Aarhus University

Bachelor Project

Advisor: Carsten Eie Frigaard

BA project no.: 2021E79

Mitigation of stigma, prejudice and discrimination during early stages of recruitment using machine learning - Main Report

Navn:	*Underskrift:*	*Studienr.:*
Morten L. Hansen	Morten Lerschow Hansen	201805227
Rasmus F. Sørensen	Rasmus Føgh Sørensen	201806493

Characters incl. space: 70263
Submission deadline: Wednesday 15. December 2021 12:00

Abstract

Due to demographic characteristics and socioeconomic status, applicants are facing discrimination in the early stages of job recruitment due to conscious or subconscious stigmatization by the employer. Not only may applicants discarded by the employer based on demographic characteristics, but leading up to pursuing a job, how a job advertisement is phrased with respect to gendered wording can affect pursuit intentions. Studies find that race and ethnicity of the applicant can entail that the applicant has to apply for double as many jobs as an applicant appertaining a majority, to receive the same amount of callbacks. Studies also find, that if a job application or advertisement contains strong gendered wording, pursuit attentions are altered, and gender inequality sustained.

This report is set to analyze how machine learning can help mitigate the discrimination that is projected onto the applicant, by studying how natural language processing tasks, such as named entity recognition and sentiment analysis, can censor demographic characteristics to emphasize experience and skills, and additionally assists in gender neutralizing job applications and advertisements to enhance pursuit intentions. By splitting the problem up in two separate analyses that operate on separate domain-irrelevant datasets, several supervised neural networks were composed based on LSTM, BiLSTM and CNN's, to perform binary and multi-label classification. Precision, recall and F1-score were measured to compare all networks during testing. The results showed that NER can assist in recognizing named entities, such as demographic characteristics, given the dataset have a uniformly and strongly represented amount of named entities upon training. The results also showed that a binary and multi-label classification of text regarding gender polarization were not efficient, with score measurements way less than required, among other things due to the fact that the lingo of the dataset was influenced by its origin and the lacking diversity of gender polarities.

The results ultimately suggest, that machine learning performing named entity recognition can assist in censoring demographic characteristics if an adequate dataset is generated of sufficient size, and that to perform sentiment analysis to classify gendered wording would require a much more diverse dataset based on a more valid methodological approach regarding categorising gender-biased words.

Resumé

På grund af demografiske karakteristika og socioøkonomisk status bliver jobansøgere diskrimineret tidligt i jobansægningsprocessen grundet bevidst eller underbevidst stigmatisering af ansætteren. Derudover bliver ansøgerens incitament til at søge et job også påvirket af kønsladningen af jobbeskrivelsen. Studier viser, at ansøgerens race og etnicitet har en afgørende betydning på responsantal sammenlignet ansøgere tilhørende en majoritetsgruppe, hvor ratioen af respons fra arbejdsgiver er dobbelt så lav for ansøgere tihørende en minoritetsgruppe. Deruover viser studier også, at en overvejende kønsladet jobbeskrivelse kan afholde ansøgere fra at søge jobbet, hvilket i sidste ende medfører kønsulighed.

Denne rapport har til formål at analysere hvordan maskinlæring kan hjælpe med at mindske den diskrimination ansøgere bliver udsat for, ved at undersøge hvordan sprogprocessering, såsom entitetsgenkendelse og sentiment-analyse kan censurere demografiske karakteristika for at fremhæve ansøgerens kvalifikationer og erfaring, samt hjælpe med at kønsneutralisere job ansøgninger og beskrivelser, for at øge andelen af ansøgere. Ved at opdele problemet i to analyser som bearbejder to forskellige domæneirrelevante datasæt, bliver forskellige superviserede neurale netværk sammensat baseret på LSTM, BiLSTM og CNN for at udføre binær- og multiklassifikation. Præcision, recall og F1-score bliver udregnet for at sammenligne alle netværk under testning. Resultaterne viser, at entitetsgenkendelse kan genkende entiteter såsom demografiske karakteristika, givet at datasættet er uniformt fordelt og stærkt repræsenteret af alle entiteter under træning. Resultaterne viser også at binær- og multiklassifikation af tekst ift. kønsladning ikke var effektivt, med scoreresultater langt lavere end påkrævet, hvilket til dels skyldes lingoen af det anvendte dataset var præget af dets kilde samt manglen på diversitet af kønsladede ord.

Ultimativt set angiver resultaterne, at NER ved brug af maskinlæring kan hjælpe med at censurere demografiske karakteristika, såfremt et dataset af tilstrækkelig størrelse er anvendt. Derudover ville det påkræve et mere divers dataset samt en mere gyldig metodisk tilgang til at kategorisere kønladede ord for at anvende sentiment-analyse til at klassificere kønsladede ord.

Preface

This report serves as the main report to the annex report, *Mitigation of stigma, prejudice and discrimination during early stages of recruitment using machine learning - Annex Report*.

This main report is executed by Rasmus Føgh Sørensen and Morten Lenschow Hansen, both students of Aarhus University School of Engineering's BSc Software Engineering, in cooperation with the assigned advisor Carsten Eie Frigaard.

The project is to be handed in December the 15th, 2021, and is to be orally defended in Janurary 2022.

N.B. - For better reading of figures and tables, it is recommend to print the report in colors, if one were to prefer a physical edition.

Conventions used

The following typographical conventions are used:

Italic
: indicates quotes, references, equations and initial third party mentions.

Bold
: indicates new names and terms, except when used in figures, tables and listings, defined in *Glossary*.

Contents

Glossary

Here explanations of terms and abbreviations used in the report can be found.

Term	**Explanation**	**Comment**
Gendered wording	A context dependent measurement of a word or phrase's masculine and feminine polarity.	Masculine examples: 'he', 'competitive', 'active', 'confident', 'We are looking for a strong...', 'candidates who are aggresive' Feminine examples: 'she', 'support', 'nurture', 'communicate', 'nurture and connect with customers', 'build relationships' [1]
SaaS	Software as a service	
A	Applicant	Used consistently throughout report.
E	Employer	Used consistently throughout report.
Linguistic analysis	Analysis of machine learning algorithm to carry out named entity recognition.	
Sentiment analysis	Analysis of machine learning algorithm to carry out sentiment analysis.	
LA	Abbreviation of linguistic analysis.	
SA	Abbreviation of sentiment analysis.	
NLP	Natural Language Processing	
NE	Named entity	
XAI	Explained AI	In contrast to black box in ML, XAI is an implementation used to understand why an ML algorithm arrives at its conclusions.
NER	Named Entity Recognition	
POS	Part-of-speech	
ML	Machine Learning	
Corpus	Dataset	In ML datasets are often refered to as corpus, but in general terms it is a collection.
GMB	Groningen Meaning Bank	Provider of dataset for the linguistic analysis.
IOB	Inside, outside, beginning	Named entity format.
SOS	start-of-string	
EOS	end-of-string	
RNN	Recurrent neural network	
LSTM	long-short-term-memory	
biLSTM	bidirectional LSTM	
Demographic characteristics & Demographic features		Are used interchangeably.

Employer, organization & recruiter		Are used interchangeably.

Table 1: Glossary explaining terms and abbreviations used in the report.

Version history

Date	Version number	Initials	Comments
10/12-21	1.0	RFS	Initial iteration of main report.
10/12-21	1.1	RFS	Added *Introduction*, *Project description* and *Conceptualization & requirements*. Started first iteration of *Analysis*.
11/12-21	1.2	RFS	Added sections *Linguistic features* and *Datasets* to *Analysis*.
13/12-21	1.3	MLH & RFS	Added sections *Methodology*, *Results* and *Discussion*.
14/12-21	1.4	RFS & MLH	Added sections *Conclusion*, *Abstract* and *Resumé*.

Table 2: Version history

1 Introduction

1.1 Description of problem

People are subject to discrimination in the initial process of job recruitment, especially ethnic and immigrant-origin minority groups are disadvantaged to access the labour market [2]. The discrimination is typically based on demographic characteristics, socioeconomic perceptions of the applicant [2], or heuristics made by the employer [3]. Studies with correspondence experiments have been carried out throughout the last two decades. A recent study carried out in the Danish labour market found that the ratio of callbacks between a Danish-sounding name and a middle Eastern-sounding name is 1.52 for the same equally qualified application, implying the minority would have to apply 52% additional applications to get the same amount of callbacks as the majority. [2]. The same study also found discrimination of intersections between gender and ethnicity statistically significant between female minorities and female majorities in female dominated occupations and especially male minorities and male majorities in male dominated occupations [4]. The empirical context can also be extended to other countries. In America, an extensive research on racial discrimination in the American labour market in 2003 found, that the ratio of callbacks between a white-sounding name and a black-sounding name was 1.5 in favor of a white-sounding name [3].

The common denominator between mentioned studies are the employers performing the screening process of the applications, either manually by screening the application or automatically by use of a hiring system. A model explaining this screening process behaviour could be lexicographic search by the employer [3], where applicants are discarded after quick heuristics, based on names or photos, leading to cognitive bias instead of rational choices. Also, if an employer uses an automated hiring system, it shows that the underlying system can have special bias towards one gender, leading to exclusion of the opposite gender [5].

These studies only show observations based on one-way communication from applicant to employer in form of job applications, but going the other way around, job advertisements tend to contain **gendered wording**, which sustains gender inequality [6]. Gendered wording is not a defined quantity, but a context dependent measurement of a word or phrase with respect to a masculine and feminine polarity. Examples of gendered wording, which causes gender bias, can be obvious gender references like he/she or fireman/firewoman, but often occurs less apparent as people of specific genders tend to subconsciously prefer certain ways of expressing themselves in terms of words and phrasing [1]. In male-dominated occupations there is a statistical significance for job advertisements containing greater masculine wording [6], which ultimately affects job seekers pursuit intentions, due to anticipated belongingness. If a job advertisement contains greater feminine wording, men tend to have higher perception of gender diversity, which subsequently lead to lower pursuit intentions [6].

All things considered, some sort of discrimination is happening all over the line, even if it being an unconscious or conscious action in a manual or automated process. As of modern days, recruitment tends to go through an intermediary, such as recruitment platforms or social media [7], where perhaps this is an evident place to take action. Co-author of *'Monitoring hiring discrimination through online recruitment platforms'*, Dr. Dominik Hangartner, said:

> *"We are optimistic that at least part of the discrimination that we document in this study can be overcome by re-designing recruitment platforms. For example, more relevant information such as a candidate's work experience and education could be placed at the top, and details which might indicate ethnicity or gender, such as name or nationality, could appear much lower down the CV"* [8].

1.2 Problem statement

To mitigate discrimination in the initial job recruitment process, either it being manual or automated, it requires a change of behaviour of how the two parties, applicant (**A**) and employer (**E**) respectively, approach each other. Reflecting on the problems described in *1.1 Description of problem* and what the aim of the project should be, two key points can be deduced:

Key point 1. discrimination of A in regard to demographic characteristics and socioeconomic status, and

Key point 2. discrimination of A in regard to gendered wording, both in terms of wording in job applications, potentially leading to implicit cognitive perceptions of A, as well as in job advertisements, potentially discouraging A from applying for the job.

These key points play an essential role in the cognitive perception of one another, that fosters a subconscious bias. Especially during job recruitment, where discrimination towards A based on individual bias by E shouldn't be present. Therefore, the key points are vital in suppressing discrimination during entry to the labor market, so a more diverse and inclusive field of candidates can be assembled. This leads to a more generalized problem statement, that can be shortened to:

How can an unbiased recruitment process be established between A and E, where demographic characteristics, socioeconomic status and perception of gender are disregarded, and people are instead evaluated based on qualifications and experience?

1.3 State-of-the-art

Before going in depth with the project, this dedicated section will unfold already state-of-the-art services in the industry. These services are already used by many large-scale organisations and are an integrated part of the job recruitment process, whether it be in the screening process of candidates or processing of job advertisements before publication.

