomsskole, ungdomsskolen, barneskolen, fagskole, barneskole, ...

Program #2

Link til program: https://tv.nrk.no/serie/nyheter/201901/NNFA17011119/avspiller

NRK Nyheter – 11. jan. 2019 kl. 17:00

27. Hvilken ordsky ER mest beskrivende for program #2? *

Markér bare én oval.

politiets, politimannen, politiet, politireformen, politimann, ...

oooooh, hahaha, haha, ooo, takke, ...

regjeringsparti, politikerne, parlamentariske, politikere, parlamentet, ...

ungdomsskole, ungdomsskolen, barneskolen, fagskole, barneskole, ...

28. Hvor nyttig synes du disse ordskyene er for å beskrive temaer i et program? *

Markér bare én oval.

	1	2	3	4	5	
Svært lite nyttig	◯	◯	◯	◯	◯	Svært nyttig

29. Har du noen kommentarer om ordskyene?

Eventuelt

Spørsmålene her er frivillige, men svar gjerne på dem om du vil bidra med ekstra kommentarer

Nyttige ordskyer

Etter å ha svart på spørsmålene i denne undersøkelsen, hva tror du er viktig for å få en "nyttig" ordsky? Med nyttig menes det at ordskyen gir god og tydelig informasjon om et eller flere temaer i et program, og at ordskyen gjør det lettere å identifisere hvilke programmer som er relevante i en gitt kontekst, samt å finne andre relevante programmer. Eksempler på spørsmål som kan være aktuelt å reflektere rundt, er:

- Hvilke ordklasser brukes til å danne emneord i ordskyene (substantiv, verb, adjektiv, osv.)?
- Skal emneordene dekke over et bredt spekter av ord som kan brukes til å beskrive temaet, eller bør de dekke et mye smalere spekter av ord som spisser temaet mest mulig?
- Omtrent hvor mange ord er passende å vise i en slik ordsky?
- Visuell utforming på ordskyen (font, størrelse på ord, fargebruk, retning, osv.)?

30. Hva mener du er viktig for å få en mest mulig "nyttig" ordsky?

31. Har du noen andre kommentarer til undersøkelsen eller ordskykonseptet?