Stance Detection in Danish Politics

Thesis Project, Course Code: KISPECI1SE
IT University of Copenhagen

by Rasmus Lehmann (rale@itu.dk)
supervised by Leon Derczynski

June 3, 2019

Contents

Abstract

A growing amount of information is available online regarding the state of politics, in the form of interviews, debates, party programs and articles, making the task of staying able to follow and participate informedly in politics increasingly difficult. This thesis seeks to propose a solution to this challenge by translation of textual data into quantitative data through the creation of a dataset of quotes from Danish politicians, and the use of Natural Language Processing (NLP) through stance detection. A primary goal of this thesis has been to generate a dataset and NLP classification models, which can be applied for further research, both within and outside the field of Danish NLP.

The thesis contains three main parts; the creation of an annotated dataset for stance detection using quotes from politicians, the design of three deep learning-based models which can classify stance for this dataset, and finally an analysis of the performance of said models, as well as the distribution of the data in the generated dataset.

The dataset was created through manual data collection, taking into consideration the objectivity of included media outlets. Immigration was chosen as the topic of the dataset, based on an analysis performed in 2018 of which topics were found most important by the Danish population. It was found that quotes within the dataset were naturally split into two sub-topics; one regarding national immigration policy, and one regarding the centralization of immigration policy. Annotation guidelines for each of the sub-topics were created, the dataset was cleaned and quotes were labelled as either *for*, *against* or *neutral* towards the sub-topic. The dataset was observed to be somewhat skewed, both in regards to label distribution, gender distribution, distribution along party lines and distribution along the political axis.

Three recurrent deep learning-based models were designed, built, optimized and tested. The simplest of the three models, outperformed the other two, as well as two benchmark models. The highest performance achieved by the strongest model was a $F1_{macro}$ score of 0.575 and a $F1_{micro}$ score of 0.733. Including the quoted politician and the political party of the quoted politician as features in the models was shown to have a large positive effect on performance. It was found that the two models taking an averaged quote embedding as input far outperformed the model taking a single word at each time step. Error analysis showed that an increased dataset size would have improved performance, and that the use of a more advanced stochastic gradient descent-based optimizer than the one applied in this paper, might also have improved both performance and runtime.

Statistical analysis of the generated dataset showed that label distributions within the right-wing and left-wing parties respectively were relatively similar, except for a few outlier parties.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

As a function of digitalization, the availability of information regarding the state of politics has never been greater, interviews, debates, party programs and articles all readily available online. This can be seen as a great democratic benefit, contributing to the enlightenment of the populous, giving individuals a large basis on which to form their opinions, and place their votes. However, the large amount of available information leads to the time consumption required for keeping up to date on the state of politics becoming increasingly higher, possibly to an extend where the populous is not able to follow. Combining this wide availability of information with an increasing tendency towards politicians changing their statements from day to day, of which the current President of the United States Donald Trump would be an obvious example, the pressure on the populous grows, to continually keep up to date, if they wish to be able to follow and participate informedly in the political debate.

Reading large amounts of text is a time-consuming task. Thus, a partial solution to this challenge, is the translation of textual data into quantitative data, representing a large amount of text in a more compact fashion. Natural Language Processing (NLP) is the theoretical field concerned with the automatic parsing, analysis and understanding of text. Within this field we find the task of stance detection, concerned with discerning the stance in a text towards some target. Within this paper, quantitative analysis will be applied as a tool to make information regarding the state of politics more easily available, and the applicability of stance detection for creating quantitative data for political analysis will be explored.

1.2 Research problem

For this thesis, the objective has been two-pronged; creating a dataset of quotes from politicians labelled for stance, allowing statistical analysis of opinions within parties and for each politician, and building a machine learning-based stance detection model, able to determine the stances within quotes in the generated dataset.

1.2.1 Dataset creation

The task of collecting data for the dataset is defined as the extraction of quotes from news articles for all political parties within the Danish parliament. Here, considerations are made regarding the objectivity of the collected data, both taking into account the subjectivity of journalists, media outlets and the researcher.

The task of data labelling will be performed using the labels *for*, *against* and *neutral*, discussed in depth in Chapter 3. For this task, the subjectivity of the researcher is the primary concern, in regards to the objectivity and general applicability of the dataset.

Finally, the task of statistical analysis consists of presenting, comparing and reflecting on the opinions of politicians and political parties, based on the collected quotes and the stance assigned to them through labelling, as well as analysing the representativity of the dataset.

1.2.2 Stance detection task

The task of stance detection is defined as the automatic detection of a stance within a given quote towards some target, using the stance classes *for* and *against* the target, or as neither *for* nor *against* the target, which for this thesis will be called *neutral*. The goal of this project is to create a model which can perform this task, both to be used as a tool for political analysis and to expand the generated dataset by automatic labelling of quotes, as well as to be used as a benchmark for further research within the field of NLP in Danish.

1.3 Approach

Data collection, parsing, cleaning and labelling is performed, based on reflections regarding the choice of sources, politicians and topics to be included in the dataset, after which the resulting dataset is used for statistical analysis and discussion of the stances within the political parties in the Danish parliament.

Three different stance detection models will be designed, and a comparative analysis for the three models is set up, to evaluate their performance on the generated dataset. A number of further experiments are performed, to identify the effect of a number of parameters on the performance of the designed models.

1.3.1 Deep Learning for stance detection

A number of different approaches have been applied to the task of stance detection, including probabilistic classifiers [Qazvinian et al., 2011], kernel-based classifiers [Mohammad et al., 2017, Enayet and El-Beltagy, 2017, Collins and Duffy, 2001] and ensemble learners [Zeng et al., 2016, Tutek et al., 2016]. However, in recent years neural network-based approaches, also referred to as deep learning approaches, have been showing great promise in solving this task. The SemEval 2016 Task 6 competition is an example of this, a stance detection task centered around identifying stances in tweets. Here, the two top performing teams both applied deep learning models [Zarrella and Marsh, 2016, Wei et al., 2016]. Due to this, the primary stance detection approach applied within this paper will

be deep learning based. An ensemble-based classifier and a probabilistic classifier will be implemented as baseline models, to evaluate whether they can be out-performed by a deep learning based approach for this task, as was the case for the SemEval 2016 Task 6.

1.4 Organization of thesis

The thesis is organized as follows; Chapter 2 lays the foundation of the paper, and gives an introduction to the application of artificial intelligence within the field of political analysis, and a brief primer on the field of NLP. In Chapter 3, the process of generating the dataset of quotes from politicians is described, including the desired properties and scope of the dataset and the applied data gathering, parsing, cleaning and labelling practices. Chapter 4 contains a primer on core ML and deep learning principles and terminology, followed by a presentation and discussion of the design of three deep learning based stance detection models, their network architecture and their optimization. Chapter 5 presents the optimizational and experimental setup used for testing the performance of the models, followed by the results of said experiments and optimization. Experiments include a comparative analysis of the three deep learning models and two benchmark models, and a number of smaller experiments to clarify the effects of a number of parameters on model performance. In Chapter 6, the results of the experiments presented in Chapter 5 are analysed, followed by an error analysis for the three models. Furthermore, statistical analyses of the generated dataset are performed, to evaluate the distribution of quote labels within each party, and the quote distribution between the parties. Chapter 7 concludes the thesis, and proposes some future avenues of research within the topic as well as applications of the work performed within this thesis.

Chapter 2

Background

Within this chapter, the theoretical background of the thesis project is presented, on which further chapters will be based. Firstly, examples of data driven artificial intelligence-based approaches to political analysis are presented in Section 2.1, followed by an overview of the field of NLP in Section 2.2, focusing on the task of stance detection and NLP for Danish.

The field of artificial intelligence (AI) has roots within, among others, the fields of philosophy, mathematics, economics, neuroscience, psychology, computer engineering and linguistics, which in turn has led to AI becoming an umbrella term covering a vast number of sub-fields. Research within the field strives for different goals, optimized to focus on mimicking human behaviour, achieving optimal behaviour, mimicking human thought processes, achieving optimal thought processes, or a combination of the above [Russell and Norvig, 2016]. The work within this thesis both seeks to mimic human thought processes in the creation of neural networks, imitating the neurological structure of the human mind and mimic human behaviour in creating a model which can correctly interpret patterns in text to identify opinions of humans. Nevertheless, the ultimate goal of this thesis is not to create a system which mimics humans, but rather using knowledge from the field of AI to build a tool for dataset creation and data analysis, meanwhile furthering the field of NLP in Danish to allow for better research in and creation of such tools in the future.

2.1 Data driven approaches to politics

Within the field of AI, a large number of tools exist, both simple search and constraint satisfaction algorithms as well as tools for complex probabilistic modeling, language processing and decision making [Russell and Norvig, 2016]. The following section contains a brief exploration of how some of the tools within AI might be applied for political analysis and within politics in general.

The growing availability of data combined with growing availability of processing power in later years, has greatly increased the potential of AI-based approaches to data analysis and decision making [Russell and Norvig, 2016]. Recently, speculations regarding the application of AI within the political sphere have made headlines. With claims of AI being a major (if not deciding) factor in the USA elections in 2016 [Polonski, 2017], AI building up to leading whole cities and playing a major role in future British politics [Christou, 2018] and, if action is not taken promptly, will come to make decisions regarding military

action and international policy [Miyake, 2019]. Despite these claims seeming far-fetched, the current applications, and even more so the potential, of AI within the fields of politics and political science are overwhelming.

As described further in Section 2.2.2, AI already shows strong results in rumour identification and rumour veracity determination. If an effective system able to solve this task with high confidence is built, it can be applied to live fact-checking politicians during political campaigns, possibly irrevocably changing the way political campaigns and debates are run. Likewise described in Section 2.2.2, AI has also shown great promise within determining the political stance of individuals based on text, both that of politicians and that of individuals towards politicians. By applying tools such as this, combined with automated scraping devices on social media such as Facebook or Twitter, a quantifiable dataset can be built over a population's stance towards a long list of political topics. A populist politician could base his or her entire political strategy on statistical analysis on such a dataset, gaining votes simply by most efficiently giving the most people what they want.

The use-cases are many, and in quick succession one might mention the classification of individuals within a populous to identify voter groups using clustering, tree-classifiers or neural networks, automatic generation of speeches using NLP, and optimization of campaign planning using search and planning algorithms. It is safe to say, that the application of AI within politics can make the jobs of politicians much more straight forward. It therefore seems prudent to develop uses of AI to set higher standards for politicians, and develop AI-based tools for more in-depth political analysis. Developing a tool for setting higher standards for politicians is one of the goals of this thesis.

2.2 Natural Language Processing

The field of NLP, like the field of AI, functions as an umbrella term covering a large number of sub-fields, all concerned with the automated processing of human language, including fields working with parsing and producing written text as well as fields working with spoken language [Goldberg, 2017]. Within this thesis, focus is on the sub-field of NLP concerning parsing and understanding written language, and specifically the task of stance detection. This section presents some of the technologies used within this field.

2.2.1 Feature engineering within NLP

A core feature of data used in NLP is the representation of words in a way that computers can understand. In this, two primary word representations are considered; one-hot feature vectors and dense embeddings. When features are translated into one-hot vectors, the vector generated is the length of the total vocabulary, and a single flag is raised at the index of the word in question. This retains no information regarding the word, other than the fact that it is present. For a dense word embedding, values within the vector will take a scalar value. The strength of this representation is, that similar words are given similar scalar values, either by training these scalar values or by using pre-trained embeddings [Grave et al., 2018, Pennington et al., 2014], allowing the embeddings to store semantics about the word rather than just its presence [Goldberg, 2017]. For a 10-word vocabulary,

a one-hot vector for the word at index 6 and a dense representation of dimensionality 5 can be seen in Equation 2.1.

$$\begin{aligned} OneHot &= [0, 0, 0, 0, 0, 1, 0, 0, 0, 0] \\ Emb &= [1, 0.3, -0.2, 0.6, -0.7] \end{aligned} \tag{2.1}$$

Aside from word representations, features used in NLP can be divided into two subcategories; text-based features and context-based features.

Text-based features

Text-based features can be generated using only the given text, and range from simple values that can be easily parsed directly from the text to more complex features, that might require use of a separate NLP system of their own.

Simple textual features such as use of punctuation and swear words or positive words found in a dictionary, or the use of URLs, images or hashtags, can be represented as a term frequency (TF). The most basic TF is a boolean value to represent the presence of a given swear word or punctuation. This can be adjusted for factors such as text length, can be scaled logarithmically, or the TF score can be based on TF of the most frequent terms in a given text, to not give the score a bias towards longer texts. Another simple textual feature is the length of a given text, either as a character or word count. A number of advanced features have been successfully used for the task of stance detection, including the use of lexicons or a sentiment analysis model to determine the sentiment of a given text [Li et al., 2017, Zeng et al., 2016, Enayet and El-Beltagy, 2017] and mappings of words to their syntactic functions in the form of part of speech tags using an NLP model [Mohammad et al., 2017, Zeng et al., 2016, Qazvinian et al., 2011].

Context-based features

Context-based features consist of knowledge regarding a data point, which can not be extracted directly from its text. These include, but are far from limited to, the sender and information regarding the sender, e.g. his home country, receiver and information regarding the receiver and text type, for instance whether it is a reply to or a forward of an earlier text and similarity to other texts. Strong results within stance detection and other NLP classification tasks have been achieved by supplementing text-based features with context-based features [Augenstein et al., 2016, Qazvinian et al., 2011, Enayet and El-Beltagy, 2017, Kochina et al., 2017], and therefore context-based features will be applied within this thesis as well.

2.2.2 Stance detection

The task of stance detection consists of automatically classifying the stance of an author of a text towards some target. Stance detection is often compared to the classification task of sentiment analysis, however sentiment analysis differs in the fact that no target is used within the classification, but instead the general sentiment of the text is identified.

The task of stance detection has been applied widely within political analysis, both analysing the stance of politicians towards some topic [Lai et al., 2016, Skeppstedt et al., 2017], and of individuals towards some politician or policy [Augenstein et al., 2016, Mohammad et al., 2016, Johnson and Goldwasser, 2016, Iyyer et al., 2014], and is a cornerstone in the analysis of rumour identification using NLP [Zeng et al., 2016, Qazvinian et al., 2011, Ma et al., 2018, Kochina et al., 2017]. For stance detection used in rumour identification, the labelling scheme SDQC has been widely applied, S denoting *supporting* some statement or rumour, D denoting *denying* it, Q denoting *querying*, requesting additional information regarding the statement or rumour, and C denoting *commenting*, supplying additional information. Within this paper, the labels *for*, *against* and *neutral* will be applied, following precedence within the research field of political stance detection of using a three-class labelling scheme, one class denoting positive, one negative and one neutral or unidentifiable stance towards some target [Mohammad et al., 2016, Skeppstedt et al., 2017, Johnson and Goldwasser, 2016].

Stance detection using bi-directionality, conditionality and gating

The model implementations within this paper are influenced by the winning entrance at the SemEval 2017 Task 7 competition by the Turing research group [Kochina et al., 2017]. Task 7 of SemEval 2017 differs from the task of this paper in a number of ways. Task 7 is concerned with rumour identification, and the dataset is thus labelled using the SDQC scheme. Furthermore, the dataset is created using data from Twitter, and includes structural information about the tweets in a tree-like format, where all tweets either are, or are connected to, a source tweet, from which tweets form conversation branches through their references to each other. This is not the case for the dataset used within this paper, and thus the branch-based approach applied by [Kochina et al., 2017] is not applicable. The model implemented by [Kochina et al., 2017] is a neural network, making use of recurrence, gating, bi-directionality and conditionality. All of these elements are described in-depth in Sections 4.2.1 and 4.3, and implemented in the models designed for this paper.

2.2.3 NLP for Danish

Seeing as the dataset for this thesis is to be generated using quotes in Danish, this Section considers the current state of NLP in Danish. Compared to English, the amount of resources available to researchers within NLP in Danish are limited. Fewer annotated datasets exist, resulting in a large over-head in the form of dataset creation and labelling being connected to performing research within the field. Furthermore, the field is lacking publicly available tools for, amongst others, sentiment analysis and stance detection.

However, tools such as lexical resources [Wikidata, 2019, KU, 2019a], tokenization and stemming tools [NLTK, 2019, Samoor, 2019] and tools for part of speech-tagging and named entity recognition [Samoor, 2019] are readily available. Furthermore, the field is growing, and specialized research units exist in most major universities, including Københavns Universitet [KU, 2019b], Syddansk Universitet [SDU, 2019] and IT-Universitetet København [ITU, 2019], suggesting that resources might become more plentiful in the future. To contribute to the field of NLP for Danish, the annotated dataset generated for this research paper will be made publicly available.

Chapter 3

Dataset

This chapter describes the process of creating a dataset for NLP and specifically the task of Stance Detection. The desired properties of the dataset are described in Section 3.1 followed by the sources used for the dataset and a discussion of their validity in Section 3.2 and the scope defined for the dataset in Section 3.3. From here, the process of creating the dataset is considered, with gathering and parsing of data described in Section 3.4, data cleaning described in 3.5 and finally the manual labelling process in 3.6. Section 3.7 contains an overview of key figures for the dataset and considerations regarding these based on a goal of attaining objectivity in the dataset.

3.1 Political quotes and their stances

This section describes the properties desired of the generated dataset, and how these properties are achieved. To enable efficient cleaning of the raw data, it is sought to create two separate datasets; one containing quotes and one containing articles, connected through an article ID. This allows flagging of articles falsely identified as of interest to our research from the article dataset, and removing the quotes extracted from these articles from the quote dataset. The initial use of two separate datasets could be avoided by flagging false positive articles before extracting quotes, but the use of two separate datasets allows performing flagging of articles concurrently with the setup and testing of automated data cleaning procedures for quotes, thus making the split dataset approach preferable.

The testing of NLP models on the dataset requires as a minimum the quote text to be included in the dataset, to allow extraction of text features. Context features are to be extracted from the politicians being quoted and their party affiliation, leading to their inclusion as well. To allow statistical analysis and model tests on specific subsets of the data, the date, media outlet, topic and sub-topic of the quote are also included. The addition of these features enables a number of interesting angles of analysis, including but not limited to:

- Change in a given politician's stance towards a topic over time
- Variation in stance among politicians in a party, identifying outliers within the party
- Representation of each political party in the media, and possible correlation between representation and stance of the parties

Thus the two datasets will be defined as shown in Table 3.1 and Table 3.2.

qID	party	politician	date	quote	fp	sTopic	fan	aID	topic
1	lorem	ipsum	01.01.18	dolor	0	p	f	1	integration
2	lorem	ipsum	01.01.18	dolor	1			1	integration
3	lorem	ipsum	01.01.18	dolor	0	c	n	2	integration

Table 3.1: Quote dataset with placeholder quote values, qID representing quote ID, aID representing quote ID, fp representing false positive, fan representing the quote label as *for*, *against* or *neutral* and sTopic representing subtopic

articleID	topic	title	text	mediaOutlet	fp
1	integration	lorem	ipsum	Ritzau	0
2	integration	lorem	ipsum	Ritzau	1
3	integration	lorem	ipsum	Ritzau	0

Table 3.2: Article dataset with placeholder article values, fp representing false positive

3.2 Data sources

Raw data for the dataset was generated based on articles loaded from the Infomedia media archive. Infomedia maintains Denmark's biggest media archive, containing articles from a long list of media outlets, including but not limited to trade journals, newspapers, magazines, local journals, web media and transcriptions of radio- and tv-programs, primarily from Danish news sources, which are all continually updated throughout the day. [Infomedia, 2019] Seeing as Infomedia does not actively in- or exclude articles and broadcasts from their database, but rather let the media outlets do this, the use of Infomedia as middle man for attaining data is not assumed to have any effect on the objectivity of the gathered data.

The Infomedia archive allows advanced searching using Insight Query Language (IQL) strings, which resemble the more common Boolean search strings. The determination of optimal search strings is discussed in Section 3.3.

3.2.1 Determining media outlets to be included

When determining which media outlets to be included in the dataset, several factors were taken under consideration, including:

- Objectivity of the media outlet
- Representativity of included media outlets in regards to political orientation
- Complexity of the PDF file generated for the media outlet by Infomedia

In an analysis published in 2015, Heidi Jønch-Clausen, former adjunct at Syddansk Universitet, shows that objectivity in reporting of politics in Danish media was in steady decline in the period 1990-2011, and that a growing tendency towards analysis and interpretation is gradually diluting the objective core of the genre of political reporting.

[Jønch-Clausen, 2015] In regards to both goals of this thesis project; creation of a dataset which is generally applicable for analysis within the fields of NLP, political sciences and sociology, and creation of a stance detection model which is generally applicable across political parties and politicians, a high level of subjectivity within the data can be detrimental. If the dataset, for instance, is to be used to analyse shifts within political discourse across party lines, use of data from subjective media outlets would result in parties with certain political opinions being represented more favourably than others, depending on the political orientation of the journalist and media outlet, thus risking making the research results skewed. Similarly, a stance detection model built on quotes extracted from a subjective news outlet would learn this subjectivity, and be likely to have reduced accuracy when tested on data from an objective news outlet, or a news outlet with another political orientation. Despite direct misquotation being illegal, journalists are free to include and exclude quotes based on which politician makes the statement, which political party this politician is from, and is furthermore largely free to contextualize the quote in a way they see fit. This leaves the dataset open to subjectivity due to analysis and interpretation, as discussed by [Jønch-Clausen, 2015].

The possibility of subjectivity influencing the choices of journalists, when choosing which quotes to include in articles, increases the importance of assuring representativity of political orientation and political parties in the generated quote dataset, to avoid generating a skewed dataset.

To tackle these first two factors, two approaches were considered. First off, media outlets might be chosen from each end of the ideological spectrum, in an attempt to let the subjectivity of the outlets cancel each other out. Here an analysis of Danish media organizations, such as that presented by Stig Hjarvard, Professor at Københavns Universitet, in 2007, might be applied in choosing which media outlets to include. Based on Figure 3.1, 3.2 and 3.3, the newspapers Politiken and Information would be strong candidates to represent the left wing, and Børsen and Berlingske Tidende would be strong candidates for the right wing.