1.3.1 Automated hiring systems

In the job recruitment domain, machine learning is already playing a vital part, especially in the early stages of recruitment. The most commonly applied machine learning fall into the pits of *talent recruitment*, *talent sourcing* and *candidate screening* [9]. The manual screening process of job application is considered a high-volume task for recruiters, which is why many large-scale organizations implement an automated screening process to shortlist candidates. The automated screening process performs machine learning on job applications to rank candidates based on sentiment analysis, and extracted criteria down to distance in commute and gaps in employment history [10]. However, it has recently surfaced, that such automated hiring systems are not without bias, and they can reject millions of viable job candidates, simply based on picky requirements with no regard to qualifications or experiences of the applicant[10]. Another reason of exclusion could be due to the machine learning algorithms being trained on previous job application data, with special bias towards one gender [10][5].

1.3.2 Gender-biased language recognition

Textio is a powerful service used by companies to be more inclusive and diverse when recruiting. Textio supports gender-biased language measurement, by processing text and then highlighting it and suggesting changes to help recruitment personnel come across less gender-biased, when writing job advertisements. The service is built on datasets of more than 350 million job posts and their hiring outcome [11]. With former job descriptions skewing masculine, *Johnson & Johnson* reported an increase of 9% of female applicants in 2017, by using Textio during job recruitment [12].

Thus, Textio is considered the state-of-the-art **SaaS** for sentiment analysis with regard to gender-biased polarization, and is used by many organizations such as *Spotify*, *McDonald's*, *Twitter* etc. Textio is only used by E's recruitment personnel to come across unbiased, when writing job advertisements, whereas this report's sentiment analysis aims to help both A and E.

2 Project description

This project will revolve around the key points from *1.2 Problem statement* and analyze how machine learning, specifically neural networks, can be applied to aid solving the problem stated. More specifically, this project is separated into two individual neural network analyses and implementation; the **linguistic analysis** and the **sentiment analysis**, henceforth referenced as **LA** and **SA** respectively. The LA is dedicated to *Key point 1.*, whereas the SA is dedicated to *Key point 2.*.

Each analysis will be designed with regard to their intended purpose, stated below:

The intention of the LA is to recognize demographic characteristics in A's job application, so they can be censored for the initial round of candidate screening by E. This approach is used to emphasize the skills and experiences that A possess.

The intention of the SA is to recognize gendered wording, both in job applications as well as in job advertisements, so gendered wording can be highlighted and pointed out for the author, allowing for rephrasing. This approach is used to establish a gender neutral two-way communication line between A and E, so conscious and subconscious discrimination based on wording can be mitigated.

These analyses will play a fundamental part of the project, which will have an analytical and scientific perspective, rather than the traditional system development perspective. The LA and SA will both be proof of concept neural networks, and are both based on a thorough underlying analysis of feature engineering and different types of relevant neural network architectures. Each neural network will be measured by score metrics to evaluate its applicability, and if the score metrics are found satisfactory, this project could be the foundation of future work.

3 Conceptualization & requirements

As stated in *2 Project description*, the LA and SA are proof of concept neural networks, however designed with the intended purpose of solving the key points stated in *1.2 Problem statement*. Therefore, the first part of this section will act as an exposition where the LA and SA are showcased with the intended job recruitment context. In the second part of this section, the scope and requirement specifications are specified for the LA and SA and what is considered satisfactory for future work.

3.1 Concept description in a domain context

In the domain context of a job recruitment platform, machine learning could be an integrated part of the processing of job applications and job advertisements. The LA and SA are ideally part of a processing pipeline on a recruitment platform as shown in figure 1, where the process of A uploading a job application and E uploading a job advertisement is shown.

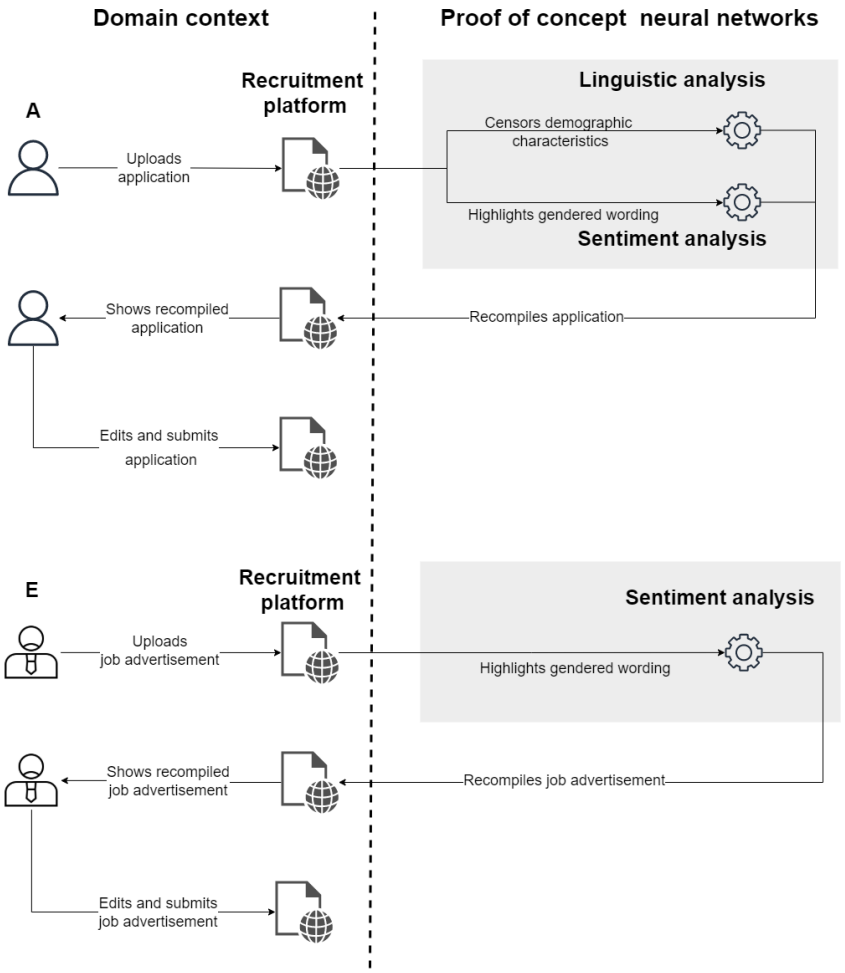
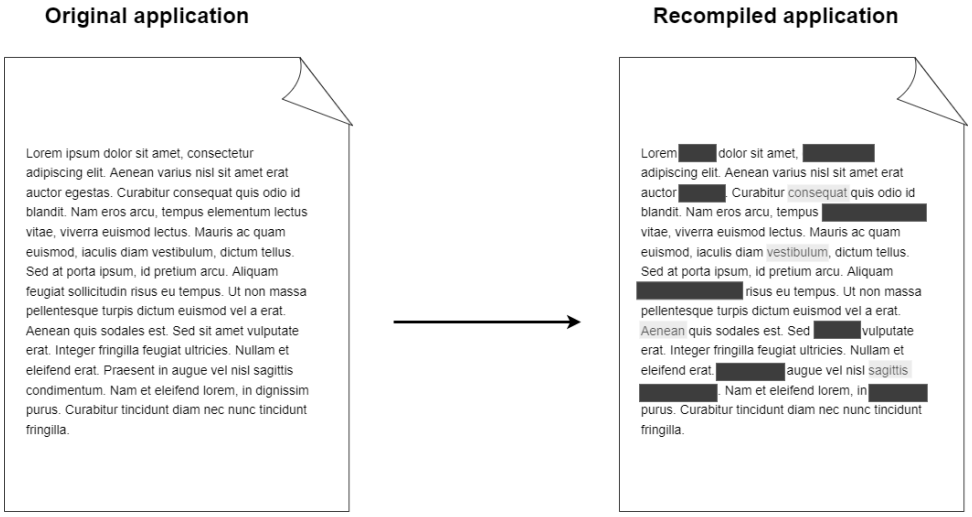
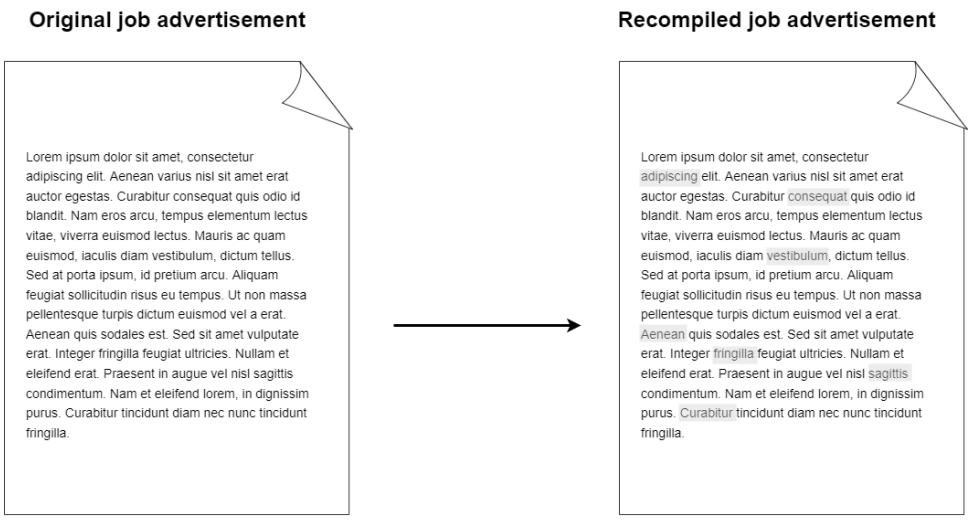


Figure 1: Rich picture showing the LA and SA in a domain context. It also becomes apparent how the communication between A and E is decoupled, since the recruitment platform acts as a mediator between the two actors. The grey areas highlighting the LA and SA is what this report revolves around, and the remaining parts are part of the domain context.

Zooming in on the conception of the recompiled documents (application and job advertisement), figure 2 shows a mock-up of how a recompiled application and job advertisement could look like. For subfigure 2a, demographic characteristics are censored and gendered wording is highlighted, whereas for subfigure 2b only gendered wording is highlighted.



(a) Recompilation of an original application written by A to a recompiled version, where demographic characteristics are censored and gendered wording is highlighted in light grey.



(b) Recompilation of an original job advertisement by E to a recompiled version, where gendered wording is highlighted in light grey.

Figure 2: Sketch of how recompilation of a job application and job advertisement could look like in a fully developed system.

For the domain context stated, the LA and SA act as backbone for the process handling. On a job recruitment platform, the process of uploading a job application → recompiling the application → editing the application → recompiling the application could be an iterative process, that allows A to give feedback on the censored and highlighted words.

3.2 Scope

Now that the applicability of the LA and SA have been established in *3.1 Concept description in a domain context*, the scope of the project can be delimited to realistic objectives. Due to the time frame of this project, this report will from now on focus solely on the proof of concept LA and SA, and for that reason the domain context depicted in figure 1 is not included. Therefore, this section is dedicated to state the scope of the LA and SA.

The scope of the LA and SA can be outlined as:

LA: Perform natural language processing, **NLP**, to conduct a linguistic analysis of sequences of words to classify words as named entities, **NE**'s (further described in *4.1.1 Named entity recognition*).

SA: Perform NLP to conduct a sentiment analysis of sequences of words to classify sentences/words based on gender-biased polarization.

Both analyses will contain implementations of several neural networks, with respect to their individual NLP task, using supervised learning to predict a classification of their input. The requirements of each analysis are unfolded in *3.3 Requirement specification*.

3.3 Requirement specification

Functional and non-functional requirements for the LA and SA are ranked in a MoSCoW analysis, where the fully disclosed version can be found in *Annex - 3.3.1 MoSCoW analysis*. In table 3 an extraction of the full size MoSCoW analysis is shown, where the most important requirements are stated.

Must	
1	The LA must be able to take a sequence of words as input, and output a sequence of predicted NE's.
2	The SA must be able to take a sequence of words as input, and output a binary sentiment classification based on the entire sequence.
3	The SA must be able to take a sequence of words as input, and output a multi sentiment classification based on each word in given sequence.
4	The SA using multi classification must be able to output a total sentiment value representing the gender bias of the entire sequence of words.
8	The SA must implement the concept of Explained AI (**XAI**) to describe to the end user, what the classification was based on.
Should	
1	The LA should have a recall score metric for each named entity of at least 75%.
2	The SA should have a precision score metric of at least 70%.
Won't	
1	The LA won't be trained on a dataset of existing job applications.
2	The SA won't be trained on a dataset of existing job applications.

Table 3: Extraction of the most important MoSCoW requirements for the LA and SA. The remaining functional and non-functional requirements are to be found in *Annex - 3.3.1 MoSCoW analysis*. In contrast to black box in ML, XAI is an implementation used to understand why an ML algorithm arrives at its conclusions, by making it explain itself. E.g a sentence is predicted to be heavily masculine, XAI will then present the most masculine words in the sentence, thereby explaining its prediction.

The trade-off between precision and recall is important for both analyses, since each metric plays a specific role in the outcome of the predictions. The LA requires good-scoring recall measurements of each classification of a NE, since it remains important as for the job recruitment context, that as many words representing demographic characteristics and socioeconomic status are recognized (respecting *Key point 1.*), and that the margin of error remains as low as possible. The trade off is then precision, which is a compromise of the high recall, It is by then accepted that rather censoring one too many words than giving away demographic characteristics and socioeconomic status.

Contrary to the LA, the SA requires good-scoring precision measurements, since it remains important as for the job recruitment context, that rather one too few gender-biased sentences/words are recognized (respecting *Key point 2.*), but with the pay off of being accurate on the most polarized sentences/words.

One of the most important requirements, which complicates how applicable the LA and SA are in the original domain context, is that the analyses won't operate on datasets comprised of job applications. This choise and its implications will be accounted for in *Annex - 4.2 Datasets* under *Availability* for both the LA and SA.

4 Analysis

Based on the requirements outlined in *3.3 Requirement specification* and the scope of the project defined in *3.2 Scope*, this section will cover a selection of the most important analyses for the comprehension and development of the project, drafted from *Annex - 4 Analysis*. Firstly linguistic features are analyzed which both form the basis for relevant features during training, but also is a comprehensive guide to better understanding of some of the linguistic terminology used throughout the report. Secondly, a four-step analysis of datasets for both the LA and SA is unfolded, where selected sections are picked out from the Annex report. Different neural network architectures are also explored with regard to the requirements stated in the MoSCow analysis in *Annex - 3.3.1 MoSCoW analysis*. At last, a small study of public demand is presented.

Analyses of NLP tools and machine learning frameworks are omitted from this report, but can be found in *Annex - 4.3 Open-source NLP libraries* and *Annex - 4.4 Machine learning frameworks*.

4.1 Linguistic features

This section will serve as an introduction and analysis to the linguistic features *named entity recognition*, *part-of-speech-tagging*, *lemmatization* and *stop words*. Named entity recognition is pivotal for the LA since it can assist in recognizing demographic characteristics. Part-of-speech tagging could prove useful for both the LA and SA, since it gives away a pattern in sentence construction, which could be beneficial during the training phase of the neural networks. Lemmatization and stop words are especially important to the SA, since they help reducing textual noise and to transform text down to a base form, on which sentiment can be extracted from.

4.1.1 Named entity recognition

Named entity recognition, **NER**, is an annotation of words as entities describing the words as the object they represent such as time, organization, person, money etc. There is no specific set of entities, as is therefore context dependent. In table 4, an example is given of NER.

Thousands	of	demonstrators	have	marched	through	London	to	protest	the	war	in	Iraq
O	O	O	O	O	O	B-geo	O	O	O	O	O	B-geo
CARDINAL						GPE						GPE

Table 4: Example of named entity recognition, NER, with the **IOB** scheme used in the second row, whereas the words are just annotated with labels in the last row, which is the practice *SpaCy* does, mentioned in *Annex - 4.3 Open-source NLP libraries*. The different schemes used for NE tagging is described in *Annex - 4.1.1 Named entity recognition*.

4.1.2 Part-of-speech tagging

Part-of-speech tagging, **POS** tagging, is the process of marking up a word as either a noun, adjective, verb, etc [19]. The same word can have different POS tags depending on the context of the sentence, therefore, this feature is important to distinguish between different meanings of a single word. In table 5 an example is given of POS tagging.

Thousands	of	demonstrators	have	marched	through	London	to	protest	the	war	in	Iraq
NNS	IN	NNS	VBP	VBN	IN	NNP	TO	VB	DT	NN	IN	NNP
NOUN	ADP	NOUN	AUX	VERB	ADP	PROPN	PART	VERB	DET	NOUN	ADP	PROPN

Table 5: Example of part-of-speech tagging, POS tagging, with both the *Penn Treebank* phrase structure in the second row and the universal phrase structure in the last row. The different schemes for POS-tagging are described in *Annex - 4.1.2 Part-of-speech tagging*. POS tagging proves great syntactical value, and is therefore an optional input for the LA in addition with the NER, whereas POS will be used for the SA.

4.1.3 Lemmatization

Lemmatization is a calculated process to return a word's base/dictionary form by removing inflectional endings such as *"-ed"* on *worked* to return the base form, *work*, which is known as the lemma [21]. In table 6, an example is given of lemmatization.

Thousands	of	demonstrators	have	marched	through	London	to	protest	the	war	in	Iraq
thousand	of	demonstrator	have	march	through	London	to	protest	the	war	in	Iraq

Table 6: Example of lemmatization.

Lemmatization is beneficial for the SA, and information retrieval in general, since it allows the algorithm to know that *is* and *be* are the same thing [22]. Lemmatization requires POS-tagging to work, and often leads to bad results in other languages than English.

Another similar syntactical feature is *stemming*, which also reduces words, but "guesses" where to chop off words, typically rule-based. It reduces words to an equal or shorter form, and is the quicker, but not as accurate, way of reducing words in a **corpus**. For instance *"University"* is reduced to *"Universe"*, even though the two words are not correlated [23].

Lemmatization will be used for the SA, since it provides more accurate results, which is prioritized. The trade-off is it requires more computational power.

4.1.4 Stop words

Stop words are words that not always provide syntactical value when performing NLP. Typical stop words are *"and"* or *"I"* and so on.

Thousands	of	demonstrators	have	marched	through	London	to	protest	the	war	in	Iraq
	x		x		x		x		x		x	

Table 7: Example of stop words, where the stop words are marked with *x*.

Stop words are typically removed in information retrieval and sentiment analyses [24], which also is going to be the case for the SA of this report, as mentioned in *4.2.2.1 Considerations* for the SA.

4.2 Datasets

Now that some common linguistic terminology is accounted for, and possible syntactical features have been selected for the LA and SA, datasets can be analyzed and selected for each analysis. From the Annex report, particular sections are picked out, such as *considerations* and *findings*, whereas *requirements* and *availability* are omitted.

4.2.1 Dataset for linguistic analysis

This section covers considerations and findings for the dataset of the LA. The omitted sections regarding requirements and availability can be found in *Annex - 4.2.1.1 Requirements* and *Annex - 4.2.1.3 Availability* respectively.

4.2.1.1 Considerations

The ideal dataset would be a **corpus** of real-world job applications in English, labelled with named entities and POS-tags for each word and punctuation occurring, since it would allow to operate in the intended domain context. The named entities would then be relevant for the context, and could therefore be categorized demographic features such as person, age, organization, ethnicity, religion and so on.

As an alternative, if there was a corpus of unlabelled real world job applications in English, then it could be possible to use a NLP library, such as those mentioned in *Annex - 4.3 Open-source NLP libraries*, to label words with their respective POS tag and NE tag (those provided by the library). This would carry along some issues, since using another trained **ML** algorithm to label a dataset would mean whatever bias and error margin that ML algorithm would have, would be passed on to this dataset.

These consideration are all dependent on the availability of datasets in the public domain, which is analyzed in *Annex - 4.2.1.3 Availability*.

4.2.1.2 Findings

Based on the analysis of availability in *Annex - 4.2.1.3 Availability*, a dataset based on public domain texts was found, generated by *Groningen Meaning Bank*, **GMB**. The dataset provided by GMB consists of 1,354,149 words, all annotated with a named entity. Furthermore, the dataset consists of 22 additional features such as descriptive, syntactical and contextual information of the word and its surroundings. In figure 3, a preview of the first 10 samples is given with selected features such as POS tag, NE tag, the word itself and the word coming before and after.

lemma	next-word	pos	prev-word	shape	word	tag
thousand	of	NNS	__START1__	capitalized	Thousands	O
of	demonstrators	IN	Thousands	lowercase	of	O
demonstr	have	NNS	of	lowercase	demonstrators	O
have	marched	VBP	demonstrators	lowercase	have	O
march	through	VBN	have	lowercase	marched	O
through	London	IN	marched	lowercase	through	O
london	to	NNP	through	capitalized	London	B-geo
to	protest	TO	London	lowercase	to	O
protest	the	VB	to	lowercase	protest	O
the	war	DT	protest	lowercase	the	O

Figure 3: Preview of first 10 samples in the dataset provided by Groningen Meaning Bank, with seven out of 25 total features selected.

The dataset is avaiable on the public forum *Kaggle.com*[27]. The NE's for the dataset, also referenced as *tags* in the dataset, are:

- *geo* - Geographical Entity
- *org* - Organization
- *per* - Person
- *gpe* - Geopolitical Entity
- *tim* - Time
- *art* - Artifact
- *eve* - Event
- *nat* - Natural Phenomenon

The remainder of words are annotated *O*, which is just a padding.

4.2.2 Dataset for sentiment analysis

This section covers considerations and findings for the dataset of the SA. The omitted sections regarding requirements and availability can be found in *Annex - 4.2.2.1 Requirements* and *Annex - 4.2.2.3 Availability* respectively.

4.2.2.1 Considerations

As per *4.2.1.1 Considerations* for LA, the ideal dataset would be a corpus of real-world job applications in English, and in regard to sentiment analysis, ideally it should be labelled with gender bias or polarization, meaning binary labels such as male or female, or a polarity ranging between -1 and 1. Ultimately that can be a hard task, due to GDPR restricting job applications availability, since they contain personal information. Hence any other dataset with texts or sentences labelled with gender by author of each sample would do for a prototype. Another way to approach the problem would be using seed words, allowing to determine the gender polarity of a sentence based on selected seed words, which is further described in *Annex - 4.2.2.1 Considerations*.

As in any machine learning algorithm, noise only introduces confusion and greater loss. In sentiment analysis and information retrieval in general, noise primarily stems from:

- URLs, email addresses, HTML tags, numbers and multiple spaces. None of these add any meaningful value or context [31] [32].
- Punctuation signs, which confuse a model to not recognise "sure" and "sure!" as the same thing [32] [33].
- Stop-words, which are extremely common words of little value, in English it could be "a", "by", "it", etc. [24]

4.2.2.2 Findings

Based on the analysis of availability in *Annex - 4.2.2.3 Availability*, the chosen public dataset is *News Category Dataset* by *Rishabh Misra* [34], where the first 10 samples are depicted in figure 4.

	category	headline	authors	link	short_description	date
0	CRIME	There Were 2 Mas...	Melissa Jeltsen	https://www.huff...	She left her hus...	2018-05-26
1	ENTERTAINMENT	Will Smith Joins...	Andy McDonald	https://www.huff...	Of course it has...	2018-05-26
2	ENTERTAINMENT	Hugh Grant Marri...	Ron Dicker	https://www.huff...	The actor and hi...	2018-05-26
3	ENTERTAINMENT	Jim Carrey Blast...	Ron Dicker	https://www.huff...	The actor gives ...	2018-05-26
4	ENTERTAINMENT	Julianna Marguli...	Ron Dicker	https://www.huff...	The "Dietland" a...	2018-05-26
5	ENTERTAINMENT	Morgan Freeman '...	Ron Dicker	https://www.huff...	"It is not right...	2018-05-26
6	ENTERTAINMENT	Donald Trump Is ...	Ron Dicker	https://www.huff...	It's catchy, all...	2018-05-26
7	ENTERTAINMENT	What To Watch On...	Todd Van Luling	https://www.huff...	There's a great ...	2018-05-26
8	ENTERTAINMENT	Mike Myers Revea...	Andy McDonald	https://www.huff...	Myer's kids may ...	2018-05-26
9	ENTERTAINMENT	What To Watch On...	Todd Van Luling	https://www.huff...	You're getting a...	2018-05-26

Figure 4: Head of *News Category Dataset. The dataset consists of 200,853 samples with six features; category, headline, authors, link, short description and date. Category and link being the only relevant features for this analysis. It is is worth at least 200,853 sentences, using short_ description, which is short summaries consisting of one or more sentences.*

The dataset is plotted in figure 5 to show the distribution of categories. Considering frequency of each category, the level of nuanced language and expected biased wording, *SPORTS*, *MONEY* and *BUSINESS* have been chosen to represent male gendered phrasing and wording. *WOMEN* and *STYLE & BEAUTY* have been chosen to represent female gendered phrasing and wording. These choices are made, because it's presumed that the target audience is the expected gender. No scientific or theoretical theory validates this presumption, it is simply chosen as a first iteration.