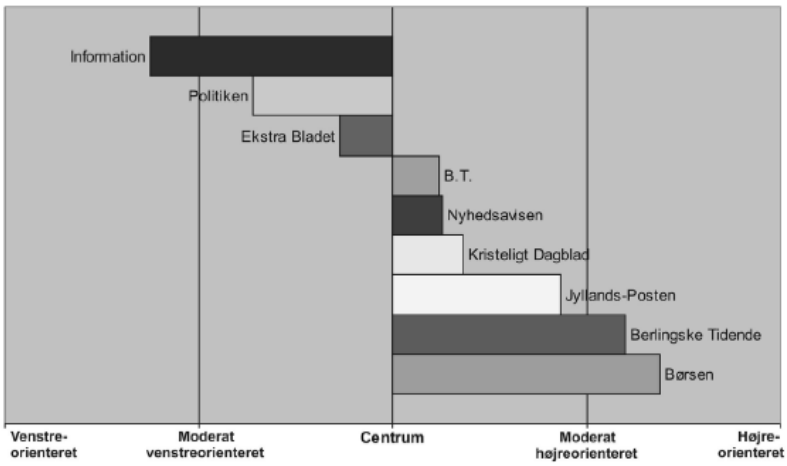

Figure 3.1: Readers' assessment of the political tendency in the papers' editorials, [Hjarvard, 2007]

An alternative to this solution would be to identify media outlets that are generally objective, and use these for the dataset. Looking at [Hjarvard, 2007], examples of news

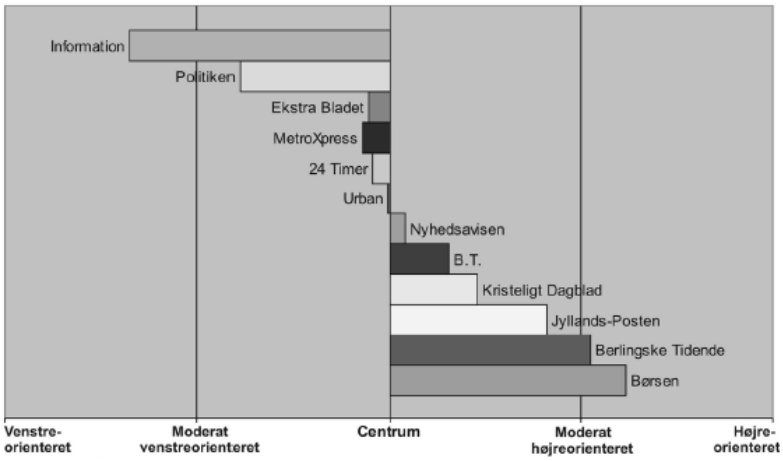

Figure 3.2: Readers' assessment of the political tendency in the papers' debates, [Hjarvard, 2007]

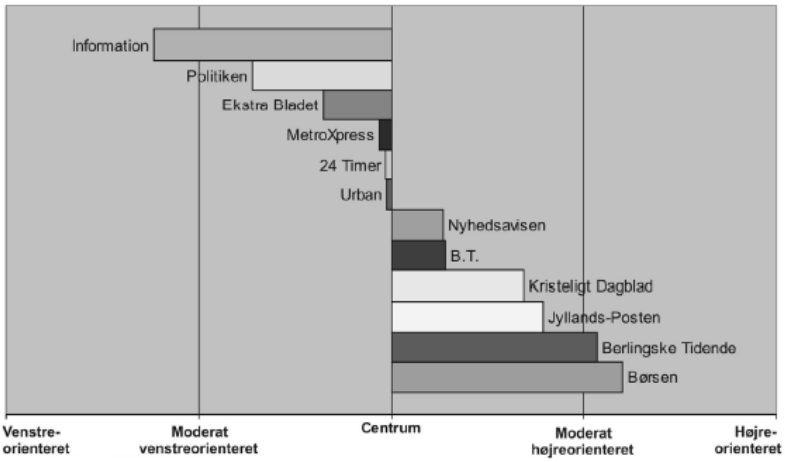

Figure 3.3: Readers' assessment of the political tendency in the papers' journalistic articles, [Hjarvard, 2007]

papers within this category would be 24 Timer, Urban and Nyhedsavisen, all of which have, unfortunately, closed since the publication of the article.

Examining examples of articles from the four media outlets considered for the first approach, we see that both Politiken and Børsen only offer article summaries through Infomedia, as shown in Section 8.1. Furthermore, all articles from these two outlets are behind paywalls on the media outlets' own websites. Articles on Infomedia only containing summaries is an issue for a large number of media outlets, including but not limited to the two aforementioned publications, Jyllands Posten, DR and BT, all of which are major Danish media outlets, limiting the pool of potential outlets significantly. Examples of articles from Jyllands Posten, DR and BT are likewise included in Section 8.1.

The large number of media outlets which do not give access to their full articles through Infomedia, combined with how many of these have their articles behind paywalls at the media outlet websites, complicates the process of attaining objectivity in the dataset by creating representativity through inclusion of media outlets on different ends of the

political spectrum. To make this approach viable, would require including a large number of smaller media outlets, which have full articles available at Infomedia. None of the media outlets sampled though Infomedia yield articles in PDF format that are more complex to parse than the others, and many make use of the same quote formatting, specifically indicating quotation with either "≫" or " - ", as can be seen in Section 8.1. This increases the viability of an approach relying on several smaller publications, as it might be possible to build a PDF parser which generalizes across several media outlets, identifying and scraping quotes.

Many of the smaller media outlets are local newspapers, such as Fyens Stifttidende, Skive Folkeblad, Århus Stifttidende and Jyske Vestkysten. Thus, geographical representativity needs to be taken into consideration alongside representativity within political ideologies. Seeing as these local newspapers are smaller publications, they have fewer articles than their national counterparts, as seen in Table 3.3, and this becomes a larger issue, when looking at less frequently quoted politicians, as seen in Table 3.4. Using several smaller media outlets increases the risk of media outlets having very few or no articles related to a given search string, leaving the geographic area related to that media outlet under-represented in the dataset for the given politician and topic, thus making the dataset lack geographic representativity

Media outlet	**Available articles**
Ritzau	172
Berlingske	131
Jyllands Posten	155
Berlingske Tidende	95
Fyens Stifttidende	69
Skive Folkeblad	62
Jyske Vestkysten	57
Århus Stifttidende	56

Table 3.3: Available articles for selected media outlets, when searching for articles related to Mattias Tesfaye and immigration policy

Media outlet	**Available articles**
Ritzau	14
Berlingske	24
Jyllands Posten	27
Berlingske Tidende	10
Fyens Stifttidende	7
Skive Folkeblad	6
Jyske Vestkysten	5
Århus Stifttidende	0

Table 3.4: Available articles for selected media outlets, when searching for articles related to Nicolai Wammen and immigration policy

For the results in the two tables above, the search string "'[POLITICIAN NAME]' AND (integ* OR indva* OR asyl*)" has been used.

3.2.2 Ritzau as objective data source

As an alternative to the use of several small objective media outlets, using a single larger publication with a high level of objectivity is considered, namely Ritzau. The objectivity of Ritzau is underlined by a number of factors. Firstly, Ritzau is owned by a conglomerate of media outlets, representing all areas of the political spectrum presented by [Hjarvard, 2007], including but not limited to Børsen, JP/Politikens hus, Danmarks Radio and Information, which is assumed to make the different political orientations of the owners cancel each other out [Ritzau, 2019]. Secondly, Ritzau is quoted and referenced by a very large number of media outlets from across the political spectrum, as shown in Section 8.2. From Section 8.2 it is also apparent, that many media outlets go even further, and publish articles directly copied from Ritzau, thus supporting the assumption that the media outlet is objective.

Using Ritzau as data source yields two additional benefits, aside from attaining objectivity of the extracted data. Firstly, using media outlets which quote Ritzau might result in extraction of the same or very similar articles and quotes based on Ritzau articles from two separate sources. Thus, seeing as a large number of media outlets use Ritzau as source, extracting data directly from Ritzau rather than other media outlets can be expected to eliminate a risk of data duplication. Secondly, using only a single media source reduces the size of the task of building tools for pdf parsing, when compared to building a tool for each media outlet. It is true that the overhead of learning the libraries used in the parsing is the same regardless of the number of media outlets included, but the parsing tool would need to be expanded for each new inclusion of a media outlet. For these reasons, Ritzau will be chosen as the data source used for this project, while all cleaning and parsing tools will be made public, to increase ease of inclusion of further media outlets in the future.

3.3 Delimitation of search space

To create a precise dataset of quotes, for which it is possible to create meaningful statistical analysis, and from which a neural network is able to learn specific patterns of speech, it is important that the boundaries of the dataset are precise. Within this section is presented the rationale behind the choice of which topic to center the dataset around, as well as which politicians to include quotes from. As this project started in January 2019, it was decided to include all of 2018 in the dataset. Considerations were made regarding including earlier years, but were discarded due to the increased workload in data cleaning and labelling this would entail. Similar considerations were made regarding continually expanding and updating the dataset during 2019.

3.3.1 Choice of stance topic

Given a broad topic such as "foreign policy", defining what constitutes a quote as *for* or *against* the topic quickly becomes a complex matter, thus increasing the risk of subjectivity in labelling. On the other hand, it would be an easy task to determine whether a quote is *for* or *against* increasing the amount of vegan food being served in kindergartens, but for such a narrow topic, acquiring a useful amount of data would constitute a significant

challenge. Thus, the definition of boundaries of the topic around which the dataset is built, need to be clear, while walking the line between too broad and too narrow.

A shortlist of possible topics to include in the dataset was attained based on an opinion poll performed by election researcher Rune Stubager on behalf of Altinget, seeking to identify the topics most important to the Danish population, when voting in the next election. The poll was performed in the period 15th-19th November, 2017. In [Kvalvik, 2017], the five most important topics were identified as health policy, social policy, immigration policy, crime and justice policy and finally environment and climate policy.

Out of these, social policy and crime and justice policy are identified as bad fits for the goals of this project. Social policy includes both educational policy, kindergarten and daycare policy and to some extent policy regarding care for the elderly, all of which constitute topics of significant size on their own. *For* and *against* quotes would need to be identified differently for each of these sub-topics, increasing the complexity of labelling and thus the risk of mislabelling or subjectivity in labelling, as well as introducing a risk of bad translation of learning from one sub-topic to others. Crime and justice policy would be easier to define a clear *for* and *against* quote for, for instance using intention of implementing stricter or more lenient punishment as a strong indication of for/against, but it is expected that the stance of politicians and political parties would vary a lot, based on which specific area of legislation the quote concerns. Furthermore, the discourse used for each area of legislation is expected to vary, making learning across areas of legislation difficult. As an example of this, we can observe that Radikale Venstre supports the legalization of hash, seen in the quote below from member of parliament Zenia Stampe.

> *Vi ønsker ikke, at unge ender som hashbrugere eller -misbrugere. Men vi må også erkende, at virkelighedens misbrugs- og bandeproblemer kræver mod til nye løsningsmodeller. Forbudspolitikken har jo ikke løst udfordringerne. Derfor går vi ind for et forsøg, hvor hashsalget flyttes fra kriminalitetens skygger til statskontrollerede butikker landet over.*
> Zenia Stampe, [Ritzau, 2018c]

On the other hand, Radikale Venstre also supports stricter legislation in regards to sexual abuse, exemplified in the quote below by Bo Nissen, head board member of Redikale Venstre København, in regards to sexual abuse through negligence.

> *Vi vil sørge for, at man i seksuelle forhold kan dømmes for at være dummere, end politiet tillader.*
> Bo Nissen, [Mansø, 2018]

Environment and climate policy likewise constitute a complex topic, as few politicians would go on record as being against our environment, or supporting worsening our climate. Thus, the definition of *for* and *against* labels would depend on a subjective distinction by the individual performing data labelling as to which are enough "for" our climate to earn a for-label, and which are not enough "for", thus earning an against or negative label, or simply accepting that a vast majority of quotes would be within the *for* category. To avoid this risk of creating an either largely subjective or largely skewed dataset, the topic is discarded.

The topics healthcare policy and immigration policy both constitute strong candidates for inclusion, and share a number of features. First off, it is possible to define *for* and *against* labels as whether a politician indicates willingness to increase, or a wish to decrease, public spending within the area. Furthermore, both topics include a subtopic of centralization. Within healthcare policy this would take the form of quotes regarding, for instance, creation of "Supersygehuse" - large, centralized hospitals. Within integration policy the subtopic takes the form of quotes regarding whether immigration should be handled on an international/EU-level, or a national level.

The final decision between these two topics has been made based on personal preference, and a larger interest within the area of immigration policy lead to this topic being chosen over healthcare policy.

3.3.2 Specification of the topic of immigration policy

To formulate a search string to use in gathering articles for the topic of immigration policy, it is necessary to define keywords connected to this topic. To attain a broad view of the topic, both integration, immigration and asylum are included as areas of interest, to be included in the dataset. A more narrow definition could be used, excluding asylum to only include quotes regarding foreigners coming to the country of their own free will, but it is assumed that these words are often used interchangeably, and that it would not be possible to assure that only this subset of quotes are included, by excluding asylum-related keywords from the search string. The inclusion of all three aspects of immigration policy creates the risk of making the dataset unclean, and the discourse more difficult for the ML models to learn, seeing as the discourse might vary between the sub-topics. If this is shown to be the case, labelling of quotes with the three sub-sets, splitting the dataset and creating models trained on specific sub-sets would solve the issue. Thus, all three areas of immigration policy are included in the dataset, resulting in the following search string.

'[POLITICIAN NAME]' AND (integ OR indva* OR asyl*)*

Asterisks indicate wildcards, meaning that integ* will both match integration, integrere, integreret etc.

3.3.3 Choice of politicians

To accurately represent the full spectrum of Danish legislative politics, politicians from all political parties with seats in parliament are included in the dataset. From each party, ten politicians have been chosen for inclusion in the dataset. Politicians with seats in parliament have been prioritized over those without seats. For the parties with more than ten politicians in parliament, prioritization has been made as follows:

1. Ministers
2. Party heads
3. Speakers
    (a) Speakers within the five top topics of interest to the Danish population as presented by [Kvalvik, 2017]

(b) Speakers not within the five top topics

4. Members of parliament without speaker positions

A number of special inclusions were made, outside of the parameters described above. Johanne Schmidt-Nielsen of Enhedslisten and Zenia Stampe of Radikale Venstre were on maternity leave during a part of 2018. Despite this reducing the number of available quotes for the two politicians, they were both included, seeing as they have held speaker positions within their party in 2018, and are both influential voices within their party. Radikale Venstre only has nine candidates when combining speakers and members of parliament and speakers, and thus Jens Rohde has been included as EU parliamentarian. Finally, Socialistisk Folkeparti only has seven candidates when including members of parliament. Therefore Margrethe Auken, EU parliamentarian, Signe Munk, Næstformand and Rikke Lauritsen, candidate for EU and Danish parliament, have been included.

The question of gender representativity

A ML model not only learns useful patterns for solving a given task when learning a dataset, but also the inherent bias within that dataset. To avoid building a neural network biased towards any one gender, it is therefore necessary to consider the inherent bias within the dataset.

Two approaches were taken into consideration regarding securing the gender neutrality of the dataset and the classification model built based on the dataset. To optimize the dataset to include the most influential voices within Danish politics, politicians would be chosen based on who hold relevant speaker positions within the parties, as well as who is the party heads of the parties. If one gender is more strongly represented within these positions, this would yield a dataset that is biased towards that gender, and possibly a classification model which more accurately classifies quotes from the over-represented gender. To optimize the dataset for gender equality, less weight would be put on the positions of the politicians within the party, instead prioritizing including an even distribution of politicians within each gender from each party. This solution would possibly increase the general applicability of the classification model, as it would more accurately be able to classify stance in quotes from all genders. The downside to this approach is, that the dataset would become skewed, misrepresenting the actual gender distribution within the Danish political system. Furthermore, it can be assumed that politicians who hold more power within a party are more likely to be asked for comments and thus be quoted. This would possibly lead to a smaller dataset, if increased focus was put on attaining an equal gender distribution.

By applying the first approach, and using the prioritization metrics defined in Section 3.3.3, the gender distribution presented in Table 3.5 is attained. It is clear that the approach as predicted creates a skewed gender distribution within the dataset, but the skewness is judged to be within a reasonable margin, with 58 % male and 42 % female politicians. It is worth noting, that left-wing parties generally seem to have more male politicians in parliament and speaker positions, with a count of 70 % males for both Alternativet, Enhedslisten and Socialdemokratiet.

A full list of the included politicians, as well as the number of quotes identified for

	Male politicians		Female politicians	
Party	Count	%	Count	%
Alternativet	7	70	3	30
Dansk Folkeparti	5	50	5	50
Det Konservative Folkeparti	6	60	4	40
Enhedslisten	7	70	3	30
Liberal Alliance	6	60	4	40
Radikale Venstre	5	50	5	50
Socialdemokratiet	7	70	3	30
Socialistisk Folkeparti	3	30	7	70
Venstre	6	60	4	40
Total	**52**	**58**	**38**	**42**

Table 3.5: Gender distribution of dataset for each political party and total

each of them based on initial data parsing and before cleaning, can be found in Section 8.3.

3.4 Data gathering and parsing

Data was initially planned to be gathered automatically using a scraping script on the Infomedia media archive to enter the search string for each chosen politician, pass over each article and scrape its contents. However, after contact with Infomedia, this was found to be against the company's terms and conditions, and a manual approach was applied instead. The search string was entered for each politician, the time period defined, and a PDF-file of search hit articles downloaded. This manual approach resulted in the necessity of building a scraper for PDF format rather than web, to extract the article text and enter it to the dataset.

3.4.1 Parsing PDF data

Parsing the PDF files generated from Infomedia showed to be a challenge. A number of Python libraries were tested for this task, including PyPDF, textract, PDFQuery and PDFMiner, all of which resulted in combinations of empty text objects, NULL values and plain out error messages. This is thought to be due to some combination of the inclusion of the Danish letters Æ, Ø and Å in the articles, encoding un-interpretable by the libraries and the somewhat non-trivial PDF file setup which includes headers, footers and page-numbering.

The least faulty implementation, which has been used for creation of the dataset for this thesis, makes use of PDFMiner. Using this library, the letter combinations 'ff', 'ft' and 'tf' are parsed as NULL values, which is handled by removing NULL values from the text objects, creating a list of words missing "ff", "ft" and "tf" during quote labelling, and using search and replace to manually enter the letters back into the words.

As can be seen in Section 8.4, the PDF files hold a large amount of text not relevant to our dataset, including Ritzau logos, terms and conditions, a link to Infomedia, word count

and article ID in the Infomedia database. The article text is identified using a rules-based scraping approach, and quotes are identified within this text by the use of the symbols ”-” and ”≫”, which Ritzau make use of for indicating an upcoming quote.

Filler sentences and phrases occurring after quotes are identified using a list of common indicators of such phrases, including words like ”says, said, stated, wrote”. Filler phrases are generated by pairing all filler indication words with politicians' names and pronouns, and all occurrences of the generated pairings are removed from the quotes. Examples of filler phrases can be found in Figure 3.4, as ”, siger han” and ”, siger Kristian Thuelsen Dahl”.

Kristian Thulesen Dahl mener, at regeringen svarede "tvetydigt" på Dansk Folkepartis kritiske spørgsmål om erklæringen.

- De (svarene, red.) bar præg af, at man forsøger både at blæse og have mel i munden, siger han.

- På den ene side siger regeringen, at det er en god erklæring, fordi man kan presse afrikanske lande til i højere grad at tage deres egne borgere hjem.

- På den anden side siger regeringen, at erklæringen ingen betydning har i de dele, der kunne genere os selv. Dér kan vi bare lade være med at tage højde for erklæringen. Men man kan ikke gøre begge dele, siger **Kristian Thulesen Dahl**.

Figure 3.4: Example of quotes in Ritzau article PDF files generated through Infomedia

3.4.2 Identifying quotees

Each politician included in the dataset has a related PDF file containing the articles containing their name. When parsing a PDF file related to a given politician, only quotes related to that specific politician are to be extracted. This is achieved using regular expression search, first looking at the filler phrases connected to the quote. If the phrase matches a regular expression containing a filler word and the politician's name, the quote can be attributed to that politician. However, if it contains a filler word and not the politicians name or a pronoun, it is assumed that the quote is made by another than the politician of interest. If looking for quotes made by Kristian Thulesen Dahl, this would identify the first and second quote of Figure 3.4 as possible matches, and quote 3 as a certain match.

Regular expressions are likewise used to identify situations as that of the first quote of Figure 3.4, where it is evident from the non-quote sentence prior to the quote, that the following quote is of interest. This is done by looking for combinations of the name of the politician of interest and filler words in all non-quote sentences. If such a sentence is observed a flag is raised, and if the next sentence is identified as a quote, and that quote is not made by the wrong quotee, it is identified as a quote by the politician of interest.

Finally, quotes of the type as the second quote of Figure 3.4 are identified by looking for quotes in a line. It is a relatively common practice to not state the quotee after a quote, if it is directly after a quote from the same quotee. Such quotes are identified by raising a flag when a quote by a politician of interest is identified. If the next sentence is also a quote, and that quote is not made by another than the politician of interest, that

quote is attributed to him or her.

In-line quotes were found to be highly difficult to extract without including a large number of false positives, and quotes attributed to the wrong quotee, and were therefore excluded from the dataset. Looking at the downloaded articles however, they constitute a very small subset of the total number of quotes, which can also be seen from Sections 8.1, 8.2 and 8.4.

The most significant indicator of bugs in the PDF parsing code was found to be duplicate quotes attributed to different quotees, and this was used to identify and eliminate a large number of errors.

3.5 Data cleaning

3.5.1 Creating automated cleaning procedures

The process of creating scripts for data cleaning was performed in a circular manner, as visualized in Figure 3.5. After generating a dataset using the procedures specified in Section 3.4, quotes in the dataset were manually tested against the PDF files from which they were parsed, to ensure that the quotes were parsed properly and attributed to the correct politician. Quotes were checked until an error was identified, after which the source of the error was removed in the code, and another dataset could be generated. The dataset was deemed acceptably error-free after no errors were identified when checking the quotes of Martin Henriksen and Inger Støjberg, two of the most quoted politicians included in the dataset. It was during this process, that the lists of filler words, flags insinuating wrong or correct quotees, identifiers for non-article text and flags insinuating an upcoming quote, that were all used in the data parsing process, were generated.

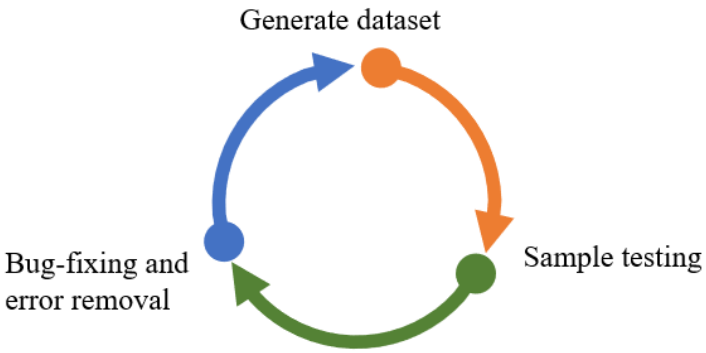

Figure 3.5: Visualization of circular process for development of automated data cleaning

During sample testing, it became obvious that the dataset contained a large number of duplicate quotes. This was found to be due to the fact that Ritzau includes several versions of the same article in the Infomedia database, each version having a slightly changed title and a few added sentences. For some versions, the changes are substantial, and for this reason old article versions were not discarded as duplicates. Duplicate quotes, resulting from several versions of the same article containing the same quote, were identified and removed.

3.5.2 Identifying false positives

Each collected article was skimmed through and judged by its relevancy to the topic of immigration policy, labelling articles with immigration policy as primary topic as true positives, and included in the dataset. Articles that did not have immigration policy as primary topic, were re-considered based on the quotes contained in the articles. If an article was found to contain one or more quote directly on the topic of immigration policy, they were labelled as true positives, the rest being labeled as false positives. In a few cases, articles were collected which contained only quotes, for instance several articles summarizing politicians' response to Lars Løkke Rasmussen's New Year speech. Such articles, containing a very large number of quotes not related to the topic of immigration policy, and a few on that topic, were flagged as false positives, as their inclusion would increase the number of quotes to be removed manually as false positives significantly, while only increasing the number of true positive quotes marginally.

A subset of the article dataset called "Ritzau's tophistorietjeneste" was removed, as the articles within this subset are copies of articles from other media outlet. Inclusion of this subset would raise issues regarding objectivity, as discussed in Section 3.2.

Simultaneously with determination of quote labels, it was determined whether a given quote was a false positive. This judgment is highly subjective, and therefore a significant potential source of error in the dataset, and thus in the model built on the dataset. A quote directly referring to immigration, integration, asylum or any aspect of these topics were directly classified as true positives, an example of which is found below.

> *Derfor er målsætningen at få både anerkendelsesprocenten og antallet, der søger asyl, længere ned.*
> Martin Henriksen, [Ritzau, 2018c]

If a quote was clearly not regarding any of the three subtopics mentioned above, it was labelled as a false positive, an example of which would be the following quote by Martin Henriksen:

> *Man skal aldrig være ked af, når man får kritik fra en radikal.*
> Martin Henriksen, [Ritzau, 2018d]

For the remaining quotes, which were not easily determined to be either a true or false positive, the related article was found in the article dataset, and the context of the quote was included in the determination of the relevance of the quote. An example of such a quote would be the following:

> *Alle kan sige, at man skal sætte stolen for døren. Men det er jo bare ord.*
> Martin Henriksen, [Ritzau, 2018b]

3.6 Data labelling

3.6.1 Determining quote subtopic

All quotes not identified as false positives, were first labelled according to two parameters. Firstly, the subtopic of the quote, differentiating between quotes regarding national immigration policy, and quotes regarding centralized immigration policy. The choice of these two subtopics were based on initial attempts at determining collective labelling criteria, which could be applied to all quotes. From these attempts, it was clear that the dataset contained a subset concerning immigration outside of Denmark's borders, which could not be labelled using the same criteria as the rest of the dataset. National immigration policy is defined as policy and topics that concern matters within Danish borders, such as the number of asylum seekers the country takes in, how these are housed, and what requirements should be set for them in regards to taking Danish education and employment. An example of a quote within this subtopic can be found below, which concerns the government's initiative to combat communities that they define as ghettos.

> *Det er godt, at der lægges op til højere straffe og en styrket politiindsats i ghettoer. Men regeringen skal passe på ikke at oversælge sit udspil. Det kan ikke løse alle problemer.*
> Martin Henriksen, [Ritzau, 2018e]

Centralized immigration policy is defined as policy and topics that concern immigration on a European or international level, for example distribution of asylum seekers among the member countries of EU, deterring immigrants at EU's boarders or the sending of immigrant from Denmark to refugee camps in other countries. An example of such a quote can be found below.

> *Den danske regering bør i stedet sige til den italienske regering, at Danmark og Italien i fællesskab kan transportere asylansøgerne tilbage til Afrika, så de kan blive sat af på kyste.*
> Martin Henriksen, [Ritzau, 2018i]

A few edge cases exist, where a quote fits both subtopics. In these cases, a duplicate quote is created, and one is labelled with each subtopic. An example of this is found below, where first half of the quote is concerning the free mobility of labour within EU, and immigration stemming from this, and the second half is concerned with the effect on immigration of legislation changes on a Danish level.

> *Det er oplagt at se på, hvordan vi kan understøtte en højere grad af mobilitet i Europa, så danske virksomheder, der har brug for arbejdskraft, kan få den, uden det betyder den indvandring, som vil følge af at sætte beløbsgrænsen ned.*
> Mette Frederiksen, [Ritzau, 2018g]

3.6.2 Determining quote stance

The choice of labelling convention is based on that applied by [Mohammad et al., 2016] in organizing SemEval-2016 Task 6, which is concerned with the detection of stance within tweets, and the creation of a dataset for this task. Thus three classes are defined, the first called *for*, declaring support of a given subtopic, the second called *against*, and a third, called *neutral* contains both quotes that are deemed to be neutral towards the subtopic, as well as quotes for which a specific stance can not be determined.

To enable the determination of precise criteria for the *for* and *against* labels for the two subtopics, it is necessary to first define the two topics clearly.

Annotation guidelines

The subtopic *national policy* is defined as tightening the policy within the borders of Denmark on the legislative fields of immigration, integration and asylum. Therefore, a quote would be classified as *for* this topic, if it exhibits one or more of the following traits.

- support for higher restrictions on immigrants or asylum seekers entering the country
- support for lowering public benefits to immigrants or asylum seekers
- a wish to get immigrants or asylum seekers to leave Denmark, after they have entered the country
- making demands specifically of immigrants or asylum seekers, for instance regarding taking language courses or job search
- seeking to make immigrants or asylum seekers change their culture or behaviour
- communicating explicitly or implicitly that immigration is a burden to Danish society
- wishing to implement changes in behaviour though negative incentives such as decreased public benefits

Quotes classified as *against* the *national policy* subtopic, on the other hand, will exhibit one or more of the following traits.

- support for lower restrictions on immigrants or asylum seekers entering the country
- support for higher public benefits to immigrants or asylum seekers
- immigrants or asylum seekers are free to stay, after having entered the country
- seeking to making fewer demands of, and give more freedom to, immigrants or asylum seekers
- not seeking to make immigrants or asylum seekers change their culture or behaviour
- communicating explicitly or implicitly that immigration is an asset to Danish society
- wishing to implement changes in behaviour though positive incentives such as increased public benefits

The subtopic *centralization* is defined as yielding decision power to EU, and/or solving more immigration issues on a European or international level, rather than on a national

level, and *for* and *against* labels are thus more clearly defined for this subtopic. A *for* quote would support yielding power, an example of which is found below.

> *Europa har en fælles udfordring med flygtninge og migranter. Vi må have et fælles asylsystem.*
> Rasmus Nordqvist, [Ritzau, 2018a]

On the other hand, an *against* quote be opposed to yielding power, an example of which is found below.

> *Der er for mange spørgsmål, som står ubesvaret hen, og derfor mener vi, at man fra dansk side skal suspendere det samarbejde, indtil der er fuldstændig klarhed over, hvad regeringen har forpligtet sig til på Danmarks vegne.*
> Martin Henriksen, [Ritzau, 2018j]

Resolving grey areas

Not all quotes contain explicit communication of a stance or even clear indicators, like the ones just described. To solve this issue, inspiration is taken from [Mohammad et al., 2016], and the questions given to annotators during the research project. In line with [Mohammad et al., 2016], when labelling quotes, stance is inferred from how the quotee refers to things and people aligned with or opposed to the topic. An example of this would be a politician indicating support towards a ban on the use of burkas, which falls within the subtopic of *national policy*. Seeing as the ban on burkas is a restriction of behaviour, the quote would be labelled as *for*, as the stance of the quote can be induced by proxy. Furthermore, when no clear stance is communicated, and no stance can be determined by proxy, the tone of the quote is analysed, looking at the use of weighted words, for instance describing immigrants as resources, nuisances or in neutral terms.

3.7 The final dataset

Looking at the quote count for the dataset as presented in Table 3.6, it is clear that the dataset is significantly skewed towards the *for* label, containing 57,2 % of the quotes, with 23,4 % labeled as *against* and 19,3 % as neutral when observing the full dataset, and the skewness remains if looking at the two subsets in isolation. Such skewness has shown to be an issue for stance detection models in earlier research, an example of this being the SemEval-2017 competition Task 8, where the dataset contained a majority label with 66 % of the data points in the train set and 74 % of data points in the test set. [Derczynski et al., 2017] For this competition, entrants found the skewed dataset to be a major challenge in the classification task, a few systems such as [Kochina et al., 2017] and [Bahuleyan and Vechtomova, 2017] managing to surpass a basic majority voting system by applying advanced model architecture, batching and extensive feature extraction, while several systems were not able to surpass majority voting.

Another potential issue for the stance detection task is the size of the dataset, as a size of 898 instances might not be sufficient to learn the language patterns within the quotes.

Party	**Subtopic**	**# Quotes**			
		For	Against	Neutral	Total
Alternativet	National Policy		7	2	9
	Centralization	2			2
	Total	2	7	2	11
Dansk Folkeparti	National Policy	187	5	25	217
	Centralization	5	18	7	30
	Total	192	23	32	247
Det Konservative Folkeparti	National Policy	18	1	5	24
	Centralization	2			2
	Total	20	1	5	26
Enhedslisten	National Policy	3	26	5	34
	Centralization		4	1	5
	Total	3	30	6	39
Liberal Alliance	National Policy	6	6	6	18
	Centralization				0
	Total	6	6	6	18
Radikale Venstre	National Policy	7	68	18	93
	Centralization	6		1	7
	Total	13	68	19	100
Socialdemokratiet	National Policy	92	20	42	154
	Centralization	7	1	1	9
	Total	99	21	43	163
Socialistisk Folkeparti	National Policy	5	26	2	33
	Centralization	2			2
	Total	7	26	2	35
Venstre	National Policy	144	14	54	212
	Centralization	38	1	8	47
	Total	182	15	62	259
All parties	National Policy	462	173	159	**794**
	Centralization	62	24	18	**104**
	Total	**524**	**197**	**177**	**898**

Table 3.6: Quote count overview for dataset

3.7.1 Assessing representativity in the dataset

It is clear from Table 3.6 that the quotes in the dataset are far from equally distributed among the parties in parliament, with the majority of the quotes coming from Dansk Folkeparti and Venstre, followed by Socialdemokratiet and Radikale Venstre, with the remaining five parties being under-represented. This is likely to be due to an under-representation of several parties in the media picture, when it comes to immigration policy, which is transferred to the dataset, and will likely result in the stance detection model learning the patterns of quotes from the four over-represented parties better than those of the five remaining parties.

Dividing parties based on their placement on the political axis, defining Alternativet, Enhedslisten, Radikale Venstre, Socialdemokratiet and Socialistisk Folkeparti as left-wing parties and Dansk Folkeparti, Det Konservative Folkeparti, Liberal Alliance and Venstre as right-wing parties, a skewness towards the right-wing parties within the dataset can be observed, as seen in Table 3.7. For the *national policy* subset we observe an over-

representation of right-wing parties with 59 % of the total quotes, for the *centralization* subset an over-representation of 76 % for right-wing parties can be observed and for the full data-set right-wing parties are over-represented with 61 % of the quotes. This constitutes a weakness in any classifier built on the dataset, as the classifier will likely be better at recognizing quotes from right-wing than from left-wing parties.

		Quote #		
		Left-wing	Right-wing	Total
National Policy	For	355	107	462
	Against	26	147	173
	Neutral	90	69	159
	Total	**471**	**323**	**794**
Centralization	For	45	17	17
	Against	19	5	24
	Neutral	15	3	18
	Total	**79**	**25**	**104**
Full dataset	For	400	124	479
	Against	45	152	197
	Neutral	105	72	177
	Total	**550**	**348**	**898**

Table 3.7: Quote count divided by political axis, for each subset and total

Similarly, a skewness towards the male gender can be observed in the dataset, as shown in Table 3.8, which is likely to translate into a bias towards males in any classification model trained on the dataset. The over-representation of males is present in both subsets, with 58 % for *national policy*, 72 % for *centralization* and 60 % for the full dataset.

		National Policy				**Centralization**				**Both subtopics**			
		F	A	N	Total	F	A	N	Total	F	A	N	Total
Gender	Male	276	102	85	**463**	40	22	13	**75**	316	124	98	**538**
	Female	185	70	75	**330**	23	2	5	**30**	208	72	80	**360**

Table 3.8: Quote count divided by gender

Chapter 4

Methodology

This chapter first describes core concepts within ML in Section 4.1 and deep learning in Section 4.2 applied within this project, including basic nerual network structure, recurrency, LSTM architectures and general terminology. Section 4.3, the design of the three stance detection models built to classify the quotes within the generated dataset, as well as the considerations made during the design process, are presented. This includes considerations regarding choice of loss function in Section 4.3.1 and optimizer in Section 4.3.2, and descriptions of how each of the three models stand apart, found in Sections 4.3.4, 4.3.5 and 4.3.6. Finally, considerations regarding the class imbalance problem observed in the dataset in Chapter 3 are presented in Section 4.4.

4.1 Machine learning

Within the field of ML, a distinction is made between supervised, unsupervised and reinforcement learning approaches. Decisions regarding whether a supervised or unsupervised approach is applicable, is based on the available data. Supervised learning approaches requiring both an input dataset and the desired output, evaluating how well the system performs based on a comparison between system predictions and desired output, unsupervised learning requires only input data, evaluating performance based on a metric such as a distance metric for clustering tasks. Reinforcement learning constitutes teaching a system through a series of rewards or punishments based on its predictions. Thus the applicability of such a system is situational, and can not be decided entirely based on the availability of data [Russell and Norvig, 2016]. Within this thesis, supervised learning is applied, using a quote as input, and a stance label as the desired output.

4.1.1 Supervised learning

The task of a supervised learning system can be defined as minimizing the discrepancy between actual labels for a dataset, and the predicted outputs generated by the system based on said dataset, passing some feedback to the system based on its performance as visualized in Figure 4.1. To ensure that a system can predict labels for data other than that on which it is trained, it is evaluated on a held-out portion of the dataset. The first dataset, on which a system is trained, is named the training set, and the dataset on which is

evaluated is called the test set [Russell and Norvig, 2016]. Furthermore, a development set might be applied for system testing during development as well as hyperparameter tuning, to avoid optimization of the system to fit a specific dataset too early in the development process. Due to the small size of the dataset generated for this paper, no development set will be utilized, as this would reduce the sizes of the test and dev sets. The dataset will be split with 80 % being used as training data and 20 % being used as test data.

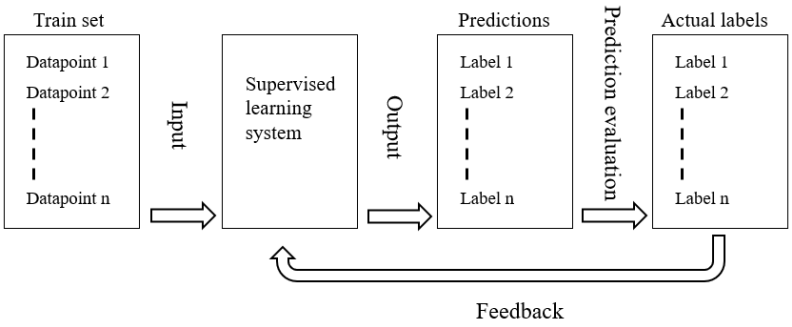

Figure 4.1: Process of training a supervised learning system

Supervised learning tasks are defined as either classification; predicting a label from a finite set of values based on a given input, or regression; predicting a number from an infinite set of values, based on a given input. [Russell and Norvig, 2016] Stance detection, the goal of this thesis, is thus a classification task.

4.1.2 Evaluation measures for classification

Performance of a classification system is evaluated by its ability to correctly predict labels for data-points within a given dataset. In recent works on stance detection, the measures *precision*, *recall* and *F1* have been applied to evaluate this ability [Augenstein et al., 2016, Zeng et al., 2016, Lai et al., 2016, Ma et al., 2018, Skeppstedt et al., 2017, Iyyer et al., 2014, Enayet and El-Beltagy, 2017, Li et al., 2017]. In this section, a primer is given on these evaluation measures, how they are calculated and how they are used, followed by an introduction to *confusion matrices*, all of which will be applied in the evaluation and comparison of classification systems within this thesis. Within this section we let TP refer to True Positives, correct predictions of a true-label for a given class, FP to False Positives, mis-predictions of a true-label for a given class, TN to True Negatives, correct predictions of a false-label for a given class and finally FN to False Negatives, mis-predictions of a false-label for a given class.

Accuracy is the percentage of predictions a given model predicts correctly, calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Precision is the ratio of correctly predicted true-labels to all predicted true-labels, calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Recall is the ratio of correctly predicted true-labels to the actual number of true-labels in the dataset, calculated as:

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

F-score is an umbrella term, containing an array of evaluation measures combining precision and recall. A simple implementation of this score called *F1*, as described in [Jurafsky and Martin, 2018], calculates F-score as:

$$F1 = 2 \times \frac{precison \times recall}{precision + recall} \tag{4.4}$$

The three evaluation measures presented above, work seamlessly for binary classification, but to allow for an evaluation of the full system, and not just of the classification properties for each class, it is necessary to combine the scores for all classes into a single measure. Within this paper, we consider both micro- and macro-averaging to solve this task.

Micro-averaging consists of pooling TP, FP, TN and FN values across all classes, and calculating evaluation measures using these pooled values. This averaging allows over-representation of classes in a dataset to be reflected in the averaged measure [Jurafsky and Martin, 2018], and should be the measure of choice, if stronger representation indicates bigger relevance in terms of results. A Micro-average precision would be calculated as:

$$Precision_{\text{micro}} = \frac{TP_{\text{pooled}}}{TP_{\text{pooled}} + FP_{\text{pooled}}} \tag{4.5}$$

Macro-averaging consists of averaging the value across all classes, thus weighting each class equally, regardless of the distribution of the dataset, making it most applicable when all classes are deemed equally relevant, regardless of their representation in the data. A Macro-average precision for a three-class classification would be calculated as:

$$Precision_{\text{macro}} = \frac{Precision_{\text{A}} + Precision_{\text{B}} + Precision_{\text{C}}}{3} \tag{4.6}$$

Confusion Matrices visualize classifications of a model across all labels, showing correct classifications as well as misclassifications, and how the data-points were misclassified. It yields insight into which classes a given model has difficulty classifying, and how these are misclassified. It consists of a matrix, with the dimensionality [number of classes]×[number of classes], with actual labels presented horizontally and predicted labels presented vertically.

4.1.3 Random Forests and Bayesian classifiers

A Bayesian classifier is a probabilistic model which calculates a probability distribution for each possible outcome based on data input, and performs tasks such as classification by returning the most probable outcome. The basic Bayesian model, Naive Bayes, assumes complete independence among features of the input data, an assumption that is often

proved faulty, yielding sub-optimal results for the classifier [Zhang, 2004]. Due to this, the slightly more advanced Gaussian Bayes Classifier is used as benchmark within this thesis. Gaussian Bayes extends the assumption that input data is independent with the assumption that it is distributed along a Gaussian distribution, in some examples yielding stronger results [Zhang, 2004, Zeng et al., 2016].

Random Forests are an extension of the tree classifier, generating a large number of tree classifiers initiated using randomized input, a so-called ensemble, each tree voting for the most popular class, thus letting the forest converge on the correct prediction, as the number of trees grow. [Breiman, 2001] Tree classifiers function by creating a structure that at each node splits a given dataset by some value, to most optimally separate the classes within the dataset. A simple three-class tree is illustrated in Figure 4.2 which, based on hair-colour and gender, can predict the name of an individual, given a very specific dataset. A dataset complex enough to require a Random Forest would be likely to produce deeper and thus more complex tree classifiers.

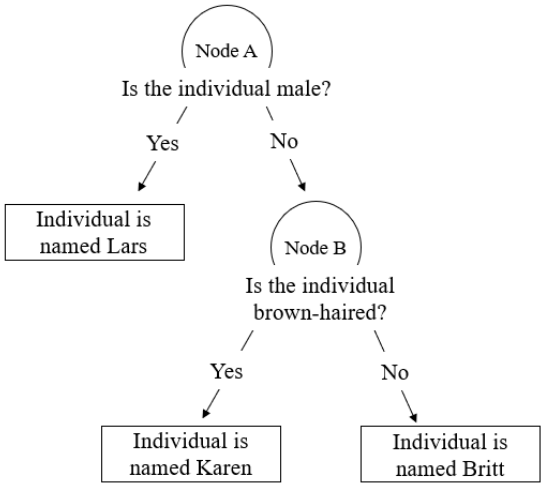

Figure 4.2: Simple tree classifier visualization

Both models are initialized with default values from the scikit learn Python library. For the Gaussian Naïve Bayes this means that a variance smoothing of 1e-9, and for the Random Forest it means that 10 tree estimators are created in the ensemble, no max depth or maximal number of leaf nodes is applied and a minimum of one sample at a leaf node is required for a split.