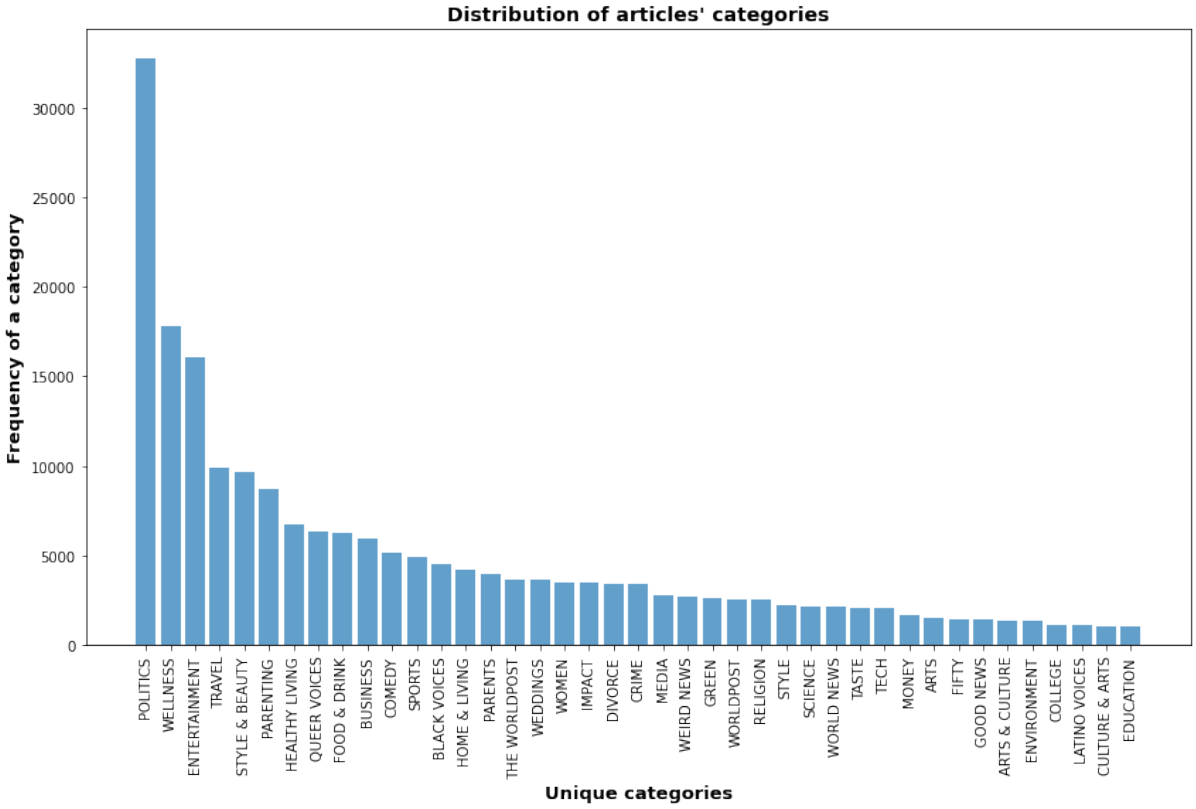


Figure 5: Categories found in *News Category Dataset* column *'categories'*. Since it is chosen that a given category is representative of gender, the dataset can be expanded beyond just short summaries, by web scraping each article for its body text. Then to create a clean dataset, a trained NLP pipeline provided by spaCy (further described in *Annex - 4.3 Open-source NLP libraries*), can be used for data augmentation, which uses NLP to breakdown entire article's texts into sentences, and sentences into words.

The approach of finding labels and choosing representative categories will be discussed further in *4.4.1 Ethical dilemmas and concerns* and *8.2 SA: Discussion*. For now the labels are accepted as the truth and results in 12,528 and 13,139 articles for male and female respectively.

4.3 Neural network architectures

For NLP tasks such as NER and sentiment analysis, some neural network architectures are more compatible than others. Especially recurrent neural networks, **RNN**'s, are common for NLP tasks, since they among others are strong on gaining contextual awareness and recognizing patterns in time sequential data. Additionally, convolutional neural networks, **CNN**'s, have proven great success with NLP tasks, despite being mostly known for image classification. Both architectures are discussed in the following sections.

4.3.1 Recurrent neural networks

Compared to common feed forward neural networks, RNN's have the same feed forward mechanism, but then backpropagates through the network, computing the gradient of the loss function and adjusts the weights one layer at a time. Since RNN's compute on sequential data, the architecture has the advantage of keeping state of previous input. When training a RNN at an arbitrary input at *t*, the network has knowledge about previous state at *t-1*. But the greater the time sequential data becomes, the harder it becomes for the network to recognize long-term dependencies. When the sequential data reaches a point where it is too long, RNN's can suffer from vanishing gradients, meaning the gradient becomes so small that the weights won't change during backpropagation, which would be a general issue for LA.

Fortunately, long-short-term-memory, **LSTM**, networks are able to withhold these long-term dependencies and also stamp out the vanishing gradient problem. LSTM is an extension of RNN, where additional units have been added to the original RNN. An LSTM cell is depicted in figure 6.

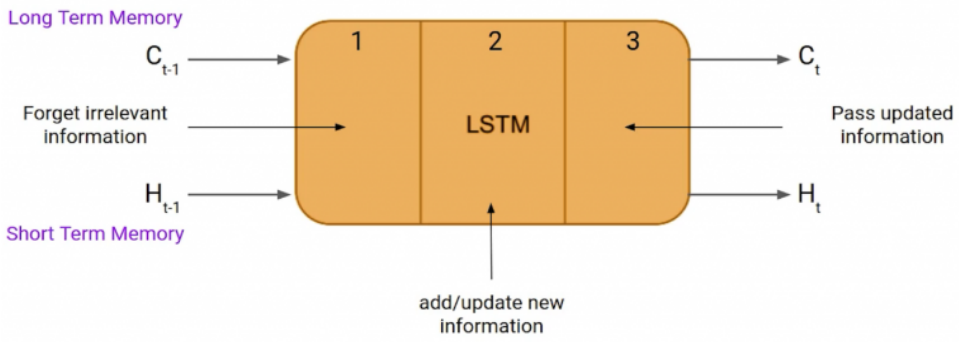


Figure 6: Overview of the of LSTM cell, where division 1 is the forget gate, division 2 is the input gate and division 3 is the output gate. Additionally, the cell has the long-term-memory C_{t-1}, also known as the cell state, and it has the short-term-memory h_{t-1}, also known as the hidden state. Source: *www.analyticsvidhya.com* [35].

Breaking down the LSTM cell in figure 7, it becomes visible how the three gates are connected and operating. The forget gate consists of the cell state C_{t-1} and hidden state h_{t-1}, the input gate consists of the input X_t and the output gate consists of the updated cell state C_t and hidden state h_t which also reflects the output.

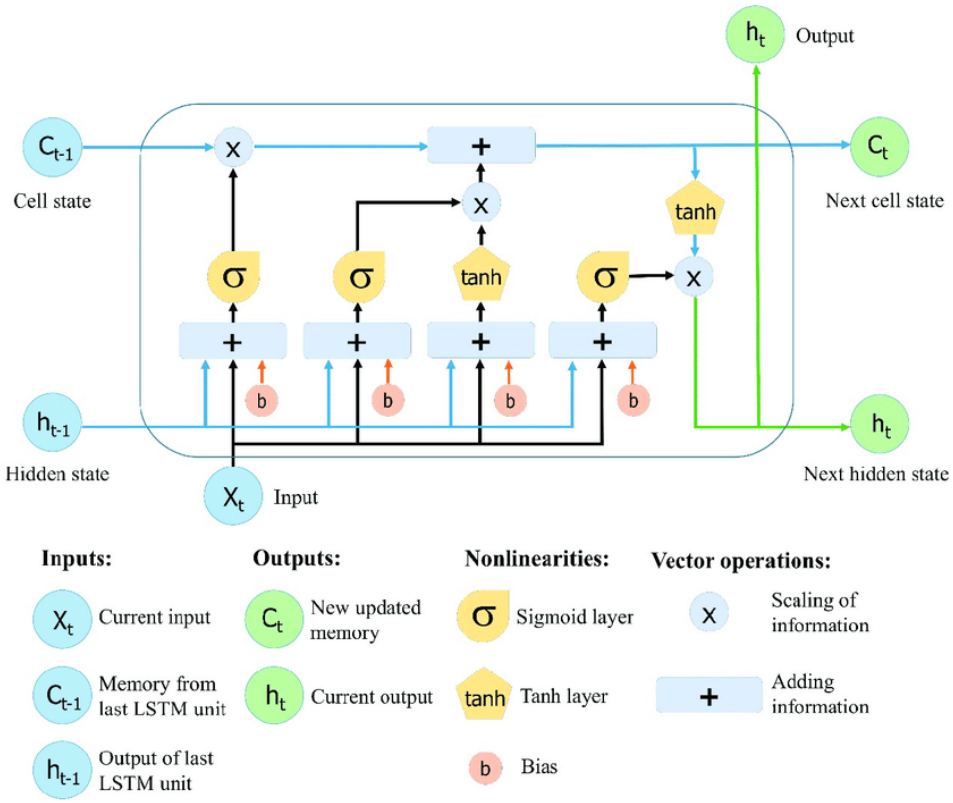


Figure 7: Structure of LSTM cell showing how information is persisted with the hidden state and cell state. The cell state C_{t-1} and hidden state h_{t-1} make up the forget gate, the input X_t makes up the input gate and the output h_t, next cell state C_t and next hidden state h_t make up the output gate. The forget gate decides what information from h_{t-1} and X_t to persist, where it applies a sigmoid function and results in a number between 0 and 1, where 0 is that it forgets everything from the previous timestamp and 1 it persists everything from the previous timestamp. The output from the sigmoid function is multiplied with the cell state C_{t-1} [35][36]. The input gate decides what information will be stored as the new cell state C_t, by multiplying the output from a sigmoid function and tanh function for h_{t-1} and X_t [36]. Lastly, the output gate decides what information is outputted and saved as the new hidden state h_t, which is the new cell state C_t through a tanh function multiplied with h_{t-1} and X_t through a sigmoid function [36]. Source: *www.researchgate.net* [37].

However, LSTM networks only have long-term dependencies from the past, whereas it could be favourable to also know about the future. For instance, taking an arbitrary sentence from the dataset at *t*, the network would have knowledge about previous state *t-1*, but it could be beneficial for the prediction if the network also had knowledge about future state *t+1*. This is allowed by bidirectional LSTM, **biLSTM**.

Opposite to LSTM, which flows in only one direction and has knowledge of only previous state, biLSTM flows in both directions; past to future and future to past. This means the network will have knowledge of both previous and future state, which is beneficial when computing on sentences where context could be given in both the start and the end of a sentence, which especially is the case for the LA, when it comes to recognizing NE's that are dependent on sentence construction from start to finish.

4.3.2 Convolutional neural networks

Convolutional neural networks are commonly used in multi-dimensional image classification problems to detect complex features, but have also proven strong in detecting complex features in two-dimensional text classification problems. CNN's consist in its simplicity of a convolution layer and a pooling layer. The convolution layer consists of a kernel that shifts over the input with its receptive field, computing a feature map by detecting the most prominent features within the receptive field[38]. The pooling layer

then subsamples the feature map by performing an operation on a receptive field of the feature map step by step and lastly outputs a matrix of fixed size, which is a necessity for classification. If a $n \times m$ matrix is considered in a classification problem, then each row represent a vectorized word, also considered a word embedding as depicted in figure 8[39].

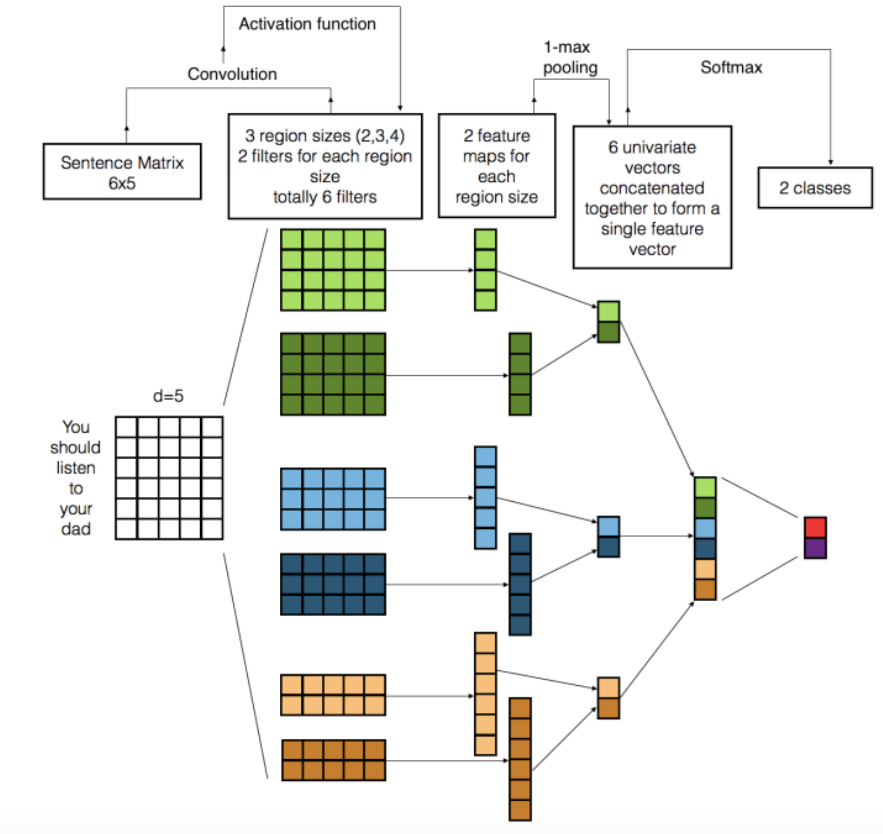


Figure 8: CNN layer handling of a text classification task. The sentence matrix contains every row representing an embedding (a vector) of a word. The embeddings are of the dimensionality of five, meaning every vector representing a word has five elements. Source: `https://arxiv.org/pdf/1703.03091.pdf`.

So for figure 8, each word is embedded with a dimensionality of five, meaning that each word has been transformed into a vector of fixed size five representing the respective word (further explained in *Annex - 4.5.2 Convolutional nerual networks*). In this case, there are six filters resulting in six feature maps, since each filter outputs a feature map. A max-pooling layer is applied, meaning the element with the greatest weight of each feature map is extracted and concatenated into a single feature vector on which a softmax function is applied to give a probability distribution of two classes. Figure 8 is a specific implementation of CNN, but gives an idea of how CNN can be applied to NLP tasks.

4.3.3 Comparison

Looking at *4.3.1 Recurrent neural networks* and *4.3.2 Convolutional neural networks*, LSTM and bidirectional LSTM networks have the ability to process long sequences of data, such as sentences, and then utilize its long-term memory and short term memory to gain contextual awareness. This would provide great value for both the LA and SA, where the inputs are sequences of words. CNN networks will be explored as well for the LA, despite they normally are being used for image classification, also are able to identify complex patterns in sequence data. Either LSTM or BiLSTM with CNN could potentially also be used as a mixture.

4.4 Humanistic and ethical insight

In this section the project's intention, stated in *3.1 Concept description in a domain context*, is discussed in relation to humanistic and ethical considerations. Furthermore a survey is conducted to study the public's perception of the project's problem statement and whether the project's intended solution is of any interest.

4.4.1 Ethical dilemmas and concerns

The SA aims to predict gender bias in a binary form, masculine and female, knowing gender definitions increasingly are becoming ambiguous. Thereby already in setting the task, the project discriminates against people, who do not define themselves by the traditional binary gender roles [41]. Even more so, the gender is classified by a news category's presumed target audience's gender. The presumption is not made on any theoretical basis, but on intuition by the project's authors, who are two cisgender white males, which is highly probable to introduce bias.

The project's intention, stated in *3.1 Concept description in a domain context*, is to reduce or completely remove possible discrimination between A and E, but in straight pursuing an solution without further humanistic or ethical considerations, contradictory the project itself may be discriminatory and contain a lot of bias.

4.4.2 Study of public demand

The project aims to help both job A's and E's, by making the early stages of the job recruitment process more anonymous, allowing A's to be evaluated and E's to evaluate job candidates without any prejudice based on their demographic characteristics, socioeconomic status. Two questionnaire surveys have been developed, one to study A's perception of the problem and proposed solution, and vice versa for E. Due to the project group's inexperience with questionnaires and the liberal arts in general, the questionnaires were developed in collaboration with three cand. mag. students of Media Studies.
The questionnaires consists of simple 'yes'/'no'/'don't know' questions for simplicity's sake. Below the selected questions and answers of highest interest are presented, but all questions and answers can be found in *10.3 Appendix C*.

	Questions	Yes	No	Don't know
1	Would you let your application be run through an algorithm, which censors your personal information (gender, age, ethnicity, race and religion), to not be picked/discarded based on it, but contrarily your competences and experience?	9	1	0
4	Would you let your application be run through an algorithm, which makes you aware of words with a high gender bias, which could give away your gender?	2	4	4
8	Would you be more prone to send an application for a job, in which the job advertisement uses a gender neutral language without gender biased words?	5	2	3

Table 8: 3 out of 8 of the questions articulated for A's, which students around the campus were chosen as, since it is highly probable that they are job seeking at the moment, recent past or the near future. The survey group consisted of 5 males and 5 females, and was primarily white. It is evident that the majority are willing to use a solution, such as the LA aims to provides, to be evaluated solely on competences and experience. While there doesn't seem to be a majority interested in using a solution such as SA, to help make an application more gender neutral, it seems from A's viewpoint, that E would benefit from applying the SA while writing job advertisements.

	Questions	Yes	No	Don't know
1	Is the applicant's gender a part of your considerations in relation to the screening process?	3	7	0
4	Would you utilize an algorithm to hide an applicant's personal information, such as gender, ethnicity, race, age and religion?	6	4	0
6	Do you think that an applicant's gender, ethnicity, race or religion may influence your choice subconsciously?	6	3	1
14	Would you be willing to make your job advertisement more gender neutral, if you were made aware that it contains gender biased wording?	9	1	0

Table 9: 3 out of 14 of the questions articulated for job employers, which the teachers in the AU building Edison were chosen as, since they seem probable to have the age and experience, which job recruiters often has. The survey group consisted of 10 lecturers, who were primarily white males. The majority agrees to not let gender influence their decision when screening candidates, but also agrees they think that demographic characteristics, socioeconomic status or gender may influence their decision subconsciously. The same partition of the group agrees to utilize an algorithm to hide such information. Lastly, there is an overwhelming amount willing to make their job advertisement more gender neutral, if a solution such as SA, was to make them aware of gender biased wording.

5 Process

This section will shortly go through how the process procedure was carried out for the project, whereas the full disclosure can be found in *10.4 Appendix D*.

First of all, the group consisted of two students with close relation, so formalities such as binding co-operation agreements were left out. Instead, a verbal cooperation agreement were conducted prior to formation of the group, where expectations where matched and process conditions resolved. Additionally, since the group members are of green and yellow personality test [42], it was established that honest criticism of each other work was required to raise the academic level of the project.

Initially the group employed Scrum as an iterative process for development, where phases of the project were organized into sprints, to create an overview over the short-term timeline as well as the long-term timeline. As the project progressed, the Scrum methodology were slowly phased out, since it was admitted, that when working close together on a big assignment, the flow of communication was casual and continuous. Furthermore, since the project revolved around a more research-oriented approach rather than a system development approach, the technical communication is less required, especially with an eye to the fact that the project was split in two separate analyses, one for each group member.

The project was supervised by an associate professor of Aarhus University, with whom weekly meetings were arranged, to discuss the progression of the process. A retrospective meeting was conducted once during the last couple of weeks of the project, where the group members deliberated pros and cons about the process development.

6 Methodology

With the datasets for the LA and SA accounted for in *4.2 Datasets*, and the neural network architectures analyzed in *4.3 Neural network architectures*, most of the preliminary work is done prior to design of the neural networks for both the LA and SA and implementation hereof. This section covers for one thing the methodology used for feature engineering while preprocessing both datasets, especially the process of web scraping the articles from the dataset accounted for in *4.2.2.2 Findings*. By applying web sraping, a new complete dataset is generated for the SA with the use of a NLP library and statistical measures. Furthermore, this section also covers a how neural networks are implemented and a final evaluation of all methodological approaches and their validity.

The methodological approaches are supplemented with concrete examples from the preprocessing of the data and implementation of the neural networks, which also can be found in the Annex report under implementation for both the LA and SA.

6.1 Linguistic analysis: Methodology

This section covers the methodology used to process the dataset analyzed in *4.2.1 Dataset for linguistic analysis* and the methodological approach to gain the results of the LA in *7.1 Linguistic analysis: Results*. As mentioned, the methodological approaches are supplemented with concrete examples based on the feature engineering and neural network implementation from *Annex - 5 Linguistic analysis*, in which more in-depth explanations can be found.

6.1.1 Data collection

The dataset from GMB is based on public domain English texts, which has been processed by GMB itself, to annotate each word and punctuation within each text with a named entity. The dataset comes in a *.csv* format and is comma-separated. A summary of the dataset is given in *4.2.1.2 Findings*. A compiled version of the dataset was fetched from *Kaggle.com*[27] and no further initial preprocessing of the dataset was performed. As a start, the dataset was compressed to only consist of words, POS tags and NE tags. In figure 9 the distribution of named entities from the dataset is shown.

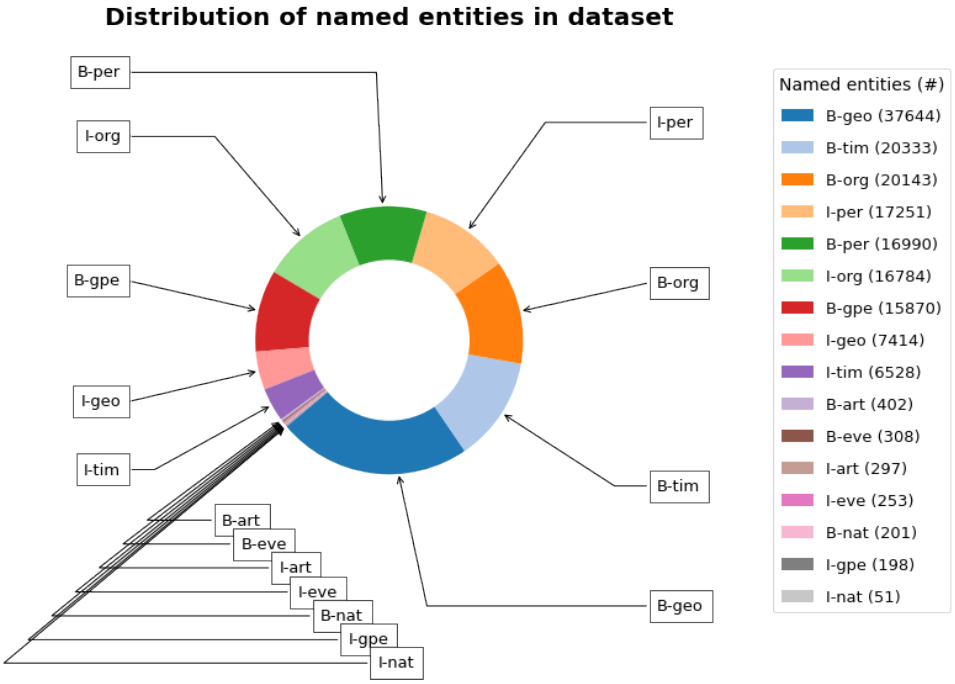


Figure 9: Distribution of named entities in dataset, with *O* tags being omitted since they account for 887,908 instances. Despite the dataset consisting of 1,354,149 words, some of the NE are only appearing a few hundred times and less, which is an expected occurrence, since a sentence consists of many stop words, which aren't assigned a named entity. Moreover, named entities such as *natural phenomenon (B-nat, I-nat)* is presumably more seasonal in articles, whereas *organizations (B-org, I-org)* are most likely mentioned on a daily basis. Therefore, the neural network won't have many occurrences of these NE's to train on. However, from *I-tim* and upwards, there is a strong representation.