4.2 Deep Learning Models

One of the most basic deep learning model implementations is the feed forward neutral network. A feed forward neural network contains an input layer, representing managing data input, a number of hidden layers, and an output layer, translating the information generated through the hidden layers into something that is interpretable on a human level, such as a class label for the input data, as visualized in a simplified format in Figure 4.3. Within the hidden layers, input data is run through computations in the form of an activation function, weighting on the input in so-called nodes, using values not given in the data, extracting the information that is most useful in describing relations within the

data, and finally[Goodfellow et al., 2016]

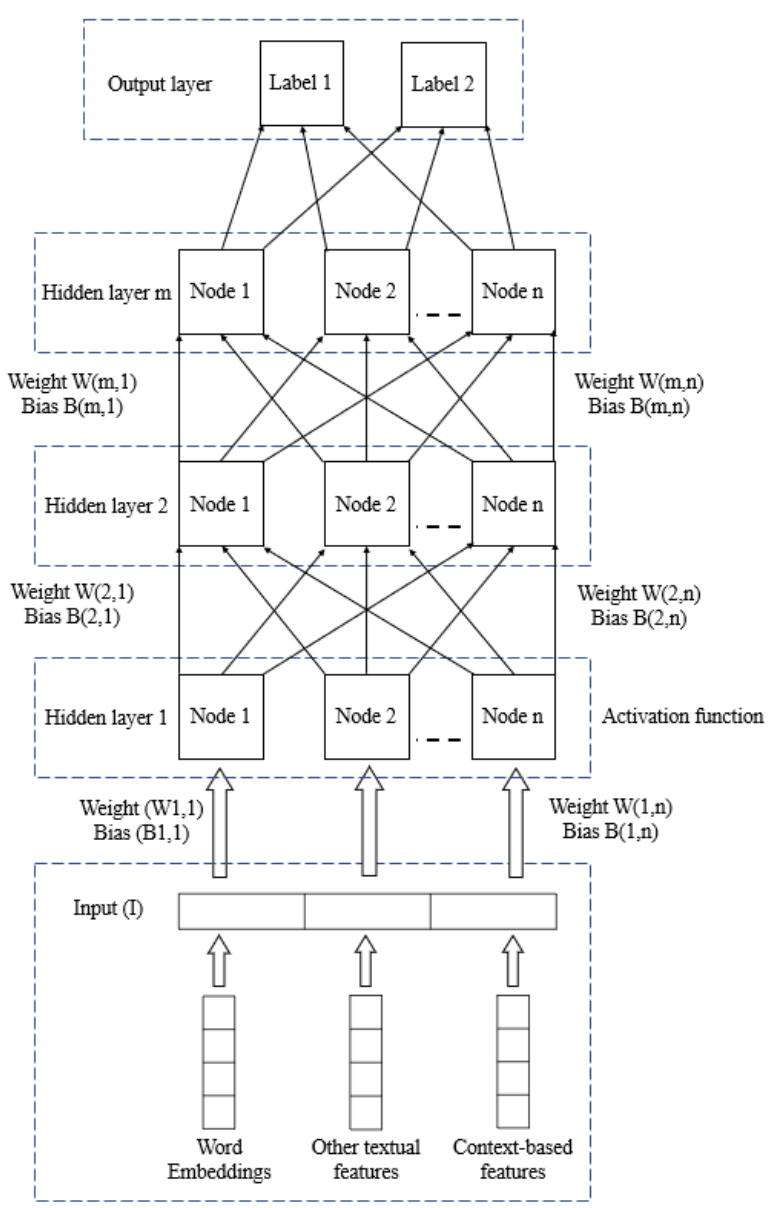

Figure 4.3: Fully connected feed forward neural network

In the connection between each layer, a bias and a weight is found, based on which the input to a node is transformed. Within each node, input data is run through so-called activation functions; non-linear functions, mapping input data to some value, usually using element-wise computations. Weights and biases are trained using the training data, seeking to find the optimal weight and bias values for each layer, to most accurately predict the desired outputs for the dataset. The updated input vector is calculated as so:

$$\hat{y} = weight \times y + bias \tag{4.7}$$

Training is performed based on feedback from prediction evaluation, as described 4.1.1. For a simple feed-forward neural network, this is generally done through the process of back-propagation, where the difference between predicted values and actual values, found using some loss function, is divided amongst the weights and biases of the network, starting

at the output layer and ending at the input layer. The distribution of loss among weight matrices in the network is based on the output of the node a given matrix is connected to, and the input vector that node has received from the rest of the neural network.

One of the key features granting deep learning approaches their merit is their ability to build simple representations of complex data, storing information regarding complex variation factors in simple weight matrices. This allows precise and fast analysis of data that is otherwise difficult to manage. This is done through the application of a deep architecture of layers, such as that shown in Figure 4.3, from which the name deep learning stems, each layer containing relatively simple computations and knowledge representations, but representing the complexity of the data through the interconnectivity of the layers. [Goodfellow et al., 2016]

4.2.1 Recurrent Neural Networks

What sets recurrent neural networks (RNNs) apart from many other deep learning approaches is their ability to take the sequentiality of texts into account. This is achieved by creating a unit within the RNN for each unit of text in a given input. This can be for each word, if the corpus consists of sentences, or for each sentence, if the corpus consists of whole documents. These units within the RNN will be denoted time units. Weights are shared across these time units, unlike for a feed forward neural network where weights differ from layer to layer, and the units of the RNN train the weights in unison. Each time unit has a hidden state which is unique to that time unit, and is applied to any input passed through the node. [Goodfellow et al., 2016] As can be seen from Figure 4.4, the output of each time step is a function of the hidden state of the time unit at that time-step, and the input and output weight matrices, denoted X and Y in the figure. The hidden state is in turn a function of the hidden state of the prior time-step, and the weight matrix Z. In terms of sentence processing, this means that the output and hidden state at each time step is a function of the words that have already passed through the RNN.

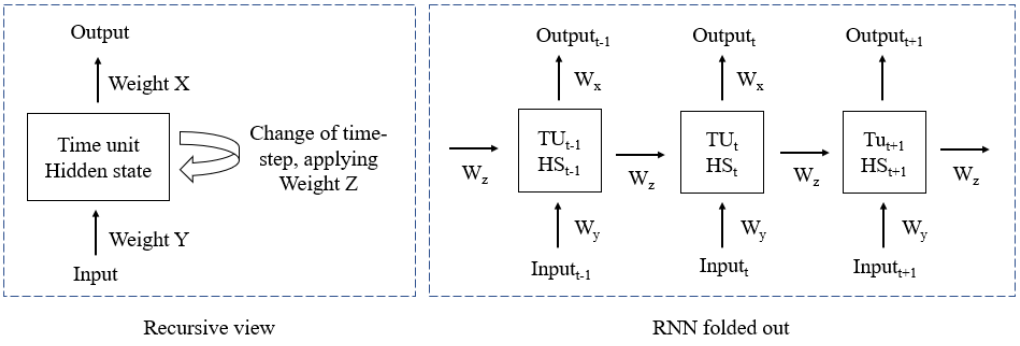

Figure 4.4: Basic RNN, TU representing time units and HS representing hidden states

It has been shown that feeding the inversed input into a second RNN, making its hidden state and output a function of the words in reverse reading order instead of normal reading order, and concatenating or averaging the final output of the two RNNs, can increase performance of the network for some stance detection tasks due to more contextualized representation [Yang et al., 2016, Li et al., 2017, Augenstein et al., 2016]. This approach is called bi-directionality.

In recent years, deep learning networks based on convolution (CNNs) [Kim, 2014, Chen et al., 2017] and recursion (RNNs) [Augenstein et al., 2016, Kochina et al., 2017, Li et al., 2017, Iyyer et al., 2014, Ma et al., 2018] have shown to be highly effective for classification tasks. Due to the popularity of recurrent networks for use in NLP classification tasks, one such model type will be used as primary model within this thesis, namely the LSTM expansion on the RNN model type.

4.2.2 LSTM architecture

All three models were built around a RNN model architecture, using an LSTM expansion [Hochreiter and Schmidhuber, 1997], where the time units of the RNN is exchanged for LSTM units. The core feature of LSTM models is their ability to circumvent the vanishing and exploding gradient problem, in which the gradient along which a unit is trained either converges on 0, or grows exponentially, and training stops for that unit, making learning a correlation between temporally distant events impossible. [Pascanu et al., 2013] This is achieved through the use of a gated architecture, designed to control access to memory states from earlier time units, avoiding full memory reads and writes. [Goldberg, 2017] To achieve this, LSTMs apply a combination of gate units and state updates, before allowing data to be parsed through the activation function of the unit, as visualized in Figure 4.5. By retaining parts of memory states, avoiding complete over-writes, LSTM models enable long-term dependencies between words, for instance letting a word passed to the model as the first in a sentence, add to the model's understanding of the last word in a sentence. An example of this is presented below, where a normal RNN would find learning plural dependency difficult, while this is easier for the LSTM.
*The **cats** living in ... **are** red*
*The **cat** living in ... **is** red*

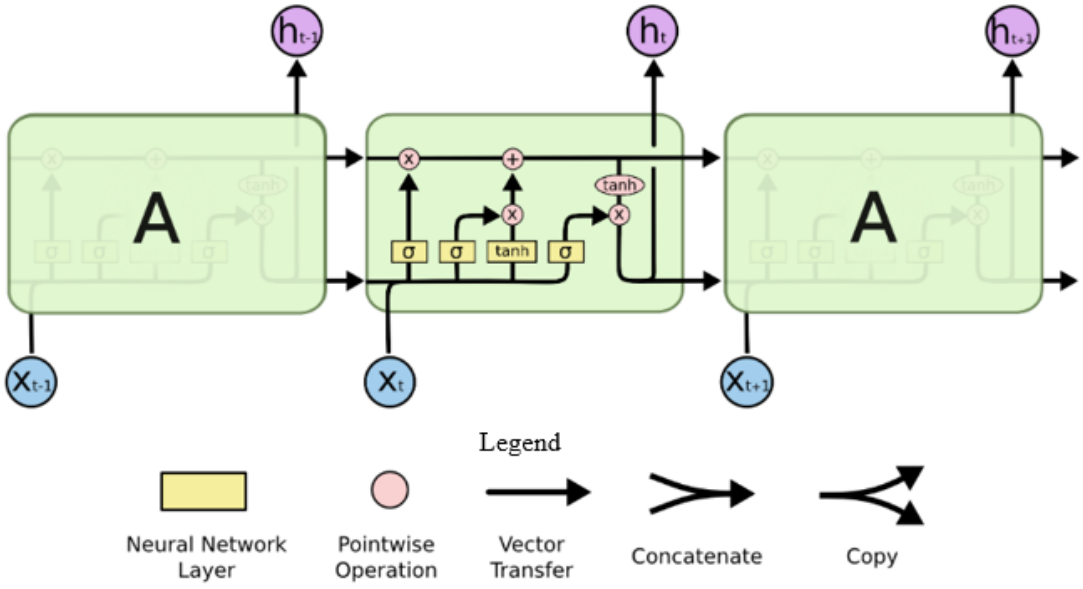

Figure 4.5: LSTM visualization [Olah, 2015]

Gates within an LSTM are composed of a combination of a non-linear activation function, determining which parts of a given vector to allow access to the memory state of the cell, a weight matrix, and a multiplicative pointwise operation, to include the parts of the vector which is allowed access in the memory state [Hochreiter and Schmidhuber, 1997]. When the model was first proposed in 1997, a sigmoid function was suggested as a strong activation function to be used in gates, and it is still the activation function of choice in recent implementations [Augenstein et al., 2016]. Part of training an LSTM, is training the output gate to realize, which error signals to allow access to the cell state, and training the input gate to realize, which errors to release, done by training aforementioned weight matrices within the gates. This way, the LSTM unit will become gradually better at retaining the correct information, while concurrently becoming better at learning new information.

The forget gate concatenates the output of the previous time unit, h_{t-1}, with the input vector x_t, multiplies the vector with a weight matrix and adds bias, finally running this through a sigmoid activation function. The output of the forget gate is calculated as so:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \tag{4.8}$$

The output f_t is added multiplicatively to the cell state.

The input gate first identifies the values which are to be updates using a sigmoid function over the concatenated h_{t-1} and x_t, and identifies potential candidate values using a tanh function over the same values, calculated as so:

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \tag{4.9}$$

$$\tilde{C}_t = tanh(W_c \times [h_{t-1}, x_t] + b_c) \tag{4.10}$$

The product of i_t and $\tilde{C}_t$ are added to the cell state. Including the forget and input gates, the cell state at timestep t is thus calculated as:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{4.11}$$

The output gate first determines which parts of the cell state will be output, calculated as:

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \tag{4.12}$$

The current cell state is then normalized to be between -1 and 1 using tanh, the resulting cell state being multiplied by o_t as so:

$$h_t = o_t \times tanh(C_t) \tag{4.13}$$

4.2.3 Neural Network terminology

Below, common terminology used in neural network theory is presented based on [Goldberg, 2017] and [Goodfellow et al., 2016], to establish a common ground for the discussions in Section 4.3 and Chapters 5 and 6.

Loss functions are used for defining the difference between a goal value y and a predicted value $\hat{y}$. The difference between y and $\hat{y}$ is defined as the loss for the current prediction.

Overfitting refers to errors stemming from a model learning training data too closely, mirroring the noise in the training data, making it unable to generalize, thus reducing its ability to accurately classify data in the test set.

Regularization works by applying penalties to weight matrices, which develop high weights, thus forcing the neural network to rely less on these weight matrices, thus helping prevent overfitting. Within this paper L2 regularization is applied, which functions by adding an extra term to the loss function. This term consists of the summed squares of all weights, w, scaled by some regularization value, λ. For a test set of size n, the L2 regularization term is calculated as:

$$\frac{\lambda}{2n}\sum_{w} w^2 \tag{4.14}$$

Dropout is a regularization tool which serves to improve generalization and prevent overfitting. This is achieved by randomly ignoring nodes within the neural network during classification and training, setting their values to 0 for a single iteration, thus forcing the network to not rely on any single node to achieve good classification scores. A visualization of a simple neural network applying dropout can be found in Figure 4.6.

Batching consists of grouping input data, calculating a collected loss for the whole batch, and returning that loss as feedback to the neural network, rather than returning loss for each datapoint. This can help combat overfitting, as the effect of noise in a single datapoint becomes less influential, as it is spread out over a whole batch.

Rectified Linear Units (ReLU) are activation functions that increase sparsity of the data, thus decreasing training time, generally by mapping negative values in a vector to 0 by finding $max(0, value)$. For the remainder of this paper, fully connected linear neural network layers applying a ReLU activation function will be referred to as ReLU layers. Neural networks using aforementioned ReLU layers have been shown to achieve better convergence performance than those applying sigmoid functions, which has historically been the activation function of choice [Krizhevsky et al., 2012].

Dimensionality of a neural network is defined by its width and length. Width refers to the number of units, or nodes, within a layer, acting in parallel, each computing their own activation value, whereas length refers to the number of layers within the network.

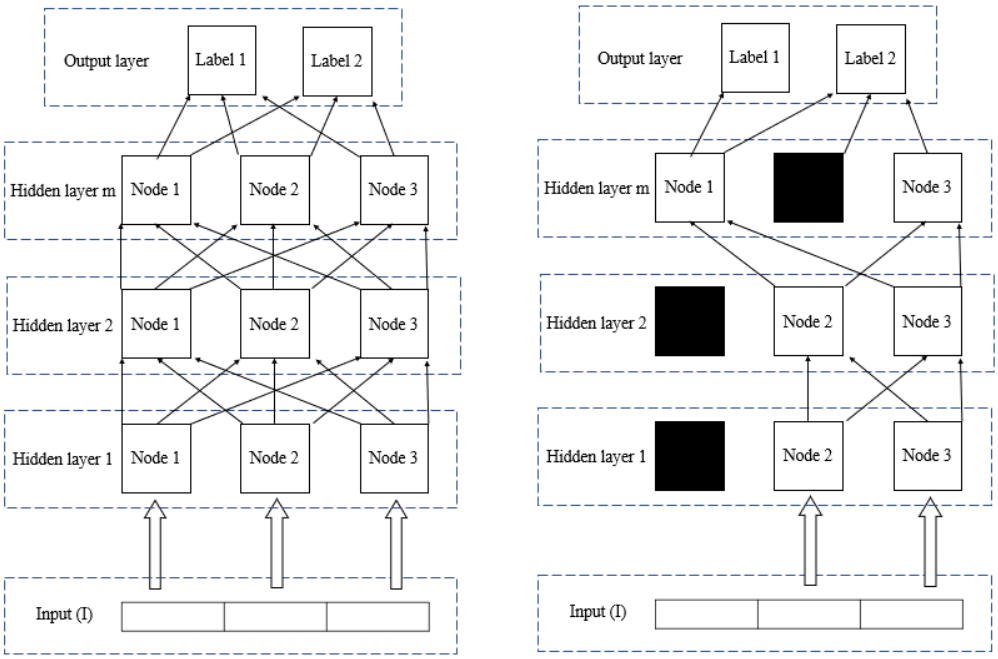

Figure 4.6: Neural network without dropout (left) and with dropout (right)

Softmax is a function which normalizes values in an input vector to the interval (0-1), values summing to 1, thus representing a probability distribution. This is often used as the final layer of neural networks as the basis for classification, as the most likely label can be picked based on which label has the highest probability.

Optimizer algorithms within the context of deep learning, are built to find the optimal training pattern for a neural network, based on some evaluation measure. The most popular current optimizers are all based on gradient descent algorithms.

Learning rate defines the change that parameters within a neural network take, when loss is back-propagated. If a learning rate is too large, it might miss a maxima, while if it is too small, it might not be able to converge on a maxima within a reasonable time-frame.

4.3 Model design

This section presents design choices made in regards to the three models created for the task of detecting stance in quotes from the generated dataset presented in Chapter 3.

4.3.1 Choice of loss function

This section presents two common loss functions used within ML, as presented by [Goldberg, 2017], and the choice of loss function for the three created models.

Hinge loss for binary classification uses a single scalar $\tilde{y}$ and the intended output as either -1 or 1, where the classification is found as $\hat{y} = sign(\tilde{y})$. If y and $\tilde{y}$ have the same sign, a classification is found to be correct, that is, if $y \times \tilde{y} > 0$. Thus, binary hinge loss can be defined as:

$$L_{hinge(binary)}(\tilde{y}, y) = max(0, 1 - y \times \tilde{y}) \tag{4.15}$$

To expand hinge loss to multi-class prediction, classification is performed by $max(\hat{y})$, where $\hat{y}$ is a class probability distribution. Denoting t as the correct class and k as the class with highest probability, where $t \neq k$, multi-class hinge loss is calculated as:

$$L_{hinge(multi-class)}(\hat{y}, y) = max(0, 1 - (\hat{y}_t - \hat{y}_k)) \tag{4.16}$$

Thus, the loss is defined as the difference between the probability of the correct class and the most probable class which is not the correct class, given a margin of at least 1.

Due to the required margin of 1, hinge loss penalizes not only incorrect predictions, but also predictions that are not confident. Nevertheless, if prediction confidence exceeds the given margin, no loss will be generated. Thus, the loss functions focuses on achieving a high level of accuracy, while ignoring the estimation of exact class probabilities.

Cross-entropy based functions use the cross entropy between a given prediction probability distribution $\hat{y}$ and the actual class probability distribution y as basis for loss calculations. For multi-class prediction using categorical cross-entropy, cross entropy across all labels is calculated as:

$$L_{cross-entropy}(\hat{y}, y) = -\sum_i y_i log(\hat{y}_i) \tag{4.17}$$

For cases where training tuples have only a single correct class assignment, such as the task defined within this paper of classifying stance within quotes, y is presented as a one-hot vector with a flag for the correct class, and loss is calculated as:

$$L_{cross-entropy(hardclassification)}(\hat{y}, y) = -log(\hat{y}_t) \tag{4.18}$$

Where t denotes the correct class. Thus, this loss function seeks to reach an assignment of 100 % probability to the correct class t.

Cross-entropy loss is widely applied within stance detection [Iyyer et al., 2014, Kochina et al., 2017, Augenstein et al., 2016, Li et al., 2017, Yang et al., 2016, Zarrella and Marsh, 2016], and is therefore used as the loss function for model implementations in this paper.

4.3.2 Choice of optimizer

This section presents two common optimizers based on gradient descent used within deep learning, as presented by [Goldberg, 2017]. Gradient descent works by minimizing some loss function $f(x)$, through following the derivative of the function $f'(x)$, identifying minima where $f'(x) = 0$.

Batch gradient descent is the most simple implementation of gradient descent, and optimizes model parameters for each epoch. This is computationally efficient, and reduces the risk of learning noisy data from single training tuples, but increases the risk of getting stuck in local minima, and furthermore requires the full dataset to be held in memory.

Stochastic gradient descent (SGD) is the more common of the two, and optimizes model parameters for each training tuple rather than for a whole epoch or batch. The increased number of model updates generally leads to slower runtime, and noisy training tuples might make the loss function converge in the wrong direction, but generally SGD has been found to quickly be able to find low values for a given cost function.

A wide variety of adaptations of SGD have been developed, including but far from limited to Adagrad which implements diminishing learning rate for parameters which get updated frequently [Duchi et al., 2011], Adadelta which bases diminishing learning rates on a moving window of gradient updates [Zeiler, 2012] and ADAM which bases the diminishing learning rate on both which parameters are frequently updated and a decaying average of prior gradients [Kingma and Ba, 2014].

For this paper, a simple SGD implementation is used in the form of PyTorch's SGD optimizer class.

4.3.3 Full model architecture

All three of the implemented models are designed to have variable dimensionality, including both width and length, where the number of each layer type, and width of each layer, is defined by the user upon model creation. A single forward pass of a model includes passing the data through one or more fully connected LSTM layers, then through one or more ReLU layers, a dropout layer and finally a softmax layer, allowing for classification.

Applied features

As described within Section 2.2.1, word embeddings retain semantic information regarding words, whereas one-hot vectors only retain the presence of the word. In current research within stance detection, the most common approach to word representation is the use of word embeddings [Mohammad et al., 2017, Zeng et al., 2016, Kim, 2014, Iyyer et al., 2014, Ma et al., 2018, Li et al., 2017, Skeppstedt et al., 2017], and therefore word embeddings are used for word representation within this project. A number of resources were considered for implementing word embeddings in Danish, including using the Polyglot Python library, fastText word embeddings based on Facebook data and the embeddings generated from the Dasem project [Årup Nielsen, 2019]. In this decision, a major role was played by the size of the embedding libraries, as RAM resources for the research project were limited. Ultimately, fastText embeddings [Grave et al., 2018] in basic text vector format were used, from which a subset of the embeddings was created based on a vocabulary generated from the quote dataset, to allow faster loading and easier handling of the embedding library.

Quote embeddings are generated as a matrix of word embeddings of the size $E \times L$, E denoting the word embedding size, 300 for the fastText embeddings, L denoting the

length of the quote. For any value x_i in the quote embedding matrix, it is true that $x_i \in R| -1 \leq x_i \leq 1$.

Average quote embeddings are generated as a vectors, where the value of the vector is the average of all word embeddings in the quote. For a quote of length N, average quote embeddings are calculated as:

$$x_i = \frac{\sum_{i=0}^{N} x_{word}}{N} \tag{4.19}$$

Thus the vector will be of length 300, when using the fastText word embeddings.

Politician and party vectors are generated using two mappings, one for politicians and one for parties. For a given quote, a one-hot politician vector is generated, with a flag representing the index of the politician behind the quote in the politician mapping, and likewise a one-hot party vector is generated with a flag representing the party index of the party of the politician behind the quote. These are concatenated to form the full feature vector. This vector is the only context-based feature applied within this project.

4.3.4 Conditional LSTM

The conditional LSTM implementation differs from the others in the way that it handles context-based features. The model is made conditional by initializing the LSTM layers at time-step t_0 passing a party and politician feature vector, zero padded to achieve a length of 300, mirroring that of the word embeddings. Thus the model learns politician and party-dependent quote representation. The inspiration of this model is derived from [Augenstein et al., 2016]'s entrance in the SemEval 2016 competition for Task 6, in which a Bi-Directional Conditional LSTM is set to a political stance detection task. For a quote $Q = \{word_1, word_2, ...word_n\}$, quote embeddings are generated as described in Section 4.3.3, and at each time-step the pre-trained fastText word embedding of the corresponding word is used as input. A visualization of this model can be found in Figure 4.7.

4.3.5 Quote LSTM

The Quote LSTM takes as input an average quote embedding concatenated with the corresponding politician and party vector, both generated as described in Section 4.3.3. Thus, the model does not utilize the full extent of the LSTM properties, as a full quote is used as input rather than a single word at each time step, but for the same reasons, training and performing predictions is significantly faster than when using the Conditional LSTM. This architecture is found depicted in Figure 4.8

4.3.6 Bi-directional Quote LSTM design

The Bi-directional Quote LSTM is an extension to the Quote LSTM, and is, like the Conditional LSTM, inspired by [Augenstein et al., 2016]'s entry into the SemEval 2016 Task 6 competition. Feature vectors are generated in the same way as for the Quote LSTM, but a copy of the feature vectors are created and inversed, so the last word of a

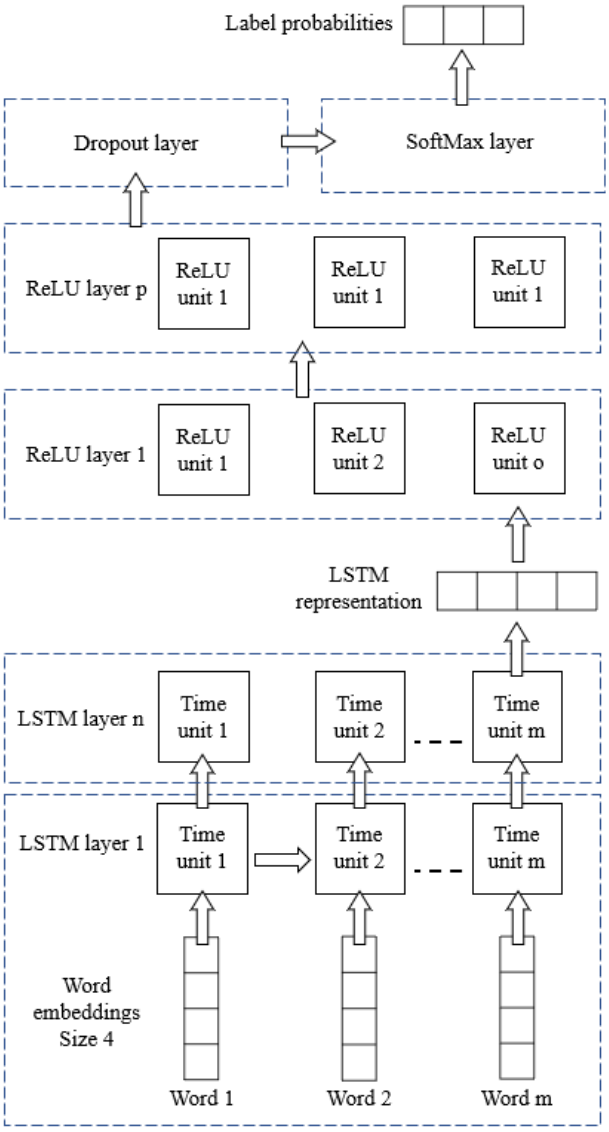

Figure 4.7: Visualization of Conditional LSTM

given quote will occur first. Bi-directionality is achieved as described in 4.2.1, by creating a second LSTM, feeding the second set of feature vectors through that, and concatenating the output vectors of the two LSTMs, after which the resulting vector is then passed to the ReLU layers, and on through the rest of the network. The two LSTMs mirror each other in dimensionality, having the same amount of dimensions and layers. A visualization of the Bi-directional Quote LSTM is found in Figure 4.9.

4.4 Addressing the class imbalance problem

As discussed in Section 3.7, a significant class imbalance can be observed within the data towards the *for* label, with *against* and *neutral* being under-represented. Several approaches are available within the field of ML for tackling this issue, and here is first presented the approaches applied in this paper, followed by a possible extension that might improve class imbalance handling further.

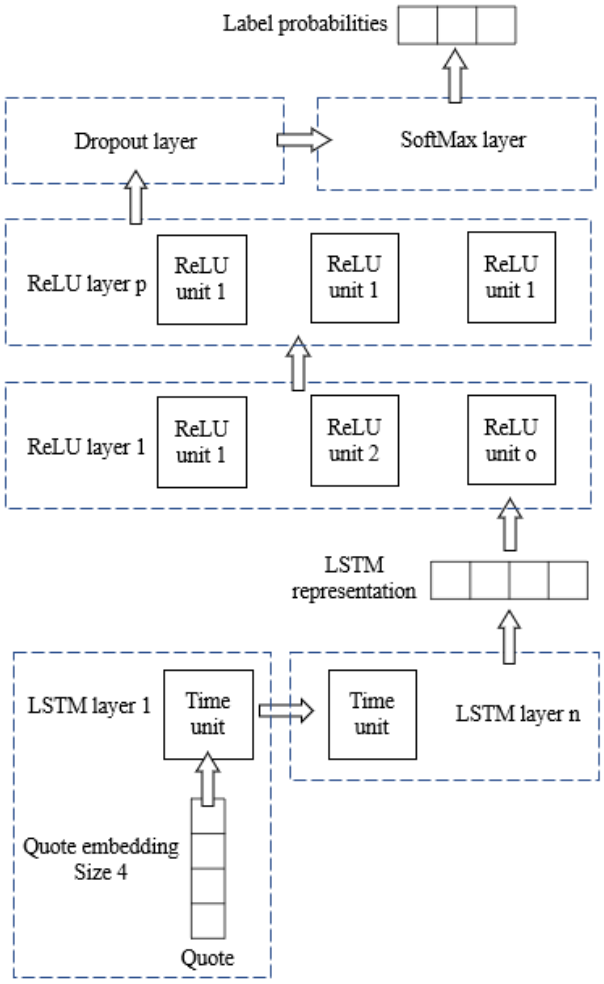

Figure 4.8: Visualization of Quote LSTM

A class imbalance is likely to lead to models training on the dataset overfitting their neural net towards that class. As a result of this, tools classically applied to prevent overfitting can also, to some extent, help solve issues of class imbalance. For this reason, $F1_{marco}$ is applied as primary evaluation measure for the models rather than $F1_{micro}$, as this prevents over-representation of classes from being reflected in the scores, as discussed in Section 4.1.2. For the same reasons, and to avoid overfitting during training, L2 regularization and dropout are applied to the models.

A well documented approach to solving issues of class imbalance is the implementation of synthetic minority over-sampling (SMOTE) [Tomanek and Hahn, 2009]. During a batching or sampling process, copies are created of data points from minority classes, thus making them more frequent in the batch or sample, and incentivizing a model to learn the pattern of minority classes to minimize loss [Tomanek and Hahn, 2009]. However, by using this approach researchers risk coaxing their models into performing underfitting, that is, neglecting to focus on training patterns that are significant to the data, by forcing the model to focus on minority classes. This is avoided by watching results of the over-sampling closely, narrowing in on the correct level of over-sampling where the model learns the patterns of minority classes, while not under-fitting majority classes.

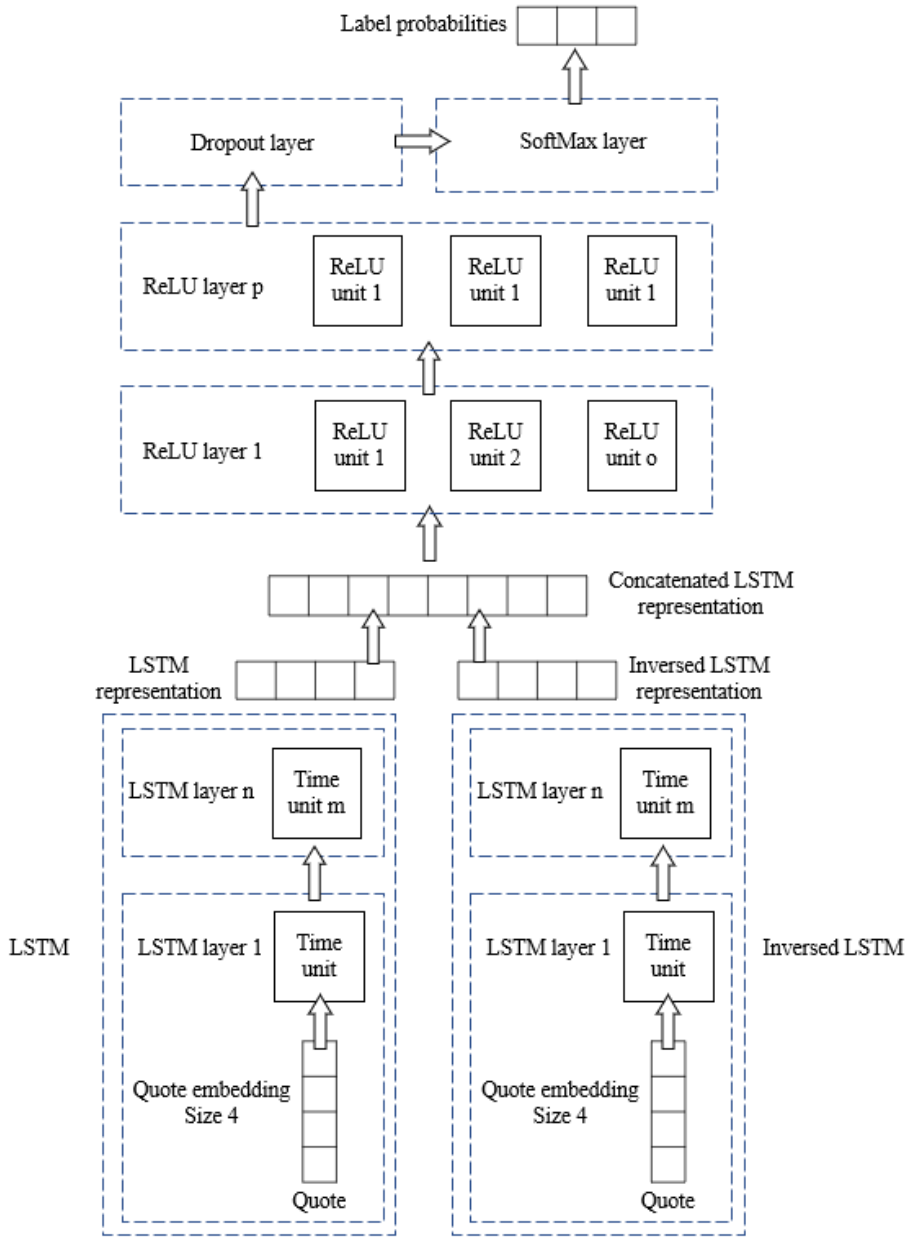

Figure 4.9: Visualization of Bi-Directional Quote LSTM

Chapter 5

Experiments

This chapter presents the experimental setup applied in optimizing and testing the three models for which designs are described in Chapter 4.3. Section 5.1 describes the process of specifying and searching hyperparameter spaces for the three models, followed by a description of the primary experimental setup for the comparative of the three models in 5.2. Finally, experimental setups for further hyperparameter experiments as well as experiments regarding the use of different feature types are described in Section 5.3.

5.1 Hyperparameter search

In search of the optimal hyperparameters for each model, three search spaces were defined, as presented in Table 5.1. The theoretical space is the full search space of all hyperparameters that might influence the results of the models. A benchmarking setup was built, performing grid-search of a given hyperparameter space to a given max epoch, saving benchmarks at that max epoch and at each point on a list of given epoch counts. Running a single hyperparameter combination, with a max epoch of 500, using the Quote LSTM, which is the fastest of the three LSTM implementations, takes 30 minutes. Given the 3888 parameter combinations of the theoretical hyperparameter search space, this leads to a total runtime of 81 days to search this hyperparameter space, making the theoretical search space non-viable. Reducing the max epoch value to 300 yields a runtime of 18 minutes for benchmarking a hyperparameter combination. Removing dropout and learning rate from the hyperparameter space, as well as the largest value for L2 Regularization, yields a total combination count of 243, and thus a total runtime of the benchmarking of slightly over 3 days, making the space viable for identifying optimal hyperparameters for the Quote LSTM. The remaining parameter values are defined as the initial search space, which is used for hyperparameter search for Quote LSTM, while leaving the effects of dropout and learning rate to be determined in subsequent experiments.

Performing hyperparameter search for the Conditional LSTM proved to be a challenge due to the training time of the system. Running a single hyperparameter combination, with a max epoch value of 300, takes 101 minutes, resulting in a runtime of 17 days to search the initial space. Seeing as this would not be viable within the time limits of the thesis, a reduced hyperparameter space was created, which yielded a total runtime of just below 4 days, consisting of 54 hyperparameter combinations, instead of the 243 of the initial space.

	Hyperparameter spaces		
Hyperparameters	Theoretical space	Initial space	Reduced space
LSTM Layers	[1, 2, 3]	[1, 2, 3]	[1]
LSTM Units	[50, 100, 200]	[50, 100, 200]	[50, 100, 200]
ReLU Layers	[1, 2, 3]	[1, 2, 3]	[1, 2]
ReLU Units	[50, 100, 200]	[50, 100, 200]	[50, 100, 200]
L2 Regularization	[0.0, 0.0001, 0.0003, 0.0005]	[0.0, 0.0001, 0.0003]	[0.0, 0.0001, 0.0003]
Dropout	[0.0, 0.2, 0.5, 0.7]	[0.5]	[0.5]
Learning Rate	[0.001, 0.01, 0.1]	[0.001]	[0.001]
Epochs	[30, 50, 70, 100, 200, 300, 500]	[30, 50, 70, 100, 200, 300]	[30, 50, 70, 100, 200, 300]
Total combinations (excluding epochs)	3888	243	54

Table 5.1: Overview of hyperparameter spaces applied in hyperparameter search

The reduced hyperparameter space was based on the results of the hyperparameter search for Quote LSTM. This showed that none of the top 100 hyperparameter combinations, when compared on $F1_{macro}$, contained more than one LSTM layer or 2 ReLU layers, leading to those hyperparameter values being discarded.

Like for the Conditional LSTM, optimal hyperparameters for the Bi-Directional Quote LSTM was found through a search of the reduced search space, seeing as the Bi-Directional Quote LSTM resembles the Quote LSTM, and it is therefore assumed that the two models have similar optimal hyperparameter values.

5.1.1 Results

In Table 5.2 the five most successful hyperparameter combinations for each of the three models are presented, evaluated on $F1_{macro}$. Conditional LSTM seems to have some specific parameter values which perform well, namely a number of LSTM dimensions of 100, 2 ReLU layer and 200 ReLU dimensions. This differs from the Quote LSTM and Bi-Directional Quote LSTM, which both show good results for LSTM dimensions of 50, 100 and 200, one or two ReLU layers and between 50 and 200 ReLU dimensions. It is worth noting that a number of LSTM layers of [1, 2, 3] were included in the Quote LSTM search space, but good results were only observed when using a single layer. Only a single LSTM layer was included in the Conditional LSTM and Bi-Directional LSTM search spaces, making the fact that all five top hyperparameter combinations for both models contain this parameter insignificant.

5.1.2 Search alternatives

The applied grid search approach is the most thorough approach to hyperparameter search, as it systematically covers all hyperparameter combinations. However, a number of alternative approaches have lower runtimes, and might thus have been applied to search a wider parameter space for the Conditional LSTM and Bi-Directional Quote LSTM.

Conditional LSTM

	LSTM		ReLU			F1		Class Accuracy		
Epochs	Layers	Dims	Layers	Dims	L2	Micro	Macro	For	Against	Neutral
200	1	100	2	200	0	0.400	**0.375**	0.580	0.244	0.440
200	1	100	2	200	0.0003	0.400	0.366	0.493	0.361	0.280
200	1	100	2	200	0.0003	**0.411**	0.362	0.372	0.200	0.536
200	1	100	2	200	0	0.400	0.357	0.302	0.240	0.580
300	1	100	2	50	0.0003	0.400	0.354	0.681	0.198	0.320

Quote LSTM

	LSTM		ReLU			F1		Class Accuracy		
Epochs	Layers	Dims	Layers	Dims	L2	Micro	Macro	For	Against	Neutral
300	1	50	1	100	0	0.717	**0.575**	0.826	0.120	0.797
300	1	200	1	100	0	0.739	0.553	0.942	0.040	0.739
300	1	200	1	200	0.0001	0.739	0.553	0.930	0.040	0.754
300	1	100	2	50	0	0.694	0.544	0.826	0.080	0.754
200	1	200	1	100	0.0001	**0.756**	0.541	0.977	0.000	0.754

Bi-Directional Quote LSTM

	LSTM		ReLU			F1		Class Accuracy		
Epochs	Layers	Dims	Layers	Dims	L2	Micro	Macro	For	Against	Neutral
300	1	100	2	50	0.0001	0.706	**0.551**	0.826	0.080	0.783
200	1	200	1	200	0	**0.750**	0.538	0.942	0.000	0.783
200	1	50	1	50	0	**0.750**	0.537	0.965	0.000	0.754
300	1	200	2	50	0.0001	0.622	0.535	0.721	0.240	0.638
200	1	50	2	200	0.0001	0.683	0.534	0.802	0.080	0.754

Table 5.2: Overview of results of hyperparameter search for Conditional LSTM, Quote LSTM and Bi-Directional Quote LSTM.

Random search chooses parameters from a search space at random, and runs for a specified number of iterations. This can be applied to give an idea of, which hyperparameters have a significant effect on performance, thus allowing the creation of a smaller hyperparameter space based on a broad array of tests.

Gradient-based search calculates a gradient for hyperparameters, and optimize these based on gradient descent. This approach is reliant on the hyperparameter space being convex, and functions best if gradients are smooth. Seeing as a number of the hyperparameters used in this paper are discrete values, e.g. number of layers within the network, gradients will not be smooth, and gradient-based hyperparameter search might not perform optimally if applying a gradient-based approach.

5.2 Primary experimental setup

Evaluation of the three models was performed with the full dataset, as well as with the *national policy* subset, using the optimal hyperparameters found through the hyperparameter search process described in Section 5.1. No tests were made using only the *centralization*

dataset, as this was deemed too small at a quote count of just 104. The models were compared on both $F1_{micro}$ and $F1_{macro}$, calculated as defined in Section 4.1.2, and confusion matrices were created for each of the two subsets, to allow analysis of the strength and weaknesses of each model, in terms of which classes are correctly and incorrectly classified. Seeing as the dataset was generated specifically for this research project, there exists no prior benchmarks with which to compare the models. For this reason, two benchmark models are built, namely a Gaussian Naive Bayes classifier and Random Forest classifier, both out-of-the-box implementations from the scikit-learn Python library.

5.2.1 Results

From Table 5.3 it can be observed, that the Quote LSTM outperforms all five other models in terms of $F1_{macro}$ on both the full and *national policy* datasets. The Quote LSTM also performs best in regards to $F1_{micro}$ on the full dataset, but is outperformed by the Bi-Directional Quote LSTM on this measure, on the *national policy* dataset.

Full Dataset

	GNB	RF	$\text{LSTM}_{\text{cond}}$	$\text{LSTM}_{\text{quote}}$	LSTM_{bi}
F1_{macro}	0.266	0.387	0.375	**0.575**	0.551
F1_{micro}	0.306	0.461	0.400	**0.717**	0.706
F_{acc}	0.442	0.267	0.580	**0.826**	**0.826**
A_{acc}	**0.6**	0.2	0.244	0.120	0.080
N_{acc}	0.029	**0.797**	0.440	**0.797**	**0.797**

National Policy dataset

	GNB	RF	$\text{LSTM}_{\text{cond}}$	$\text{LSTM}_{\text{quote}}$	LSTM_{bi}
F1_{macro}	0.254	0.435	0.358	**0.585**	0.562
F1_{micro}	0.283	0.560	0.372	0.774	**0.779**
F_{acc}	0.337	0.525	0.256	**0.963**	**0.963**
A_{acc}	0.696	0.435	**0.480**	0.043	0
N_{acc}	0.036	0.821	0.478	0.804	**0.839**

Table 5.3: Performance comparison of all models, including benchmark models, using optimized hyperparameters, GNB referring to Gaussian Naïve Bayes, RF referring to Random Forest, $\text{LSTM}_{\text{cond}}$ referring to Conditional LSTM, $\text{LSTM}_{\text{quote}}$ referring to quote LSTM and LSTM_{bi} referring to Bi-Directional Quote LSTM

Misclassification analysis

In Table 5.4 the confusion matrices for each model, run with optimal hyperparameters on both the full and *national policy* dataset, can be found. It should be noted that the confusion matrices are based on a separate run-through of the models than the hyperparameter search, as confusion matrices were not saved during hyperparameter search. Therefore, calculations performed using the classifications within the confusion matrices are highly likely to differ from the results presented in Sections 5.2, 5.3, 5.5, 5.6 and 5.7.