6.1.2 Data factorization

Since the neural networks should be trained on sequences of words, as per requirement, the dataset was factorized into groups of sentences, so each new sample consisted of a sequence of words, POS tags and NE tags. The result of the concrete factorization from the feature engineering is shown in figure 10.

	Word	POS	Tag
0	[Thousands, of, demonstrators, have, marched, ...	[NNS, IN, NNS, VBP, VBN, IN, NNP, TO, VB, DT, ...	[O, O, O, O, O, O, B-geo, O, O, O, O, O, B-geo...
1	[Iranian, officials, say, they, expect, to, ge...	[JJ, NNS, VBP, PRP, VBP, TO, VB, NN, TO, JJ, J...	[B-gpe, O, O, O, O, O, O, O, O, O, O, O, O,...
2	[Helicopter, gunships, Saturday, pounded, mili...	[NN, NNS, NNP, VBD, JJ, NNS, IN, DT, NNP, JJ, ...	[O, O, B-tim, O, O, O, O, O, B-geo, O, O, O, O...
3	[They, left, after, a, tense, hour-long, stand...	[PRP, VBD, IN, DT, NN, JJ, NN, IN, NN, NNS, .]	[O, O, O, O, O, O, O, O, O, O, O]
4	[U.N., relief, coordinator, Jan, Egeland, said...	[NNP, NN, NN, NNP, NNP, VBD, NNP, ,, NNP, ,, J...	[B-geo, O, O, B-per, I-per, O, B-tim, O, B-geo...

Figure 10: First five samples of factorized LA dataset, grouped by sentences.

6.1.3 Partition of data

The distribution of length of all sentences were measured against their frequency to identify sentences with a particularly high word count. The distribution is shown in figure 11.

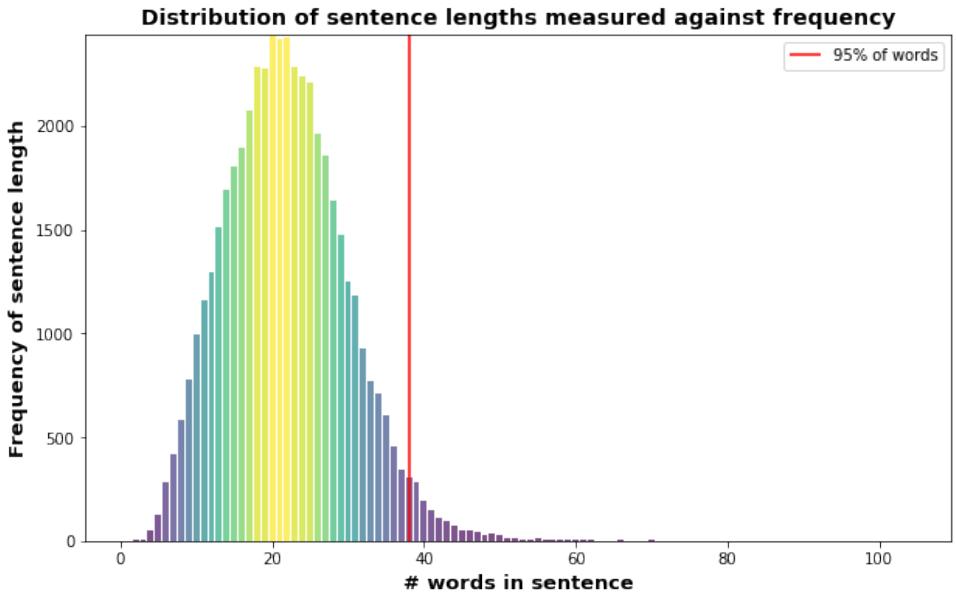


Figure 11: Distribution of sentence lengths measured against frequency with a decision boundary partitioning the sentences at a word and punctuation count at 38, discarding the remaining 5% of the dataset.

A decision boundary was set in the distribution, partitioning the distribution at 95% of the word accumulation for all sentences. This was done to discard the remaining 5% of sentences with the highest word count, thus reducing the required padding of all sentences in *6.1.4 Feature tokenization* and subsequently reducing training time. The target value of the decision boundary was based on an empirical estimate of the dataset with no specific theoretical argumentation.

6.1.4 Feature tokenization

Every sequence of words, POS-tags and NE-tags were tokenized as integers, so each sequence would consist of unique integer representations of its original value. The decision boundary was then used as a cap in combination with a padding process, so samples containing sequences with encoded elements greater than the boundary would be discarded and the remaining samples were padded with a padding token, so they all had the length of the decision boundary. An example of the tokenization process is shown in figure 12 and is further described in *Annex - 5.2.2 Word tokenization.*

['Iranian', 'officials', 'say', 'they', 'expect', 'to', 'get', 'access', 'to', 'sealed', 'sensitive', 'parts', 'of', 'the', 'plant', 'Wednesday', ',', 'after', 'an', 'IAEA', 'surveillance', 'system', 'begins', 'functioning', '.', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>']

⇕

[8194, 27728, 31034, 33290, 22578, 33465, 23724, 16666, 33465, 31143, 31320, 28268, 27701, 33247, 28647, 16053, 22, 16916, 17350, 7925, 32880, 32986, 18239, 23556, 25, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Figure 12: Example of mapping between a tokenized sequence and its original string value. It comes to show the padding with the padding token *<PAD>* is carried out. From the sequence it is also visible that two occurrences of "to" has the same unique encoding.

Finally, every sequence of NE-tags were one-hot encoded into binary class matrices, meaning a binary representation of each NE-tag in a sequence as a matrix as depicted in figure 10. This is further described in *Annex - 5.2.2 Word tokenization.*

Tag	Tokenized tag	Binary class matrix
O	0	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
B-art	1	[0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
I-art	2	[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
B-eve	3	[0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0]
I-eve	4	[0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0]
B-geo	5	[0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0]
I-geo	6	[0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0]
B-gpe	7	[0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0]
I-gpe	8	[0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0]
B-nat	9	[0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0]
I-nat	10	[0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0]
B-org	11	[0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0]
I-org	12	[0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0]
B-per	13	[0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0]
I-per	14	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0]
B-tim	15	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0]
I-tim	16	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]

Table 10: Visualization of tokenization and binary class matrix transformation of NE-tags, also known as one-hot encoding. The binary class matrix has the length of the sum of all NE tags, and each element is an index representation of a NE tag. Considering the NE tag *B-geo*, the tokenized tag value 5 is the index value in the binary class matrix, where there has to be a 1, meaning the matrix represents that specific NE tag.

6.1.5 Token embedding

Since the input sequences of words and POS-tags all were encoded and of equal length, they could be embedded. The embedding was the first hidden layer in each neural network, and creates a lookup table during training where the tokenized words map to vectorized tokens of a fixed dimensionality. Embeddings allow for the neural network to understand semantically similar words over time, and is build on the idea that similar words occur together more frequently than dissimilar words [45]. Semantic similarity between words is crucial for gaining contextual awareness, which is why token embedding is an important step prior to model composition.

6.1.6 Implementation

After above mentioned methodological approaches were carried out, only the input layer was changed between single-input and multi-input. Prerequisites were carried out prior to training such as implementing a custom learning rate scheduler with exponential decay and early stopping as well as deciding on a proper loss function. These subjects are further described in *Annex - 5.3.1 Prerequisites.*

At this point, LSTM, BiLSTM, CNN and BiLSTM CNN architectures were composed. The layer architecture of the multi-input BiLSTM CNN model can be seen in figure 13 with its layer configurations in table 11. An in-depth description of the composition is found in *Annex - 5.3 Implementation of neural networks.*

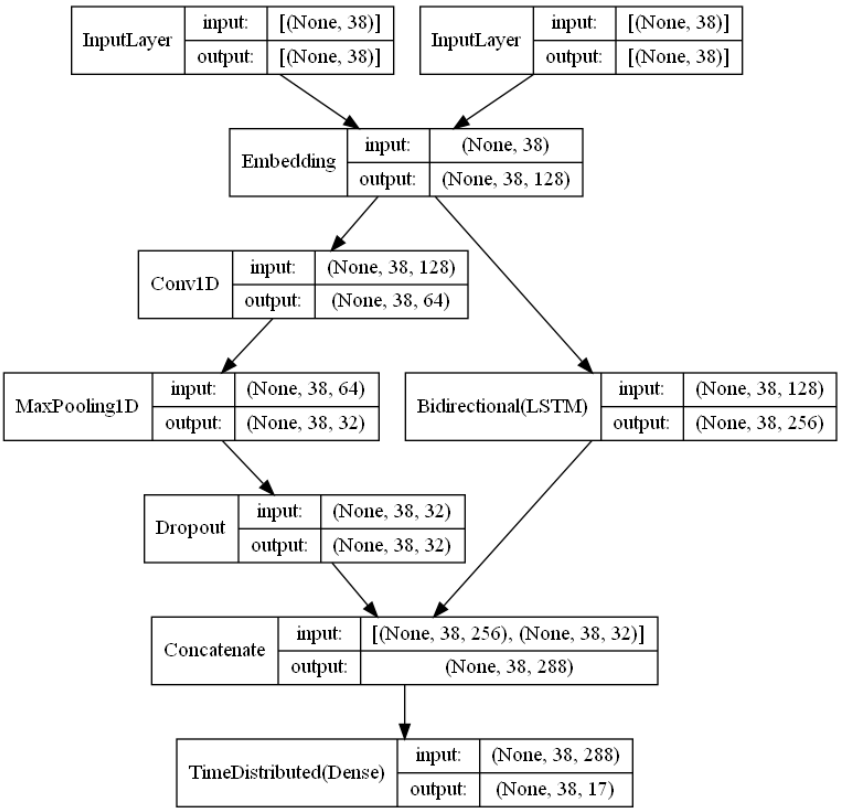


Figure 13: Layer architecture of a multi-input multi-label classification CNN BiLSTM. The two input layers are representing the tokenized words and POS tags, which are embedded separately by a shared embedding layer. From then on the POS tags are run through a one dimensional convolutional layer, followed by a pooling layer and at last a dropout layer, all the while words are running through a bidirectional layer wrapping a dense layer. At the end of each lane, the outputs are concatenated and run though a time distributed layer, wrapping a dense layer, that allows for continuous prediction of NE-tags for each combined word and POS tag. The input and output values represent the shapes for each specific layer, and how they are potentially transformed.

Architectural composition of BiLSTM		
Multi-input multi-label classification		
Conv1D layer:		
	Filters:	64
	Kernel size:	3
	Activation function:	Relu
	Padding:	same
MaxPooling1D layer:		
	Pool size:	2
	Data format:	channels first
Dropout layer:		
	Rate:	0.2
Bidirectional layer (LSTM):		
	Dropout:	0.2
	Recurrent dropout:	0.2
	Return sequences:	true
	Merge mode:	concat
	Output dimensionality:	256
	Activation function:	tanh
Concatenation layer:		
	Output dimensionality:	64
Time distributed layer (dense):		
	Activation function:	softmax

Table 11: Architectural composition of multi-input (figure 13) CNN BiLSTM. Displaying actively set hyper parameters, whereas many hyper parameters are default values for the layers provided by the used machine learning framework Keras, which is further described in *Annex - 4.4 Machine learning frameworks*.