Observing Table 5.4, no general difference in classification performance can be observed across all models, when comparing confusion matrices for the full dataset with those for the *national policy* dataset.

Gaussian Naïve Bayes

Policy Dataset

	F	A	N
F	38	47	1
A	8	15	2
N	10	57	2

Full Dataset

	F	A	N
F	27	52	1
A	5	16	2
N	2	52	2

Quote LSTM

Policy Dataset

	F	A	N
F	77	0	3
A	19	1	3
N	7	4	45

Full Dataset

	F	A	N
F	81	0	2
A	18	0	7
N	21	0	48

Random Forest

Policy Dataset

	F	A	N
F	42	7	31
A	6	1	16
N	5	5	46

Full Dataset

	F	A	N
F	23	12	51
A	5	5	15
N	4	10	55

Bi-Directional Quote LSTM

Policy Dataset

	F	A	N
F	77	0	3
A	19	0	4
N	9	0	47

Full Dataset

	F	A	N
F	79	0	9
A	19	0	6
N	16	0	53

Conditional LSTM

Policy Dataset

	F	A	N
F	22	25	39
A	3	12	10
N	6	30	33

Full Dataset

	F	A	N
F	20	23	43
A	4	4	17
N	7	20	42

Table 5.4: Confusion matrices for each classification model, using optimized hyperparameters

5.3 Secondary experiments

All additional hyperparameter experiments were performed using the Quote LSTM model, with the optimal hyperparameter combination found through hyperparameter search as described in Section 5.1. That is 1 LSTM layer, 50 LSTM dimensions, 1 ReLU layer, 100 ReLU dimensions, no L2 regularization and 300 epochs.

5.3.1 Additional hyperparameters

Experiments with varying dropout values, results of which are presented in Table 5.5, show that the initial dropout value of 0.5 performs better in terms of $F1_{micro}$ and $F1_{macro}$ than the three alternative values of 0.2, 0.7 and no dropout.

Observing results of the experiments on learning rate presented in Table 5.6, it can be observed that the initially applied value of 0.001 achieves the best results overall in terms of both $F1_{micro}$ and $F1_{macro}$, outperforming the alternatives of 0.005, 0.01 and 0.1.

Dropout = 0.0

Epochs	$\mathbf{F1_{micro}}$	$\mathbf{F1_{macro}}$	$\mathbf{F_{acc}}$	$\mathbf{A_{acc}}$	$\mathbf{N_{acc}}$
30	0.383	0.185	0.000	0.000	1.000
50	0.383	0.185	0.000	0.000	1.000
70	0.383	0.185	0.000	0.000	1.000
100	0.639	0.457	0.605	0.000	0.913
200	**0.722**	**0.517**	0.930	0.000	0.725
300	0.589	0.465	0.674	0.080	0.667

Dropout = 0.2

Epochs	$\mathbf{F1_{micro}}$	$\mathbf{F1_{macro}}$	$\mathbf{F_{acc}}$	$\mathbf{A_{acc}}$	$\mathbf{N_{acc}}$
30	0.383	0.185	0.000	0.000	1.000
50	0.383	0.185	0.000	0.000	1.000
70	0.383	0.185	0.000	0.000	1.000
100	0.567	0.403	0.465	0.000	0.899
200	**0.700**	**0.495**	0.977	0.000	0.609
300	0.589	0.394	0.977	0.000	0.319

Dropout = 0.5

Epochs	$\mathbf{F1_{micro}}$	$\mathbf{F1_{macro}}$	$\mathbf{F_{acc}}$	$\mathbf{A_{acc}}$	$\mathbf{N_{acc}}$
30	0.383	0.185	0.000	0.000	1.000
50	0.383	0.185	0.000	0.000	1.000
70	0.383	0.185	0.000	0.000	1.000
100	0.506	0.348	0.314	0.000	0.928
200	**0.733**	0.525	0.954	0.000	0.725
300	0.717	**0.575**	0.826	0.120	0.797

Dropout = 0.7

Epochs	$\mathbf{F1_{micro}}$	$\mathbf{F1_{macro}}$	$\mathbf{F_{acc}}$	$\mathbf{A_{acc}}$	$\mathbf{N_{acc}}$
30	0.383	0.185	0.000	0.000	1.000
50	0.383	0.185	0.000	0.000	1.000
70	0.383	0.185	0.000	0.000	1.000
100	0.378	0.183	0.000	0.000	0.986
200	**0.661**	**0.468**	0.907	0.000	0.594
300	0.578	0.375	0.977	0.000	0.290

Table 5.5: Results of dropout experiments on Quote LSTM using optimal hyperparameters

Learning Rate = 0.001

Epoch	$F1_{Micro}$	$F1_{Macro}$	F_{acc}	A_{acc}	N_{acc}
30	0.383	0.185	0.000	0.000	1.000
50	0.383	0.185	0.000	0.000	1.000
70	0.383	0.185	0.000	0.000	1.000
100	0.506	0.348	0.314	0.000	0.928
200	**0.733**	0.525	0.954	0.000	0.725
300	0.717	**0.575**	0.826	0.120	0.797

Learning Rate = 0.005

Epoch	$F1_{Micro}$	$F1_{Macro}$	F_{acc}	A_{acc}	N_{acc}
30	**0.728**	0.521	0.907	0.000	0.768
50	0.717	**0.538**	0.919	0.040	0.710
70	0.633	0.450	0.930	0.000	0.493
100	0.622	0.443	0.895	0.000	0.507
200	0.572	0.460	0.756	0.120	0.507
300	0.528	0.433	0.616	0.120	0.565

Learning Rate = 0.01

Epoch	$F1_{Micro}$	$F1_{Macro}$	F_{acc}	A_{acc}	N_{acc}
30	**0.733**	**0.526**	0.965	0.000	0.710
50	0.606	0.410	0.965	0.000	0.377
70	0.594	0.396	0.965	0.000	0.348
100	0.578	0.454	0.744	0.080	0.551
200	0.633	0.478	0.698	0.040	0.768
300	0.500	0.410	0.616	0.120	0.493

Learning Rate = 0.1

Epoch	$F1_{Micro}$	$F1_{Macro}$	F_{acc}	A_{acc}	N_{acc}
30	0.589	0.423	0.547	0.000	0.855
50	**0.706**	**0.504**	0.930	0.000	0.681
70	0.617	0.442	0.709	0.000	0.725
100	0.639	0.437	0.988	0.000	0.435
200	0.572	0.397	0.907	0.000	0.362
300	0.667	0.497	0.872	0.040	0.638

Table 5.6: Results of Learning Rate experiments on Quote LSTM using optimal hyperparameters

5.3.2 The effect of context-based features

Comparing Table 5.7 and Table 5.2, it can be observed that removal of either party or politician from the context-based features significantly reduces the Quote LSTM's resutls, while removal of both makes the model label all quotes as *for*.

No politician vector

Epochs	$\mathbf{F1_{micro}}$	$\mathbf{F1_{macro}}$	$\mathbf{F_{acc}}$	$\mathbf{A_{acc}}$	$\mathbf{N_{acc}}$
30	0.261	0.138	1.000	0.000	0.000
50	0.261	0.138	1.000	0.000	0.000
70	0.261	0.138	1.000	0.000	0.000
100	0.261	0.138	1.000	0.000	0.000
200	**0.594**	0.441	0.681	0.904	0.000
300	0.589	**0.441**	0.681	0.892	0.000

No party vector

Epochs	$\mathbf{F1_{micro}}$	$\mathbf{F1_{macro}}$	$\mathbf{F_{acc}}$	$\mathbf{A_{acc}}$	$\mathbf{N_{acc}}$
30	0.261	0.138	1.000	0.000	0.000
50	0.261	0.138	1.000	0.000	0.000
70	0.261	0.138	1.000	0.000	0.000
100	0.261	0.138	1.000	0.000	0.000
200	0.511	0.394	0.809	0.651	0.000
300	**0.522**	**0.405**	0.787	0.687	0.000

No contextual features

Epochs	$\mathbf{F1_{micro}}$	$\mathbf{F1_{macro}}$	$\mathbf{F_{acc}}$	$\mathbf{A_{acc}}$	$\mathbf{N_{acc}}$
30	0.261	0.138	1.000	0.000	0.000
50	0.261	0.138	1.000	0.000	0.000
70	0.261	0.138	1.000	0.000	0.000
100	0.261	0.138	1.000	0.000	0.000
200	0.261	0.138	1.000	0.000	0.000
300	0.261	0.138	1.000	0.000	0.000

Table 5.7: Results of experiments on Quote LSTM with reduced contextual features

Chapter 6

Analysis

This chapter presents the results of experiments on the models described in Chapter 5, first through a comparison of model performance in Section 6.1.1, based on the performance measures presented in Section 4.1.2, followed by an analysis of the classification capabilities of each model for each class in Section 6.1.2. Section 6.1.3 presents an analysis of the effect of tweaking dropout and learning rate for the Quote LSTM, and the effect of contextual features are analysed in Section 6.1.4. Section 6.1.5 presents an error analysis, containing observations regarding shortcomings of the three models, and considerations regarding how these shortcomings might be eliminated. This is followed by an analysis of the data distribution of the dataset in Section 6.2, both within party boundaries and between the parties in the dataset.

6.1 Model experiments

6.1.1 Performance comparison

Looking at Table 5.3 it can be observed, that the two LSTM models based on average quote embeddings, Quote LSTM and Bi-Directional Quote LSTM, out-perform the three other models in regards to both $F1_{macro}$ and $F1_{micro}$ on both datasets. Nevertheless, the Gaussian Naïve Bayes classifier scores highest in regard to correctly classifying the *against* class, and the Random Forest classifier scores highest for classifying the *neutral* class for the full dataset. The Random Forest model manages to outperform the Conditional LSTM for $F1_{macro}$ and $F1_{micro}$ on both datasets. In regards to both $F1_{macro}$ and $F1_{micro}$, the Quote LSTM outperforms the Bi-Directional Quote LSTM for the full dataset. The Bi-Directional Quote LSTM performs comparably to the Quote LSTM, while both the Conditional LSTM and two baseline models are lacking significantly behind.

For the *national policy* subset, the Random Forest classifier performs significantly better than for the full dataset, both in regards to $F1_{macro}$ and $F1_{micro}$, whereas the Gaussian Naïve Bayes classifier performs slightly worse. It can again be observed, that the Conditional LSTM is out-performed significantly by the Quote LSTM and Bi-Directional Quote LSTM, both in regard to $F1_{macro}$ and $F1_{micro}$, that the Random Forest classifier outperforms the Conditional LSTM, and that both Quote LSTM and Bi-Directional Quote

LSTM outperform the Random Forest classifier. All three LSTM implementations perform better on the *national policy* subset than on the full dataset, indicating that the more specified dataset makes it easier for the models to learn the patterns within the data.

6.1.2 Misclassification analysis

For both datasets, the strength of the Quote and Bi-Directional Quote LSTMs is their ability to correctly classify *for* and *neutral* quotes. As a function of this, the models more or less ignore the *against* class, in pursuit of correctly classifying the two larger classes instead. For both models, a tendency can be observed towards classifying *against*-quotes as *for*, and to some extend also misclassifying some *neutral* quotes as *for*. This is not surprising, *for* being the majority class. For the Conditional LSTM, a significant under-representation of quotes classified as *for* can be observed, with more quotes classified as *against* for both the *national policy* and full dataset.

The confusion matrices of the two benchmark models differs significantly from each other, as well as those of the LSTMs. The Gaussian Naïve Bayes seems to overestimate the number of quotes in the *against* class significantly, labelling the majority of quotes as within this class. In doing so, it misclassifies a large number of *neutral* and *for* quotes as *against*. The Random Forest classifier correctly labels significantly more quotes as *neutral* than the Bayes classifier, its strength coming primarily from identifying this class correctly. In doing so, however, it overestimates the number of quotes within the class, much like the Bayes classifier does for *against* quotes, and misclassifies a significant number of *for* and *against* quotes as *neutral*, for both datasets.

6.1.3 Additional hyperparameter analysis

The fact that a dropout value of 0.5 performs better than 0.0, 0.2 and 0.7, as presented in 5.5, might have been influenced by the fact that the other hyperparameters were trained using a dropout value of 0.5, meaning that another optimal dropout value might have been found, had the parameter been included in the full hyperparameter search. From Table 5.5 it is observable that a dropout value of 0.7 leads to a slower learning curve, as the model did still not escape the local maxima of voting exclusively for *neutral* after 100 epochs, which is the case for the other dropout values. This is not surprising, as too high a dropout might lead to slower learning, as too high a number of nodes are excluded. Furthermore, using a lower dropout is shown to speed up the initial learning, observing as a dropout value of 0.2 and no dropout at all outperform a dropout value of 0.5 at 100 and 200 epochs. A decrease in performance can be observed for both a dropout value of 0.0, 0.2 and 0.7 when passing the 200 epoch mark.

As can be seen in Table 5.6, a higher learning rate is shown to decrease the time it takes for the model to converge on a high F1-score significantly, managing to escape the local maxima of voting exclusively for *neutral* after just 30 epochs for learning rate values of 0.005, 0.01 and 0.1, whereas a value of 0.001 only achieves this after 100 epochs. A higher learning rate can be seen to reduce the performance of models for higher numbers of epochs, as the peak performance of the models are found at either 30 or 50 epochs. The fact that the models applying a higher learning rate can not achieve as strong a performance as that using a learning rate of 0.001 is likely to be due to the models skipping some maxima,

as they take too large steps. A solution to this, which might be able to achieve the quick convergence of the models with high learning rate simultaneously with the precision of the low learning rate model, would be to implement a variable learning rate. A variable learning rate would function by reducing the learning rate, once the model starts to show diminishing loss reduction for each epoch.

6.1.4 The effect of context-based features

Within this subsection, we explore the results of in- and exclusion of the context-based features described in Section 2.2.1. The first and possibly most noteworthy observation that can be extracted from the overview of model performance for varying contextual features, presented in Table 5.7, is the fact that all three models initially converge on classifying all quotes as being within the *for* class, whereas for other models, it has been the rule that they initially converge on classifying all quotes as being within the *neutral* class.

This indicates that information regarding which politician the quote is from, and what political party the politician is from, pushes models towards classifying quotes as within the *neutral* class. Furthermore, the complete exclusion of contextual features leaves the model unable to move past the local maximum of classifying all quotes as *for*. This has been tested several times, with the same result. While the models using data including either the politician-vector or party-vector does manage to move on from this maximum, the models still perform significantly worse than the model including both, which reaches a $F1_{macro}$ of 0.575, thus showing that the inclusion of contextual features has a significant effect of the model's performance.

6.1.5 Error analysis

Dataset size

It is assumed that the small size of the quote dataset is a significant factor in preventing the models from achieving better performance, seeing as a smaller dataset size makes generalization to unobserved data points more difficult. To test this hypothesis, experiments were performed on the Quote LSTM using the optimal model hyperparameters, but a reduced training set sizes, in the range of 10 - 100 % of the total quote dataset, the results of which can be found in Table 6.1. The F1-score included in the table was extracted from the epoch with best model performance. From this table, it is clear that decreasing the training set size reduces the performance of the model.

In Figure 6.2 a graph representation of the data in Table 6.1 can be found, including a simple linear forecast of that data. Disregarding the expected diminishing returns of a growing dataset, an increase in the dataset size of between 60 and 80 % should be able to generate a perfect F1-score of 1. However, due to the presence of noisy data and edge cases, both in the test and training data, this is not likely to be the case. Nevertheless, it can be assumed based on the data, that an increase in dataset size would increase the model performance.

	Quotes	Optimal epoch	$\mathbf{F1_{micro}}$	$\mathbf{F1_{macro}}$
10%	72	Any	0.383	0.185
20%	144	Any	0.383	0.185
30%	216	Any	0.383	0.185
40%	288	200	0.5	0.33
50%	360	300	0.478	0.33
60%	432	200	0.567	0.425
70%	504	200	0.583	0.428
80%	576	200	0.656	0.488
90%	648	300	0.727	0.52
100%	720	300	0.717	0.575

Table 6.1: Overview of the effect of dataset size on the performance of Quote LSTM, percentages representing what amount of the quote dataset was used for model training

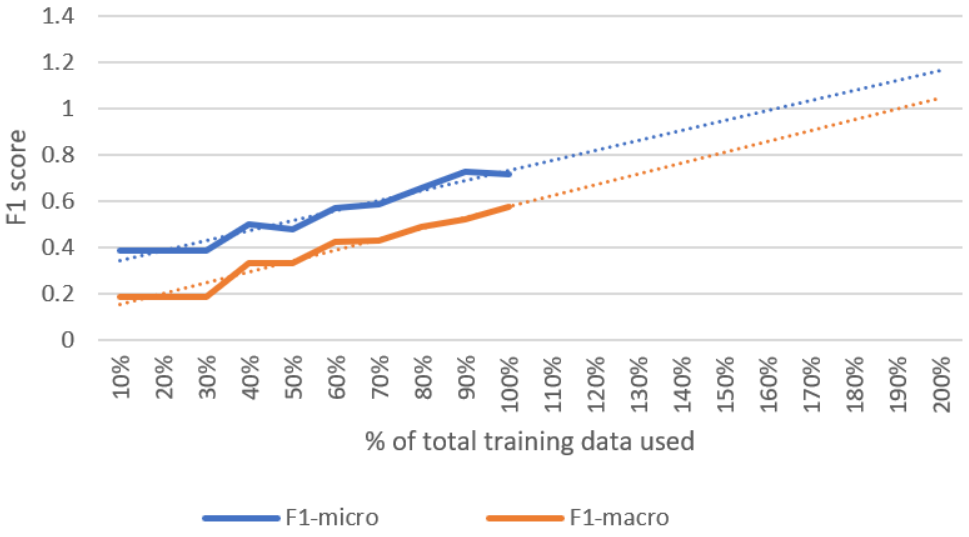

Figure 6.1: Graph over $F1_{micro}$ and $F1_{macro}$ using varying dataset sizes, including linear forecast

Overfitting

A number of steps have been taken to prevent the implemented models from overfitting due to the observed skewness of the dataset, including the use of dropout and L2-regularization. However, as can be observed from the test performed on the Quote LSTM visualized in Figure 6.2, the model still seems to overfit. Optimization on the training set works well up to and including 500 epochs, as seen in the falling loss for each epoch on the training data. However, past 300 epochs, the $F1_{macro}$ score for the test set decreases, showing the model to overfit to the training data past that point. Further steps to minimize overfitting might include implementation of a simple batching method or, if this is not effective, synthetic minority over-sampling as discussed in Section 4.4.

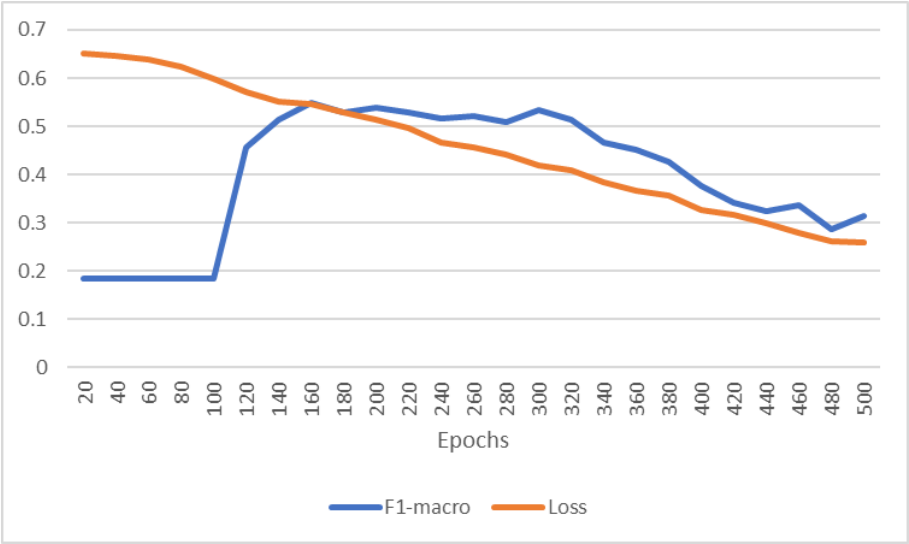

Figure 6.2: Graph showing the training of Quote LSTM, comparing loss to $F1_{macro}$

The optimal optimizer

The decision to use a simple SGD optimizer was made early in the development process, prior to the search of hyperparameter spaces for models. Thus, little testing was performed for the alternative optimizers mentioned in Section 4.3.2. To gain insight into whether the use of these alternative optimizers would have improved performance, a comparative experiment was performed, the results of which are presented in Table 6.2.

	Adagrad		**Adadelta**		**Adam**	
learning rate	0.001	0.01	0.001	0.01	0.001	0.01
Best epoch	Any	200	300	30	30	Any
F1micro	0.383	0.633	0.722	0.622	0.661	0.383
F1macro	0.185	0.490	0.518	0.536	0.547	0.185
Facc	0.000	0.907	0.884	0.759	0.791	0.000
Aacc	0.000	0.080	0.000	0.240	0.160	0.000
Nacc	1.000	0.493	0.783	0.594	0.681	1.000

Table 6.2: Comparison of performance of the Adagrad, Adadelta and Adam optimizers in the Quote LSTM

From this table it can be seen, that the Adam optimizer reaches an $F1_{macro}$ score of 0.547, comparable to the best score of the basic SGD optimizer which was 0.575, despite hyperparameters being trained using the basic SGD optimizer. It is worth noting that this result is achieved after only 30 epochs, whereas the basic SGD optimizer required 300 epochs.

Hyperparameter search

There is a risk that the optimal hyperparameter combinations for the Conditional LSTM and Bi-Directional Quote LSTM were overlooked during search of the hyperparameter spaces, due to the reduced space used for the two models. By applying a random search

approach, as described in Section 5.1, and identifying which hyperparameters generally achieve good results for the two models, a hyperparameter space might be defined specifically for each of the models. The reduced and model-specific hyperparameter space might subsequently be searched grid-wise, identifying the actual optimal hyperparameter combinations for the two models. This approach might also be applied to the Quote LSTM, identifying and discarding hyperparameter values that do not perform well, thus speeding up the search process.

6.2 Dataset analysis

The purpose of this section is to extract knowledge regarding the opinions and media coverage of Danish political parties in regards to the topic of immigration, from the dataset generated as described in Chapter 3. It is worth noting the quote count for each party in each dataset, shown in Table 6.3. The *centralization* dataset is small, and thus attaining a univocal or near univocal distribution of quote labels for the dataset is easier, especially so for parties such as Alternativet, Det Konservative Folkeparti and Socialistisk Folkeparti, who each have only two quotes within this dataset. Likewise, it is easier to attain a univocal quote distribution on the *national policy* dataset for Alternativet, than it is for Dansk Folkeparti, Venstre or Socialdemokratiet.

Party	**Dataset**	
	Centralization	Naitonal policy
Alternativet	2	9
Dansk Folkeparti	30	217
Det Konservative Folkeparti	2	24
Enhedslisten	5	34
Liberal Alliance	0	18
Radikale Venstre	7	93
Socialdemokratiet	9	154
Socialistisk Folkeparti	2	33
Venstre	47	212

Table 6.3: Quote count for each party by dataset

6.2.1 Quote distribution within parties

From Figure 6.3 it can be observed that, for most political parties, either none or very few quotes labeled as *against* were identified for the *centralization* sub-set. Only the parties Enhedslisten and Dansk Folkeparti deviate from this pattern, with no quotes identified as *for centralization* from Enhedslisten, and a few identified from Dansk Folkeparti. It is worth noting how only *for* quotes have been observed for Det Konservative Folkeparti on this subtopic, despite the party not being unreserved supporters of yielding legislative power to EU [Det Konservative Folkeparti, 2019].

Figure 6.4 shows a different picture than Figure 6.3, with a significantly larger degree of dispersion. This can probably partially be attributed to the larger size of the *national policy* subset, on which the figure is based. From Figure 6.4 it can be observed that

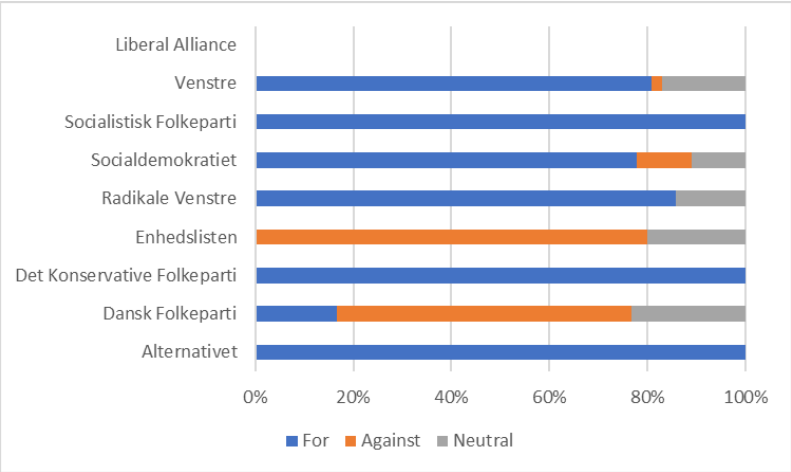

Figure 6.3: Quote distribution for the subtopic *centralization* between the labels for, against and neutral, for each party, in percentage

Alternativet is the only party univocally against implementing tighter immigration policy. Enhedslisten, Radikale Venstre and Socialistisk Folkeparti are largely *against*, with approximately 80 % of quotes within this class. It is worth noting that the quote distributions of both Socialdemokratiet and Liberal Alliance differ significantly from the rest of the parties within their half of the political spectrum, and to a higher degree resembles that of their political opponents. With a *for* distribution of 60 %, Socialdemokratiet resides more closely to the right-wing total of 75 % than the 32 % of the left-wing total, and with a *for* distribtuion of 32 % Liberal Alliance matches that of the left-wing total. However, Liberal Alliance has a lower *against* quote distribution than the left-wing total, and Socialdemokratiet has a larger *against* distribution than the right-wing total. Venstre, Det Konservative Folkeparti and Dansk Folkeparti all have a very low *against* distribution, with Dansk Folkeparti holding the smallest at just a few %.

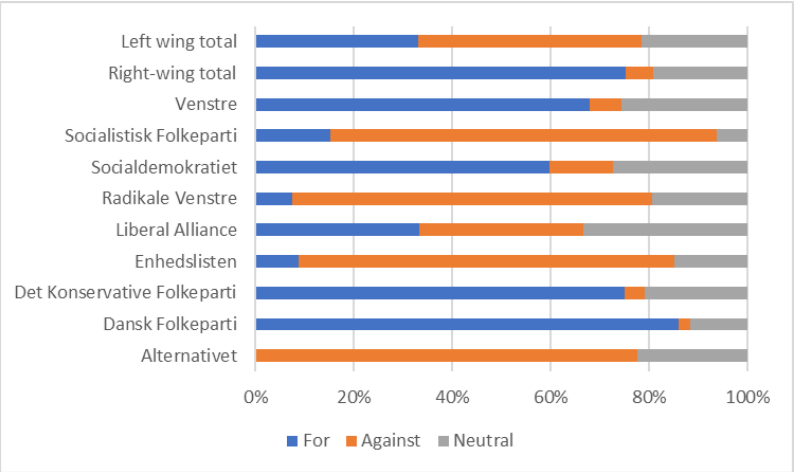

Figure 6.4: Quote distribution for the subtopic *national policy* between the labels for, against and neutral, for each party, in percentage, totals calculated as sums of quotes

6.2.2 Quote distribution between parties

In Section 3.7, the distribution and skewness of the final dataset was presented, and the skewness towards right-wing parties, in particular Venstre and Dansk Folkeparti was made clear. When comparing the distribution of quotes within the dataset, as presented in Figure 6.5, with the distribution of mandates in parliament between political parties, as presented in Figure 6.6, a significant discrepancy can be observed. It might be expected, that a larger number of mandates, and as a function of this larger power in parliament, would result in a given political party being more quoted, but this is not necessarily the case. Despite having the most mandates, Socialdemokratiet resides at a third place in regards to number of quotes, whereas Radikale Venstre resides at a fourth place in regards to quotes, despite ranking seventh for number of mandates.

This discrepancy might have many root courses, examples of which include an explicit decision from the party's side to focus on other political topics, power in parliament depending on more than the number of mandates a given party has, for instance minister posts and which wing is in government, and explicit or implicit bias from the media, caused by journalists' own political opinions or simply which politicians generally provide more interesting and news-worthy interviews.

It is worth noting, that this quote distribution seems to go against the observations made in Section 6.1.4. Here we see that inclusion of contextual features regarding who the politician behind a given quote is, and what party that politician comes from, pushes models towards classifying quotes with the *neutral* label. However, from Figure 6.5 it is observed that the two parties with the most quotes in the dataset are Dansk Folkeparti and Venstre, and from Figure 6.4 it is observed that the quote distribution within these parties lean heavily towards the *for* quote. Thus, it would be expected that the inclusion of the aforementioned context-based features would push the models towards classifying quotes with the *for* label. This might be due to a higher level of pattern complexity within quotes of the *for* label, making them more difficult to identify than those of the *neutral* label.

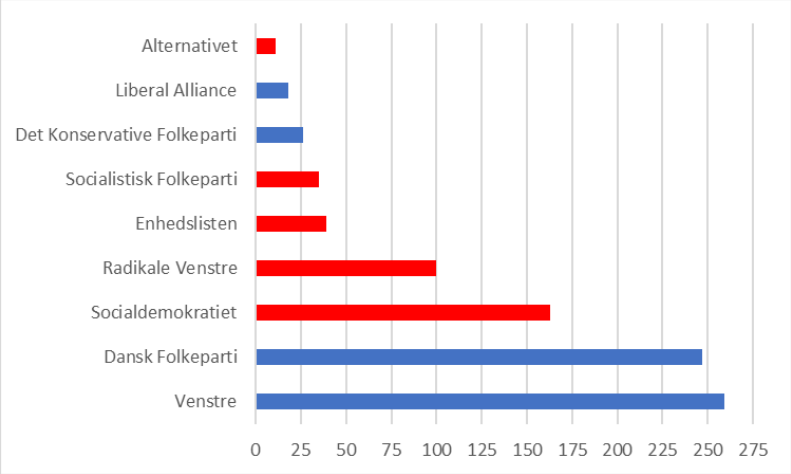

Figure 6.5: Total quote count comparison between all parties for the full dataset, bars coloured red representing left-wing parties, bars coloured blue representing right-wing parties

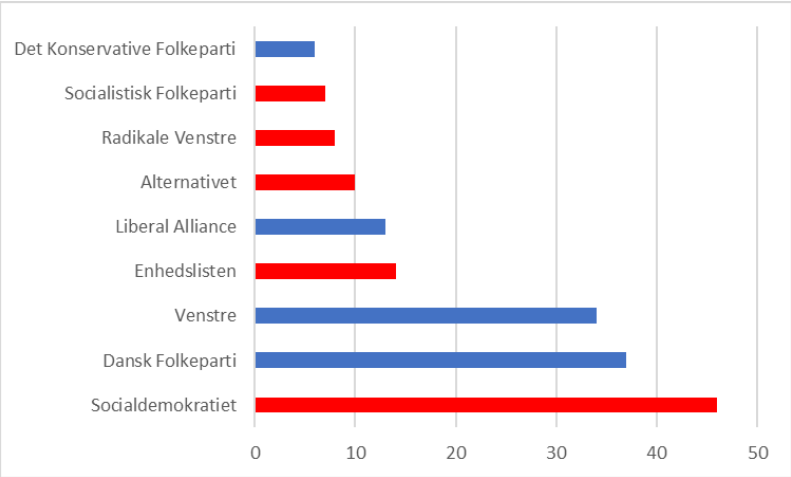

Figure 6.6: Number of mandates in parliament for each party, bars coloured red representing left-wing parties, bars coloured blue representing right-wing parties

6.2.3 Outlier analysis for Radikale Venstre and Dansk Folkeparti

From Figure 6.4 it can be observed that Dansk Folkeparti and Radikale Venstre seem to have very clear party lines in regards to immigration policy, with few quotes falling into the minority label, being *against* for Dansk Folkeparti and *for* for Radikale Venstre.

For Radikale Venstre, we observe three *for*-quotes from both Morten Østergaard and Sofie Carsten Nielsen. Five of these voice the opinion, that requirements regarding participation in the Danish work force should be set for immigrants, after coming to Denmark, and a single articulate immigration as a problem. An example of a quotes speaking *for* setting requirements for immigrants is found below.

> *Alle skal arbejde, hvis de kan. Det forventer vi af alle etniske danskere på offentlige ydelser, og det gælder naturligvis også for indvandrere og flygtninge.*
> Sofie Carsten Nielsen, [Ritzau, 2018f]

For Dansk Folkeparti, we observe one *against*-quote from Kristian Thuesen Dahl and two for Martin Henriksen. In general for these three quotes, it is found that the lack of interest in legislating against immigrants and immigration does not seem based on a belief that immigration might benefit society, but simply on finding a proposed piece of legislation ineffective or redundant. An example of such a quote is found below.

> *Selvfølgelig skal man da have lov til at være fodboldtræner, hvis ens søn spiller på et hold. Den del kan vi godt støtte. Mange af disse ting ser jeg som uproblematiske, men også unødvendige.*
> Martin Henriksen, [Ritzau, 2018h]

Overall for the outliers within the two parties, it does not seem to be the case that these are due to mis-labelling, but from an edge case opinion for Radikale Venstre, and an edge case situation for Dansk Folkeparti.

Chapter 7

Conclusion

The primary goal of this research has been to produce a dataset and a stance detection model which can be applied for future research, within the field of Danish NLP as well as others, and this goal has been achieved.

A dataset of quotes from Danish politicians from all political parties in parliament regarding the topic immigration was created through manual data collection, cleaning and annotation, and an annotation guideline was defined for the dataset. The media outlet Ritzau was used as source, due to considerations regarding objectivity of the data. Here, Ritzau was determined to be the most objective out of the considered alternatives, due to the outlet being owned by a conglomerate of other media outlets, from each side of the political spectrum. Immigration was defined as the topic to be included in the dataset, due to the fact that the topic ranked highly among the topics most important to the Danish population when deciding where to place their votes, and alternative topics being defined too broadly to easily allow a clear definition of *for*, *against* and *neutral* quotes. A skewness in the dataset was observed towards right-wing parties with a total of 61 % of all quotes, and males with a 60 % of all quotes. Furthermore, the dataset was observed to be skewed towards the class *for*, with the classes *against* and *neutral* being under-represented.

Three deep learning-based stance detection classifiers were built based on a recurrent LSTM architecture applying both conditionality and bi-directionality. A hyperparameter space was defined and searched for each of these models, identifying the optimal hyper-parameter settings. The simplest model implementation, using average word embeddings across a full quote rather than a sequence of words as input, performed the best with a $F1_{macro}$ score of 0.575, outperforming the two other LSTM implementations as well as the implemented Random Forest and Gaussian Naïve Bayes benchmark models. The two best performing models were shown to primarily misclassify the label *against*, while performing well on classification of the labels *for* and *neutral*. Experiments with varying learning rate showed that a variable learning rate might be implemented to increase convergence on good results, lowering the learning rate after convergence to find the specific performance maximum. Furthermore, contextual features, including which politician was quoted, and the party affiliation of that politician, were shown to have highly significant influence over the model results, achieving poor results while excluding one of the two contextual features, and performing majority voting if both features were excluded. Error analysis showed that an increased dataset size would have improved performance, and that the models perform overfitting when a large amount of epochs are used. It was found that the

use of a more advanced stochastic gradient descent-based optimizer than the one applied in this paper, might have improved both performance and runtime.

Finally, an analysis of the distribution of quotes between the three labels within each party showed that distributions within the left-wing and right-wing respectively were very similar across parties, with the exception of the right-wing party Liberal Alliance, for which the label distribution was more similar to that of left-wing parties, and the left-wing party Socialdemokratiet, for which the label distribution was more similar to right-wing parties. By looking closer at the parties with the lowest number of labels within the minority class within their parties, from the left and right wing respectively, it was found that these outlier quotes stemmed from an edge opinion outside of the party's regular opinion pattern for Radikale Venstre, and an edge case situation for Dansk Folkeparti, where the candidate simply did not have an interest in the topic about which he was asked.

In conclusion, a dataset of quotes from Danish politicians, including the quoted politician and the quoted politician's party, annotated for use in stance detection was generated, and annotation guidelines for this dataset were defined. Three deep learning-based classifiers using an LSTM architecture were designed, implemented and optimized for the task, two of which achieve reasonable performances. It was found that the two models taking an averaged quote embedding as input far outperformed the model taking a single word at each time step.

7.1 Future application of research

This section presents possible applications of the dataset, models and research presented within this paper.

7.1.1 Applications within Danish politics

The generated dataset can be applied for comparative analyses of the opinions expressed by Danish politicians, both between politicians and parties, but also, by enhancing the dataset with statistics regarding voting patterns and political parties' programs, to uncover discrepancies between these datasets, identifying which parties and politicians express opinions in line with their party programs, and which vote in parliament in line with the opinions they express. Furthermore, statistical analysis using the quote dataset might be used to identify possible political alliances based on, for instance, a comparison of stance distribution across parties.

During elections, so-called candidate tests, where a user inputs their opinions and receive the political candidate best matching their political orientation, are wide-spread. [DR, 2019, TV2, 2019, Politikken, 2019, Rosendahl, 2019] These tests, are based on politicians entering their own opinions, thus creating their own profile, with which the users are matched. Using a labelled quote dataset such as that generated for this thesis project would be an alternative solution to this, matching users with politicians based on the opinions they express to the media, and based on several instances rather than just one.

7.1.2 The Danish field of NLP

The generated stance detection classifiers might be applied to expansion of the dataset, by automatically labelling new quotes, and supplementing training data with this. Tests would have to be run with manually labelled quotes, to uncover whether this improves performance, and for this approach to become applicable, further improvements of the models might be necessary, to increase performance.

Further research might include analyses regarding how well knowledge travels between politicians and political parties. One might look into whether knowledge regarding one politician or party can translate to other politicians and parties, by training a model on a single politician or a single party, and testing the model on another politician or party. Was the dataset to be expanded with further topics, similar analyses could be applied to examine how well knowledge travels between topics.

Furthermore, the generated dataset might be applied as a resource for testing stance detection models for Danish. Here, the implemented classification models can be applied as benchmarks for comparison with new models.

7.1.3 Politological, sociological and communications research

The generated dataset might be applied to several areas of research outside NLP, and three such approaches are described below.

By identifying the most influential n-grams for stance classification, for instance using the NLTK package which contains a tool for this task [Bird et al., 2019], and using these n-grams as a basis for a discourse analysis, using n-grams as nodal points, chains of equivalence and difference could be identified within the political discourse surrounding immigration, following the discourse analysis approaches introduced by [Mouffe and Laclau, 2000].

An analysis of power structures within Danish politics might be performed, by observing the flow of certain words and phrases over time, identifying which politicians first make use of a phrase or word, that later becomes part of the vocabulary of other politicians. Such an analysis might be applied in correlation with an analysis of Strategic Action Fields within the Danish political field, following the work of [Fligstein and McAdam, 2011], using the analysis of flow of words and phrases to identify power relations, social relations and rules within the field.

Finally, an analysis of how Danish politicians communicate, through the quotes in the generated dataset, might be applied within an analysis of the systemic structure within political sub-systems as well as between political and journalistic sub-systems, using systems theory from [Luhmann, 1993] or [Katz and Kahn, 1979].

Bibliography

[Augenstein et al., 2016] Augenstein, I., Rocktäschel, T., Vlachos, A., and Bontcheva, K. (2016). Stance Detection with Bidirectional Conditional Encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, USA.

[Bahuleyan and Vechtomova, 2017] Bahuleyan, H. and Vechtomova, O. (2017). UWaterloo at SemEval-2017 Task 8: Detecting Stance towards Rumours with Topic Independent Features . In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, page 461–464, Vancouver, Canada.

[Bird et al., 2019] Bird, S., Klein, E., and Loper, E. (2019). *Natural Language Processing with Python*.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*.

[Chen et al., 2017] Chen, Y.-C., Liu, Z.-Y., and Kao, H.-Y. (2017). IKM at SemEval-2017 Task 8: Convolutional Neural Networks for Stance Detection and Rumor Verification. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, Vancouver, Canada.

[Christou, 2018] Christou, L. (2018). Artificial intelligence and the future of politics. *Verdict*.

[Collins and Duffy, 2001] Collins, M. and Duffy, N. (2001). Convolutional Kernels for Natural Language. In *Proceeding NIPS'01 Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, pages 625–632, Vancouver, Canada.

[Derczynski et al., 2017] Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Hoi, G. W. S., and Zubiaga, A. (2017). SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, page 69–76, Vancouver, Canada.

[Det Konservative Folkeparti, 2019] Det Konservative Folkeparti (2019). Eu og europa.

[DR, 2019] DR (2019). Hvem er du mest enig med?

[Duchi et al., 2011] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research 12*.

[Enayet and El-Beltagy, 2017] Enayet, O. and El-Beltagy, S. R. (2017). Determining Rumour and Veracity Support for Rumours on Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, pages 470–474, Vancouver, Canada.

[Fares et al., 2017] Fares, M., Kutuzov, A., Oepen, S., and Velldal, E. (2017). Word vectors, reuse, and replicability: Towards a community repository of large-text resources. *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa.*

[Fligstein and McAdam, 2011] Fligstein, N. and McAdam, D. (2011). *Toward a General Theory of Strategic Action Fields.*

[Goldberg, 2017] Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing.* Morgan and Claypool.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning.* MIT Press.

[Grave et al., 2018] Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).*

[Hjarvard, 2007] Hjarvard, S. (2007). Den politiske presse - en analyse af danske avisers politiske orientering. *Journalistica - Tidsskrift for Forskning I Journalistik, Nr. 5: Journalistik og presse i forandring.*

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation.*

[Infomedia, 2019] Infomedia (2019). Mediearkiv.

[ITU, 2019] ITU (2019). Nlp at itu copenhagen.

[Iyyer et al., 2014] Iyyer, M., Enns, P., Boyd-Graber, J., and Resnik, P. (2014). Political Ideology Detection Using Recursive Neural Networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1113–1122, Baltimore, Maryland, USA.

[Jønch-Clausen, 2015] Jønch-Clausen, H. (2015). Fra hvad til hvorfor? - subjektive formidlingsaktiviteter i valgreportager 1990-2007. *Journalistica - Tidsskrift for Forskning I Journalistik, Nr. 1: Åbent Tema.*

[Johnson and Goldwasser, 2016] Johnson, K. and Goldwasser, D. (2016). "All I know about politics is what I read in Twitter": Weakly Supervised Models for Extracting Politicians' Stances From Twitter. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2966–2977, Osaka, Japan.

[Jurafsky and Martin, 2018] Jurafsky, D. and Martin, J. H. (2018). *Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.*

[Katz and Kahn, 1979] Katz, D. and Kahn, R. L. (1979). *The Social Psychology of Organizations.*

[Kim, 2014] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations.*

[Kochina et al., 2017] Kochina, E., Liakata, M., and Augenstein, I. (2017). Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, pages 475–480, Vancouver, Canada.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems.*

[KU, 2019a] KU (2019a). Dannet.

[KU, 2019b] KU (2019b). Natural language processing (nlp).

[Kvalvik, 2017] Kvalvik, K. (2017). Køn, indkomst og geografi: Her er vælgernes vigtigste dagsordener.

[Lai et al., 2016] Lai, M., Farias, D. I. H., Patti, V., and Rosso, P. (2016). Friends and Enemies of Clinton and Trump: Using Context for Detecting Stance in Political Tweets. In *Advances in Computational Intelligence: 15th Mexican International Conference on Artificial Intelligence, MICAI 2016*, pages 155–168, Cancun, Mexico.

[Li et al., 2017] Li, X., Chen, W., Wang, T., and Huang, W. (2017). Target-Specific Convolutional Bi-directional LSTM Neural Network for Political Ideology Analysis. In *Web and Big Data. APWeb-WAIM 2017.*, pages 64–72, Beijing, China.