The neural networks where benchmarked and compared with respect to score metrics and confusion matrices.

6.1.7 Evaluation

This methodology is a widespread approach for NER, but how data is preprocessed varies. Especially the tokenization approach could have some side effects, since every sequence is padded with a padding token. Another approach could also be adding a start-of-string token, **SOS**, and a end-of-string token, **EOS**. These could help the network to recognize patterns in the sentences, by learning a semantic start and stop value.

Both words and punctuations were tokenized and trained upon, so another approach could be to discard punctuation since these possibly could just create more complexity than advantage while training the neural networks.

When partitioning the data, the 5% of sentences with greatest word count were discarded. The reason was to reduce training time as well as cutting of specifically long sentences, which could be considered as 'outliers'. However, it could have been worth considering if sentences with a very low word count also should have been discarded, when talking about partitioning the data. A research into a standard distribution of sentence lengths in formally written text in English could have proven beneficial for this objective.

6.2 Sentiment analysis: Methodology

This section covers the methodology used to define the input and output of the sentiment ML algorithms, define and craft the sentiment corpus, feature engineer a new corpus and implement the ML algorithms. As mentioned, the methodological approaches are supplemented with concrete examples based on the data collection, feature engineering and neural network implementation from *Annex - 6 Sentiment analysis*, in which more in-depth explanations can be found. During crafting of the sentiment corpus, the NLP library spaCy will be used, which is accounted for in *Annex - 4.3 Open-source NLP libraries*.

6.2.1 Features and classification

Input text during training will be classified on two different levels, word and sentence level. On a sentence level the only input feature will be the sentence's text. On a word level the features will consist of the word, the word's lemma, sentence number, POS tag and polarity. These features are assessed to help provide a contextual understanding of each sentence, as found in *4.1 Linguistic features*.

Both levels will be classified on binary and multi level. The output of the binary classification is either masculine or feminine, a direct representation of gender label categories. The output of the multi classification is valued between -1 (masculine) and 1 (feminine), with 0 being neutral. The step size from -1 to 1 is 0.1, to limit the number of classes, resulting in 21 different classes.

This results in four different ML algorithms stated below:

Sentiment algorithm 1. Sequence of lemmas as single feature input and binary-classification output.

Sentiment algorithm 2. Sequence of lemmas as single feature input and multi-classification output.

Sentiment algorithm 3. Sequence of lemmas, sentence index, POS-tag and polarity as multiple feature input and binary output.

Sentiment algorithm 4. Sequence of lemmas, sentence index, POS-tag and polarity as multiple feature input and multi-classification output.

6.2.2 Data collection

As a start, each sample of the corpus *News Category Dataset* [34], contains an article link, category and other (irrelevant) key/value pairs. The goal for the end corpus, is to have as many noise-redued samples as possible, labeled with the gender of the of the targeted audience in accordance to categories chosen for each gender in *4.2.2.2 Findings*. The end corpus is defined as:

Sentiment corpus. Each sample consists of a sentence number, word, lemma, POS-tag, the lemma's polarity and gender.

The process to create the *Sentiment corpus* has been split into several steps, as seen in figure 14. In the following subsections the steps will be explained in a chronological order, according to the figure.

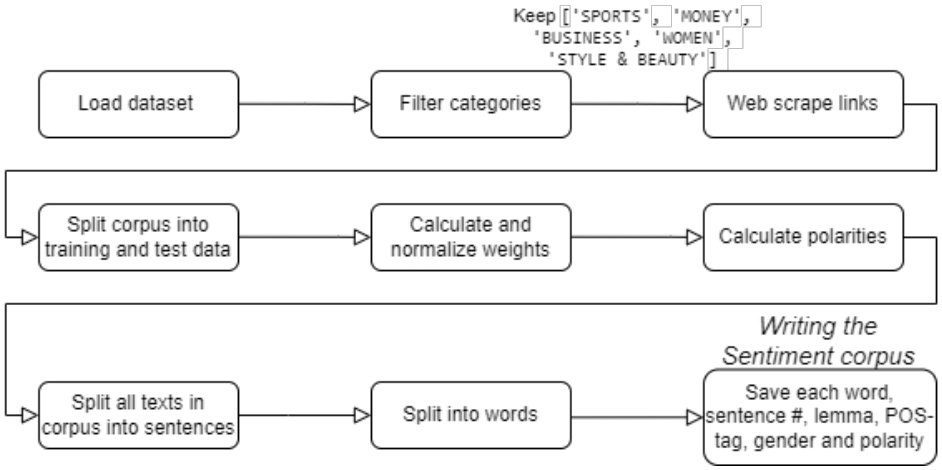


Figure 14: Visualizing the steps in methodology of creating the *Sentiment corpus*. Each step is a iteration of the corpus, meaning after each step the dataset is saved as it is, allowing each component to be interchangeable and decoupled of each other. In short, the methodology is to load the corpus with all articles, filter them to match the selected news categories and web scrape the links of the remaining articles to extract each article's body text, creating the foundation for the *Sentiment corpus* with news articles' body text paired with the presumed gender of the target audience. Afterwards it is split, 75% to training and 25% to test, a polarity dictionary is computed from the training data, and lastly the *Sentiment corpus* is written by splitting entire articles' texts into sentences, sentences into words, and saving each word with its sentence index, lemma, POS-tag, gender and polarity as a sample.

6.2.2.1 Load dataset

Loading the dataset is the first step in the methodology of creating the *Sentiment corpus*, described in figure 14, which an example is given of in figure 15.

	articles
0	{'category': 'CRIME', 'headline': 'There Were 2 Mass Shootings In Texas Last...
1	{'category': 'ENTERTAINMENT', 'headline': 'Will Smith Joins Diplo And Nicky ...
2	{'category': 'ENTERTAINMENT', 'headline': 'Hugh Grant Marries For The First ...
3	{'category': 'ENTERTAINMENT', 'headline': 'Jim Carrey Blasts 'Castrato' Adam...
4	{'category': 'ENTERTAINMENT', 'headline': 'Julianna Margulies Uses Donald Tr...
...	...
200848	{'category': 'TECH', 'headline': 'RIM CEO Thorsten Heins' 'Significant' Plan...
200849	{'category': 'SPORTS', 'headline': 'Maria Sharapova Stunned By Victoria Azar...
200850	{'category': 'SPORTS', 'headline': 'Giants Over Patriots, Jets Over Colts Am...
200851	{'category': 'SPORTS', 'headline': 'Aldon Smith Arrested: 49ers Linebacker B...
200852	{'category': 'SPORTS', 'headline': 'Dwight Howard Rips Teammates After Magic...

Figure 15: The base corpus, *News Category Dataset* [34], is loaded. Each sample contains a news category, headline, authors, article web link, short description and publishing date.

6.2.2.2 Filter categories

The second step in the methodology, is to filter all samples by their news category which is shown in figure 16.

	category	headline	authors	link	short_descrisption	date
0	WOMEN	Morgan Freeman D...	Mary Papenfuss	https://www.huff...	Both Visa and Va...	2018-05-25
1	WOMEN	The Joy Of Watch...	Emma Gray	https://www.huff...	There's a delici...	2018-05-25
2	WOMEN	The 20 Funniest ...	Hollis Miller	https://www.huff...	"Welcome to adul...	2018-05-25
3	WOMEN	Morgan Freeman A...	Sebastian Murdock	https://www.huff...	Eight people tol...	2018-05-24
4	SPORTS	Jets Chairman Ch...	Ron Dicker	https://www.huff...	"I never want to...	2018-05-24
...	...	...	...	...	...	...
24076	BUSINESS	Positive Custome...	Ernan Roman, Con...	https://www.huff...	"Analysts at Ado...	2012-01-28
24077	SPORTS	Maria Sharapova ...		https://www.huff...	Afterward, Azare...	2012-01-28
24078	SPORTS	Giants Over Patr...		https://www.huff...	Leading up to Su...	2012-01-28
24079	SPORTS	Aldon Smith Arre...		https://www.huff...	CORRECTION: An e...	2012-01-28
24080	SPORTS	Dwight Howard Ri...		https://www.huff...	The five-time al...	2012-01-28

Figure 16: The base corpus, shown in figure 15 is filtered to only keep categories of *SPORTS*, *MONEY*, *BUSINESS*, *WOMEN* or *STYLE & BEAUTY*, in accordance to to categories chosen for each gender in *4.2.2.2 Findings*. Each of the 24080 samples contain a short description of a given article with at least one sentence, which could be used to train the sentiment algorithms, but it can be expand by web scraping each link of each sample, which links to a given article's HTML site.

6.2.2.3 Web scrape links

Each sample in the corpus shown in figure 16 is processed where each sample's link is web scraped for the articles body text. The extracted body text of each sample and its gender label are appended to a new corpus, as seen in figure 17, which forms the basis for creating an entirely new dataset, the *Sentiment corpus*.

	text	gender
0	At least two organizations have deci...	F
1	The best way to watch Harvey Weinste...	F
2	The ladies of Twitter never fail to ...	F
3	Multiple women have accused actor Mo...	F
4	When it comes to New York Jets playe...	M
...	...	...
23306	The Challenge: How do you build a bu...	M
23307	By Steve Tignor, Tennis.com\n\nMELBO...	M
23308	Leading up to Super Bowl XLVI, the m...	M
23309	49ers rookie and rising star Aldon S...	M
23310	Judging by the comments that Magic c...	M

Figure 17: The extracted body text and gender label of each sample from the filtered corpus shown in figure 16. Web scraping allows to increase the sample size immensely, although 770 articles are discarded due to being textless, the resulting corpus contains 23310 body texts. The filtered dataset contains at least some sentences, since each sample has short description of at least one sentence describing the article. But as it will be discovered later, the current 23310 extracted body texts are worth 668.175 sentences.

At this point the dataset, shown in figure 17, is split in training and test data, the first 75% is used for training and the remaining 25% for test. The result is 17483 samples for training and 5828 samples for test.

6.2.2.4 Calculate weights

The gender-labels "M" and "F", are sufficient for binary classification, but something more nuanced is needed for multi classification, such as gender polarity on a continuous scale between masculine and feminine. The solution thought appropriate, is a dictionary of the most important words used by males and females, each word with a polarity somewhere between -1 (*masculine*) and 1 (*feminine*), with 0 being *neutral*. The weights and polarities are computed only from the partitioned training data, to prevent overfitting.

To determine the importance of words, *tf-idf weighting* is used, which combines *term frequency* and *inverse document frequency*. It is a statistical measure used to determine the mathematical significance of words in documents [49], and commonly used in search engines to filter documents from a query term and in word embedding [50]. Throughout this section, a mini-example corpus c, consisting of six made up short sentences, is used as example of an article's body text. Furthermore the corpus is split by gender and computed independently, resulting in two temporary corpus, the masculine corpus c_m and feminine corpus c_f, as seen in table 12. Throughout this section multiple tables will be shown, which all are inherited from the root table 12.

corpus c	
corpus c_m	corpus c_f
"John Johnson is stronger than strong Jane"	"Jane Smalls is prettier than pretty John"
"I am a very strong person"	"I am a very supporting person"
"I thrive in a competitive setting"	"I thrive in a nurturing setting"

Table 12: Base mini-corpus c with examples of gendered sentences. Left is categorized by masculine, right is categorized as feminine. The corpus is split by gender, resulting in two temporary corpus, corpus c_m containing masculine categorized articles and corpus c_f containing feminine categorized articles, which are computed independently.

Each article is referred to as a document d consisting of one or more words. Each d is preprocessed to first remove noise and secondly lemmatize each word (transforming a word to its base form), where each lemma is referred to as a term t.