[Luhmann, 1993] Luhmann, N. (1993). *Gesellschaftsstruktur und Semantik: Studien zur Wissenssoziologie der modernen Gesellschaft. Band 2.*

[Ma et al., 2018] Ma, J., Gao, W., and Wong, K.-F. (2018). Detection of stance and sentiment modifiers in political blogs. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 1980–1989, Melbourne, Australia.

[Mansø, 2018] Mansø, R. G. (2018). R-forslag: Man skal kunne straffe for uagtsom voldtægt. *Danmarks Radio.*

[Miyake, 2019] Miyake, K. (2019). How will ai change international politics? *The Japan Times.*

[Mohammad et al., 2016] Mohammad, S. M., Kiritchenko, S., Sobhani, P., Zhu, X., and Cherry, C. (2016). A Dataset for Detecting Stance in Tweets. In *LREC*, Portoroz, Slovenia.

[Mohammad et al., 2017] Mohammad, S. M., Sobhani, P., and Kiritchenko, S. (2017). Stance and sentiment in tweets. *ACM Transactions on Internet Technology (TOIT) - Special Issue on Argumentation in Social Media and Regular Papers*.

[Mouffe and Laclau, 2000] Mouffe, C. and Laclau, E. (2000). *Hegemony and Socialist Strategy - Towards a Radical Democratic Politics*.

[NLTK, 2019] NLTK (2019). Natural language toolkit.

[Olah, 2015] Olah, C. (2015). Understanding lstm networks.

[Pascanu et al., 2013] Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceeding ICML'13 Proceedings of the 30th International Conference on International Conference on Machine Learning*.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[Politikken, 2019] Politikken (2019). Tag kandidattesten.

[Polonski, 2017] Polonski, V. W. (2017). How artificial intelligence conquered democracy. *The Independent*.

[Qazvinian et al., 2011] Qazvinian, V., Rosengren, E., Radev, D. R., and Mei, Q. (2011). Rumor has it: identifying misinformation in microblogs. In *EMNLP '11 Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599, Edinburgh, United Kingdom.

[Ritzau, 2018a] Ritzau (2018a). Alternativet vil i eu med fyret græsk finansminister. *Ritzau*.

[Ritzau, 2018b] Ritzau (2018b). Df: S rykker sig mere og mere i udlændingepolitiken. *Ritzau*.

[Ritzau, 2018c] Ritzau (2018c). Laveste asyltal i knap et årti tilfredsstiller ikke df. *Ritzau*.

[Ritzau, 2018d] Ritzau (2018d). Nye borgerlige vil have fn-aftale til folkeafstemning. *Ritzau*.

[Ritzau, 2018e] Ritzau (2018e). Overblik: Det siger politikerne om ghettoplanen. *Ritzau*.

[Ritzau, 2018f] Ritzau (2018f). Radikale: Flygtninge skal arbejde fra dag et! *Ritzau*.

[Ritzau, 2018g] Ritzau (2018g). S vil skabe fem danske jobcentre i sydeuropa. *Ritzau*.

[Ritzau, 2018h] Ritzau (2018h). Støjberg om lempede bijob-regler: Jeg forventer kort proces. *Ritzau*.

[Ritzau, 2018i] Ritzau (2018i). Støjberg vil have migranter fra dansk skib i land i italien. *Ritzau*.

[Ritzau, 2018j] Ritzau (2018j). Tyrkiske kulturformænds sympati for rabiat bevægelse alarmerer politikere. *Ritzau*.

[Ritzau, 2019] Ritzau (2019). Ejerskab, bestyrelse og årsregnskab.

[Rosendahl, 2019] Rosendahl, N. Y. (2019). Eu-kandidattest 2019: Hvem skal du stemme på til ep-valget?

[Årup Nielsen, 2019] Årup Nielsen, F. (2019). Danish semantic analysis.

[Russell and Norvig, 2016] Russell, S. J. and Norvig, P. (2016). *Artificial Intelligence, A Modern Approach*. Pearson Education Limited.

[Samoor, 2019] Samoor, A. (2019). polyglot.

[SDU, 2019] SDU (2019). Corpus linguistics.

[Skeppstedt et al., 2017] Skeppstedt, M., Simaki, V., Paradis, C., and Kerren, A. (2017). Detection of stance and sentiment modifiers in political blogs. In *19th International Conference, SPECOM 2017*, pages 1589–1599, Hatfield, United Kingdom.

[Tomanek and Hahn, 2009] Tomanek, K. and Hahn, U. (2009). Reducing class imbalance during active learning for named entity annotation. In *Proceeding K-CAP '09 Proceedings of the fifth international conference on Knowledge capture*.

[Tutek et al., 2016] Tutek, M., Sekulic, I., Gombar, P., Paljak, I., Culinovic, F., Boltuzic, F., Karan, M., Alagic, D., and Snajder, J. (2016). TakeLab at SemEval-2016 Task 6: Stance Classification in Tweets Using a Genetic Algorithm Based Ensemble. In *Conference: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 464–468.

[TV2, 2019] TV2 (2019). Kandidattest til folketingsvalget 2019.

[Wei et al., 2016] Wei, W., Zhang, X., Liu, X., Chen, W., and Wang, T. (2016). pkudblab at SemEval-2016 Task 6 : A Specific Convolutional Neural Network System for Effective Stance Detection. In *Conference: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 384–388.

[Wikidata, 2019] Wikidata (2019). Front page.

[Yang et al., 2016] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical Attention Networks for Document Classification. In *Proceedings of NAACL-HLT*, pages 1480–1489, San Diego, USA.

[Zarrella and Marsh, 2016] Zarrella, G. and Marsh, A. (2016). MITRE at SemEval-2016 Task 6: Transfer Learning for Stance Detection. In *Conference: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 458–463.

[Zeiler, 2012] Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. *ArXiv 2012*.

[Zeng et al., 2016] Zeng, L., Starbird, K., and Spiro, E. S. (2016). Unconfirmed: Classifying Rumor Stance in Crisis-Related Social Media Messages. In *Proceedings of the Tenth International AAAI Conference on Web and Social Media (ICWSM 2016)*, pages 747–750, Cologne, Germany.

[Zhang, 2004] Zhang, H. (2004). The Optimality of Naive Bayes. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference*, Miami Beach, USA.

Chapter 8

Appendixes

8.1 Appendix A - Examples of articles in PDF format

Below is presented examples of articles downloaded in PDF format on 15/03/2019 from Infomedia's media archives, for the media outlets Information, Politiken, Berlingske, Børsen, DR, BT and Jyllands Posten. For the articles from Information and Berlingske, only the first page is included. Articles are represented by links to their location in the Infomedia media archive, as the articles are not public domain. Access to the Infomedia media archive is granted by IP at most major Danish universities.

Martin Henriksen: Danmark risikerer at få ry som discountland. Derfor må vi begrænse antallet af udenlandske studerende available at `https://apps.infomedia.dk/mediearkiv/link?articles=e6d4bb74`

Svensk stjernereporter: Journalisterne passede ikke deres arbejde, da antallet af flygtninge eksploderede, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e6ab8514`

Martin Henriksen: Det handler vist mest om at placere nogle bygninger. Det løser jo ikke de udfordringer, vi står over for, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e6d01a13`

DF: Flygtninge må med i strammere familiesammenføring, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e69446cb`

Martin Henriksen i hårdt debat-angreb på Pernille Vermund: "I er direkte uansvarlige", available at `https://apps.infomedia.dk/mediearkiv/link?articles=e6cab485`

Se billedet: Martin Henriksen omfavner mørk kvinde på Facebook - og hans følgere magter det ikke, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e6ad2ba1`

Martin Henriksen om udgangsforbud: Selvfølgelig skal Fatima passe sit job på tankstationen, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e691143f`

8.2 Appendix B - Articles referring to Ritzau as source

Below is presented examples of articles downloaded in PDF format on 15/03/2019 from Infomedia's media archives, referring Ritzau as co-author or citing Ritzau. Articles are represented by links to their location in the Infomedia media archive, as the articles are not public domain. Access to the Infomedia media archive is granted by IP at most major Danish universities.

DF: S rykker sig mere og mere i udlændingepolitikken, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e6dfd37e`

Dansk Folkeparti opruster på udlændingepolitikken, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e6e6029a`

DF: Hold asylansøgere væk med stramme regler, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e692fe56` from BT

DF siger nej til S-plan om jobcentre i Sydeuropa, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e6e704df`

DF: Hold asylansøgere væk med stramme regler, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e692d5cc` from Jyllands-Posten

Støjberg efter ø-besøg: Beslutningen står helt fast, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e700e01e`

8.3 Appendix C - Politicians chosen for inclusion in the dataset

List of politicians chosen for inclusion in the dataset for each political party, and the number of quote available for each politician, using the search criteria defined in Chapter 3, before data cleaning.

8.3.1 Dansk Folkeparti

- Kristian Thulesen Dahl, Partiformand og politisk ordfører
- Martin Henriksen, Gruppesekretær og Udlændinge- og værdiordfører
- Marie Krarup, Integrationsordfører og Gymnasieordfører
- Liselotte Blixt, Sundhedsordfører og Psykiatriordfører
- Mikkel Dencker, Klima- og Energiordfører
- Pia Kjærsgaard, Head of Parliament
- Søren Espersen, Næstformand, Udenrigsordfører
- Pia Adelsteen, Miljøordfører
- Peter Skaarup, Gruppeformand
- Lise Bech, Fødevare- og Landbrugsordfører

Politician	Number of quotes
Kristian Thulesen Dahl	243
Martin Henriksen	462
Marie Krarup	5
Liselotte Blixt	0
Mikkel Dencker	0
Pia Kjærsgaard	12
Søren Espersen	29
Pia Adelsteen	0
Peter Skaarup	137
Lise Bech	0

Table 8.1: Number of available quotes attained for Dansk Folkeparti pre cleaning

8.3.2 Socialdemokratiet

- Mette Frederiksen, Partiformand
- Mattias Tesfaye, Erhvervsuddannelses-, social dumping-, udlændinge- og integrationsordfører
- Jens Joel, Energi-, Forsynings- og Klimaordfører
- Flemming Møller Mortensen, Sundheds-, Forebyggelses- og Psykiatriordfører
- Mogens Jensen, Politisk næstformand og Kultur-, Idræt- og medieordfører

- Henrik Sass Larsen, Gruppeformand
- Nicolai Wammen, Politisk ordfører
- Astrid Krag, Ældre og indfødsretsordfører
- Lea Wemelin, Bæredygtighedsordfører
- Lise Bech, Fødevare- og Landbrugsordfører
- Christian Rabjerg Madsen, Miljøordfører

Politician	Number of quotes
Mette Frederiksen	297
Mattias Tesfaye	249
Christian Rabjerg Madsen	18
Jens Joel	0
Flemming Møller Mortensen	0
Astrid Krag	12
Mogens Jensen	0
Henrik Sass Larsen	5
Nicolai Wammen	63
Lea Wermelin	0

Table 8.2: Number of available quotes attained for Socialdemokratiet pre cleaning

8.3.3 Venstre

- Lars Løkke Rasmussen, Statsminister og Formand
- Inger Støjberg, Udlændinge- og Integrationsminister
- Lars Christian Lilleholt, Energi-, Forsynings- og Klimaminister
- Ellen Trane Nørby, Sundhedsminister
- Jakob Elleman-Jensen, Miljø- og Fødevareminister
- Britt Bager, Politisk Ordfører
- Erling Bonnesen, Fødevare- og Miljøordfører
- Thomas Danielsen, Energi- og Klimaordfører
- Jane Heitmann, Sundheds- og Psykiatriordfører
- Mads Fuglede, Flygninge-, Stabiliserings-, Integrations- og Udlændingeordfører

8.3.4 Enhedslisten

- Øjvind Vilsholm, Miljø-, Natur-, Familie- og Forbrugerordfører
- Rosa Lund, Rets-, Uddannelses- og Integrationsordfører
- Pelle Dragsted, Erhvervs-, Demokrati-, Grøn Omstillings og Kirkeordfører

Politician	Number of quotes
Lars Løkke Rasmussen	618
Inger Støjberg	822
Lars Christian Lilleholt	0
Ellen Trane Nørby	2
Jakob Elleman-Jensen	0
Britt Bager	24
Erling Bonnesen	0
Thomas Danielsen	0
Jane Heitmann	0
Mads Fuglede	19

Table 8.3: Number of available quotes attained for Venstre pre cleaning

- Søren Egge Rasmussen, Klima-, Energi-, Fødevare-, Fiskeri-, Bolig- og Dyrevelfærdsordfører
- Søren Søndergaard, EU-, Kultur-, Medie- og Udlændingeordfører
- Peder Hvelplund, Sundheds-, Psykiatri-, Ældre-, Indfødsrets og Velfærdsordfører
- Henning Hyllested, Transport-, Social Dumping-, Landdistrikt og Turismeordfører
- Pernille Skipper, Politisk-, Social-, Ligestillings- og LGBTQI-ordfører
- Nikolaj Villumsen, Europaråds- og Menneskeretsordfører
- Johanne Schmidt Nielsen, Barsel

Politician	Number of quotes
Øjvind Vilsholm	0
Rosa Lund	1
Pelle Dragsted	14
Søren Egge Rasmussen	0
Søren Søndergaard	6
Peder Hvelplund	0
Henning Hyllested	11
Pernille Skipper	86
Nikolaj Villumsen	68
Johanne Schmidt Nielsen	47

Table 8.4: Number of available quotes attained for Enhedslisten pre cleaning

8.3.5 Liberal Alliance

- Anders Samuelsen, Partileder og Udenrigsordfører
- Carsten Bach, Miljø-, Klima-, Energi-, Forsynings-, Udviklings-, Landbrugs-, Fødevare-, Grøndlands og Færøerneordfører
- Christina Egelund, Gruppeformand, Politisk og Retsordfører

- Henrik Dahl, Udenrigs-, Undervisnings-, Uddannelses-, Forsknings-, Afbureaukratiserings-, Indfødsrets og Europaordfører
- Joachim B. Olsen, Finans-, Skatte-, Integrations og Udlændingeordfører
- Laura Lindahl, Gruppesekretær, Beskæftigelses-, Social-, Børne-, Ligestillings og Offentlig Sektor ordfører
- May-Britt Kattrup, Sundheds-, Ældre-, Psykiatri-, Ældre råds-, §71 og Erhvervsordfører
- Villum Christensen, Bolig-, Landdistrikts og Forsvarsordfører samt statsrevisor
- Thyra Frank, Ældreminister
- Simon Emil Ammitzbøl-Bille, Økonomi- og Indenrigsminister

Politician	Number of quotes
Anders Samuelsen	29
Carsten Bach	4
Christina Egelund	21
Henrik Dahl	275
Joachim B. Olsen	9
Laura Lindahl	13
May-Britt Kattrup	4
Villum Christensen	2
Thyra Frank	0
Simon Emil Ammitzbøl-Bille	31

Table 8.5: Number of available quotes attained for Liberal Alliance pre cleaning

8.3.6 Alternativet

- Uffe Elbæk, Politisk Leder
- Carolina Magdalene Maier, Gruppeforperson, Idræts-, Folkeoplysnings-, Udlændinge-, Integrations-, Undersivnings-, Grundlovs-, Nydanskere og Mennesker på flugtordfører
- Julius Gorm Graakjær Grantzau, Børne-, Familie-, Skat-, Fordelings-, Statsborgerskabs og Samarbejdsordfører
- Christian Poll, Finans-, Miljø-, Landbrugs-, Fødevare-, Dyrevelfærds-, Energi-, Forsynings-, Vild Naturs og Grøn omstillingsordfører
- Torsten Gejl, Beskæftigelses-, Indenrigs-, Cannabis-, Social- og Ungeordfører
- Rasmus Nordqvist, Politisk-, EU-, Udenrigspolitisk-, Klima-, Freds-, Forsvars-, Kunst og Kultur og LGBTQI-ordfører
- Ulla Sandbæk, Udviklingsordfører
- René Gade, Rets-, IT og Digital Forbedringsordfører

- Roger Matthiesen, Fiskeri-, Færøerne-, Grønlands-, Tysk mindretals-, Bolig-, Mobilitets-, Ligestillings- og Mangfoldighedsordfører samt ordfører for et sammenhængende Danmark
- Pernille Schnoor, Helbeds-, Trivsels-, Sundheds-, Psykiatri-, Ældre-, Uddannelses og Forksningsordfører

Politician	Number of quotes
Uffe Elbæk	111
Carolina Magdalene Maier	12
Julius Graakjær Grantzau	0
Christian Poll	2
Torsten Gejl	0
Rasmus Nordqvist	3
Ulla Sandbæk	8
René Gade	0
Roger Matthisen	0
Pernille Schnoor	0

Table 8.6: Number of available quotes attained for Alternativet pre cleaning

8.3.7 Radikale Venstre

- Anders Steenberg, Skatte-, Udlændinge-, Landdistrikts og Transportordfører
- Ida Auken, Energi og Klima-, Miljø-, Erhvervs-, IT-, Iværksætteri og Forbrugerordfører
- Lotte Rod, Rets-, Børne-, Ungdomsuddannelses-, Sundheds-, Nærværs- og Indførdsretsordfører
- Marianne Jelved, Folkeoplysnings-, Kirke-, Skole-, Social-, Kultur- og Medieordfører samt orfdører for de frie skoler
- Martin Lidegaard, Gruppenæstformand, Finans-, Forsvars-, Udenrigs- og Kommunalordfører samt ordfører for Grønland, Færøerne og Nordisk Samarbejde
- Morten Østergaard, Politisk Leder, Gruppeformand, Verdensmålsordfører
- Rasmus Helveg Petersen, Arbejdsmarkeds-, Beskæftigelses-, Landbrugs-, Fødevare-, Fiskeri-, Tele og Boligordfører
- Sofie Carsten Nielsen, Gruppenæstformand, Uddannelses-, Forsknings-, Ligestillings-, Menneskerettigheds-, Integrations og EU-ordfører
- Zenia Stampe, p.t. på barsel, tidligere ordfører på lang række områder og opstillende til valget 2019
- Jens Rohde, Medlem af Europaparlamentet

8.3.8 Socialistisk Folkeparti

- Pia Olsen Dyhr, Formand, Klima- og Energiordfører
- Jacob Mark, Gruppeformand, Børne-, Uddannelses-, Undervisnings-, Forsknings-, Kultur-, Idræts og Medieordfører

Politician	Number of quotes
Anders Steenberg	0
Ida Auken	1
Lotte Rod	9
Marianne Jelved	6
Martin Lidegaard	7
Morten Østergaard	239
Rasmus Helveg Petersen	6
Sofie Carsten Nielsen	47
Zenia Stampe	4
Jens Rohde	16

Table 8.7: Number of available quotes attained for Radikale Venstre pre cleaning

- Karsten Hønge, Politisk-, Beskæftigelses-, Social Dumping-, Rets-, Transport- og Landdistriktsordfører samt ordfører for Grønland og Færøerne
- Holger K Nielsen, Udenrigs-, Forsvars-, Europa-, Flygtninge-, Integrations og Udviklingsordfører
- Lisbeth Bech Poulsen, Finans-, Uligheds-, Skatte-, Erhvervs-, Forbruger og IT-ordfører
- Trine Torp, Social-, Ligestillings-, Psykiatri-, Familie-, Natur-, Miljø-, Fødevare-, Langbrugs-, Fiskeri-, Kirke og Dyrevelfærdsordfører
- Kirsten Normann Andersen, Velfærds-, Sundheds-, Ældre-, Handicap-, Kommunal-, Bolig og Indfødsretsordfører
- Margrethe Auken, Medlem af Europa-parlamentet
- Signe Munk, Næstformand
- Rikke Lauritsen, folketings- og EU-parlamentskandidat

Politician	Number of quotes
Pia Olsen Dyhr	132
Jacob Mark	41
Karsten Hønge	23
Holger K Nielsen	1
Lisbeth Bech Poulsen	14
Trine Torp	10
Kirsten Normann	15
Margrethe Auken	0
Signe Munk	0
Rikke Lauritsen	0

Table 8.8: Number of available quotes attained for Socialistisk Folkeparti pre cleaning

8.3.9 Det Konservative Folkeparti

- Søren Pape Poulsen, Partiformand, Justitsminister

- Mai Mercado, Børne og Socialminister
- Rasmus Jarlov, Erhvervsminister
- Naser Khader, Forsvars-, Indfødsrets-, Kirke-, Medie-, Menneskerettigheds-, Rets-, Integrations og Værdiordfører
- Mette Abildgaard, Gruppeformand, Gruppesekretær, Grundolvs-, Klima-, Miljø-, Politisk og Sundhedsordfører
- Anders Johansson, Beskæftigelses-, By-, Bolig-, Bygnings-, Ehvervs-, Finans-, Reform-, Forbrugspolitisk-, Skatte og Afgiftordfører
- Birgitte Klintskov Jerkel, Familie-, Børne-, Forebyggelses-, Nordisk Samarbejde-, Psykiatri-, Social-, Handicap-, Transport-, Undervisnings og Ældreordfører
- Merete Scheelsbeck, EU-, Kultur-, Ligestillings-, Uddannelses-, Forsknings og Udviklingsbistandsordfører samt ordfører for Grøndland og Færøerne
- Orla Østerby, Dyrevelfærds-, Energi-, Forsynings-, Fiskeri-, IT-, Tele-, Idræts-, Lanbrugs-, Fødevare og Kommunapolitiksordfører samt ordfører for landdistrikter og små-øer
- Henrik Sølje Weiglin, Organisatorisk Næstformand

Politician	Number of quotes
Søren Pape Poulsen	71
Mai Mercado	22
Rasmus Jarlov	5
Naser Khader	40
Mette Abildgaard	23
Anders Johansson	0
Brigitte Klintskov Jerkel	4
Merete Scheelsbeck	0
Orla Østerby	0
Henrik Sølje Weiglin	0

Table 8.9: Number of available quotes attained for Det Konservative Folkeparti pre cleaning

8.4 Appendix D - Ritzau PDF examples from Infomedia

Below is presented an example of articles downloaded in PDF format on 01/03/2019 from Infomedia's media archives, from the media outlet Ritzau. The article represented by links to their location in the Infomedia media archive, as the articles are not public domain. Access to the Infomedia media archive is granted by IP at most major Danish universities.

Fock-afløser er alligevel klar til Folketinget, available at `https://apps.infomedia.dk/mediearkiv/link?articles=e6f218ee`