Table 12

⇓

['be', 'strong', 'than', 'strong']	['be', 'pretty', 'than', 'pretty', 'John']
['I', 'be', 'a', 'very', 'strong', 'person']	['I', 'be', 'a', 'very', 'support', 'person']
['I', 'thrive', 'in', 'a', 'competitive', 'setting']	['I', 'thrive', 'in', 'a', 'nurture', 'set']

Table 13: The base-mini corpus from table 12 is preprocessed, by filtering each document and lemmatizing each word. Each document is processed by spaCy, used to differentiate all words from one another, identify stop words and named entities. The document is filtered by special characters and NE's of type *['TIME', 'DATE', 'GPE', 'CARDINAL', 'PERSON', 'MONEY', 'PERCENT']*, since they are thought to not be representative of any gender's language usage. Lastly, each word is lemmatized by spaCy, resulting in an all lowercase lemma. Occasionally words will not be lemmatized by spaCy, since lemmatization is designed with recall in mind [22], therefore qualitative lemmatization are prioritized above quantitative lemmatization. In such cases the word is still appended.

To determine a term's importance in a document, a mechanism is used to compute a score between a term t and document d, based on the weight of t in d. The chosen weighting scheme is quantitative digest, that assigns the weight for t based on number of occurrences of t in d, referred to as term frequency $tf_{t,d}$.

Table 13

⇓

[1, 2, 1, 2]	[1, 2, 1, 2, 1]
[1, 1, 1, 1, 1, 1]	[1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 1, 1]	[1, 1, 1, 1, 1, 1]

Table 14: Term frequency values computed of each term in each doc/sentence in the preprocessed corpus from table 13. Term frequency's way of perceiving a document is called a *bag of words model*, which refers that the order of terms is ignored [51].

Using $tf_{t,d}$ it can be assumed that stop words, such "a", "by", "it", etc., would have high scores, but does not contribute in any way to how men and women write. All terms in a document are not equally important, and stop words are not the most important, therefore a mechanism is needed to attenuate the effect of terms that appear too often across all documents in the corpus. Thus, inverse document frequency idf_t, which is the inverse of document frequency df_t, is defined to be the amount of documents in the collection that contains term t [52]. The rarer a term is, the higher idf_t. It is defined in equation (1).

$$idf_t = log(\frac{N}{df_t}) \tag{1}$$

N being the total number of documents in c_m or c_f and df_t the document frequency. When calculating idf_t each unique term from a corpus is computed, resulting in a dictionary as seen in table 15.

Table 13

⇓

Masculine idf_t dictionary		Feminine idf_t dictionary	
'than', 'very', 'person', 'in', 'thrive', 'setting', 'competitive'	1.79	'than', 'pretty', 'John', 'support', 'very', 'person', 'in', 'thrive', 'set', 'nurture'	1.79
'strong', 'be', 'I', 'a'	1.1	'be', 'I', 'a'	1.1

Table 15: Inverse document frequency values computed of each unique term across each corpus, c_m and c_f, from table 13. Equal values are grouped for simplicity, but are not affiliated in any way.

By combining the definitions term frequency and inverse document frequency, a composite weight can be computed for each term in each document, accounting for its significance in a given document. The tf-idf weighting scheme for term t in document d is given by equation (2) [53].

$$tf\text{-}idf_{t,d} = tf_{t,d} \cdot idf_t \tag{2}$$

When computing tf-$idf_{t,d}$ for such a small dataset as in table 15, where every term's $tf_{t,d}$ is 1, the result would equal to idf_t in table 15. Furthermore tf-$idf_{t,d}$ is highest when t occurs many times in a small number of documents, lower when the term occurs fewer times in a document or occurs in many documents, and lowest when the term occurs in all documents [53].

Generally when using $tf\text{-}idf$, it is used to find the significance of a term in a document, therefore denoted $tf\text{-}idf_{t,d}$. No publicly available equation to find the significance of a term in the whole corpus was be found, therefore equation (2) has been further developed into a self-developed equation. The self-developed equation computes the mean of each term's $tf\text{-}idf_{t,d}$, as seen on equation (3).

$$tf\text{-}idf_{t,c} = \frac{\sum_{n=1}^{|d|} tf_{t,d} \cdot idf_t}{N} \tag{3}$$

N being the total number of documents in c_m or c_f. The self-developed equation is implemented in an algorithm and used to compute a masculine and feminine $tf\text{-}idf_{t,c}$ dictionary. In figure 17, an example is given of an extraction of the actual computed $tf\text{-}idf_{t,c}$ dictionary for both masculine and feminine.

$tf\text{-}idf_{t,c}$ **dictionary**			
masculline		*feminine*	
company	1.28	woman	1.80
say	1.28	like	0.87
work	1.09	say	0.81
people	0.99	look	0.80
business	0.98	man	0.79
...		...	

Table 16: The five highest weighted lemmas for masculine and feminine $tf\text{-}idf_{t,c}$ dictionary, computed from the corpus shown in figure 17. As shown in figure 14, showing each step in the data collection methodology, the weights are only computed on the corpus' training data, to prevent overfitting.

The distribution of weights from the masculine and feminine $tf\text{-}idf_{t,c}$ dictionaries are plotted in figure 18.

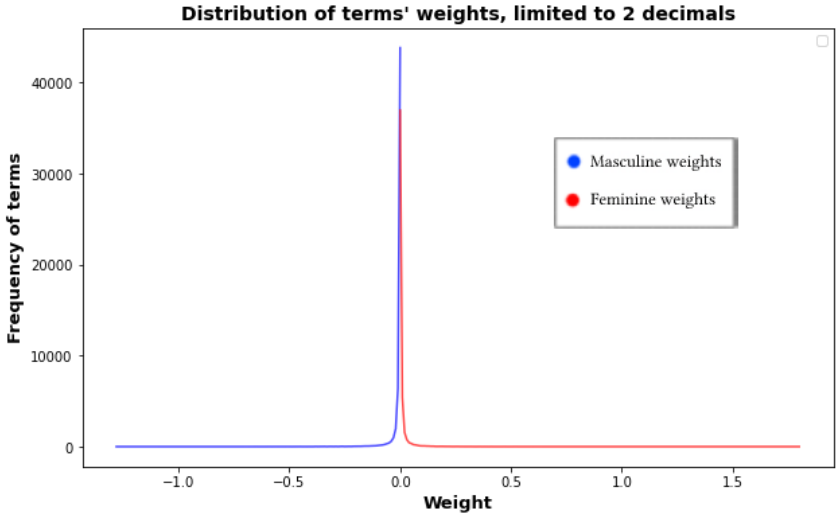


Figure 18: Distribution of weights, computed from the corpus shown in figure 17. The masculine weights are set as negative in order to easier compare the two dictionaries. Many terms of low significance will appears both in the masculine and feminine weight dictionaries close or equal to 0, therefore all weights equal or lower than 0.05 are cut off, since anything lower is thought to be noise. This results in 2260 masculine and 1505 feminine weights. To prevent bias, the length of the masculine and feminine dictionary is cut to match each other in length. All weights will subsequently be normalized, to span between 0 and 1. This allows them be used to compute the polarities of each term, since we want a values between -1 and 1. The general formula for normalizing arrays is used.

6.2.2.5 Calculate polarities

No publicly available equation to find a term's polarity on a continuous scale between -1 and 1, representing masculine and feminine respectively, could be found. To solve this issue, equation (4) has been self-developed and is of first iteration, that subtracts masculine weights from feminine weights, to compute a single polarity for a lemma.

$$polarity_{t,c} = tf\text{-}idf_{t,c_f} - tf\text{-}idf_{t,c_m} \tag{4}$$

Equation (4) is then used to compute the polarity dictionary for the whole corpus, based on the $tf\text{-}idf_{t,c}$ dictionaries found in figure 18.

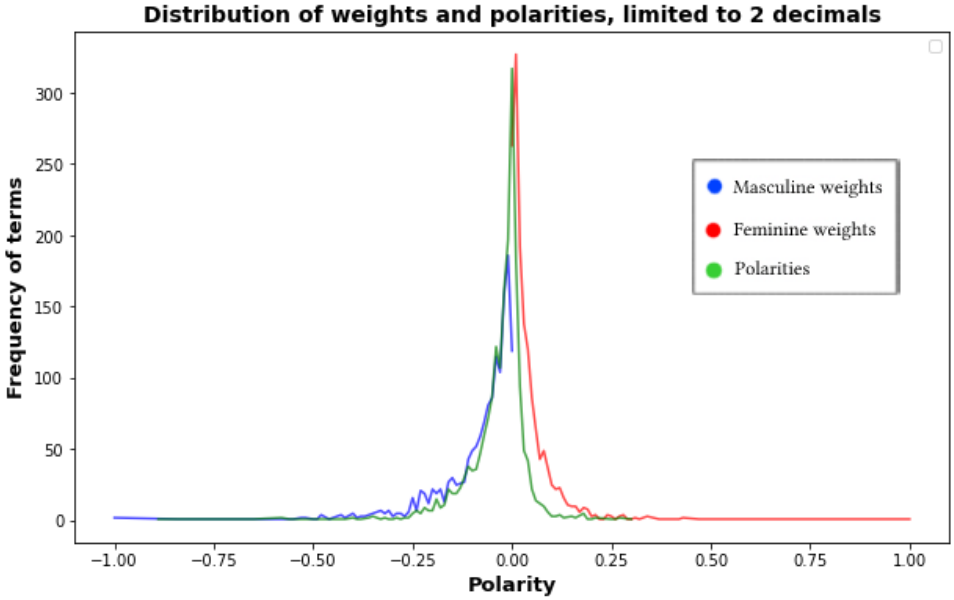


Figure 19: Distribution of normalized weights from figure 18, and the computed polarities using equation (4). It is evident that the weights are primarily below 0.25, which presumably is due to a few highly weighted lemmas in each dictionary, that when normalized makes the remaining polarities low. The highly weighted lemmas can't be discarded to get a better distributed dictionary, since it would be forgery of the data. The polarity is then computed by subtracting masculine from feminine weights, resulting in the polarities seen in the graph. It can be seen on the distribution of polarities, that no polarities go lower than -0.9 and higher than 0.3. Compared to the normalized weights, the span is decreased, due to highly significant words appearing on for both genders, equaling them out.

In table 17 the polarity dictionary's most masculine and feminine lemmas shown, which seem to be heavily influenced by the news categories chosen for each gender.

Polarity dictionary			
top masculline		*top feminine*	
company	-0.89	hair	0.30
business	-0.65	woman	0.27
team	-0.58	dress	0.27
game	-0.58	wear	0.26
say	-0.56	fashion	0.25
...		...	

Table 17: The 5 most masculine/feminine lemmas in the polarity dictionary, *6_polarity_dict_norm*.

6.2.2.6 Writing the *Sentiment corpus*

Lastly, it is all wrapped up into a final dataset, as shown in figure 14. The figure is created iterating each sample outputted from *6.2.2.3 Web scrape links*, with each sample containing a text and a gender, the text being the full body text scraped from the article link, and the gender being the presumed targeted audience.

	Sentence #	Word	Lemma	POS	Polarity	Gender
0	1	organizations	organization	NOUN	-0.223025	F
1	1	decided	decide	VERB	-0.039008	F
2	1	drop	drop	VERB	-0.058436	F
3	1	women	woman	NOUN	0.272143	F
4	1	accused	accuse	VERB	0.006282	F
...	...	...	...	...	...	...
4970462	668175	named	name	VERB	-0.030568	M
4970463	668175	options	option	NOUN	-0.051049	M
4970464	668175	potential	potential	ADJ	-0.096828	M
4970465	668175	landing	landing	NOUN	0.000000	M
4970466	668175	spot	spot	NOUN	-0.005818	M

Figure 20: The final *Sentiment corpus*, stretching 4.970.466 samples, each with a sentence index, word, lemma, POS, polarity and gender. In total it is 688.175 sentences. Each text from the web scraped articles in figure 17, is processed by spaCy to identify all sentences and afterwards to identify all words in the sentences. If the word isn't noise in form of a stop word or one of named entity types *['TIME', 'DATE', 'GPE', 'CARDINAL', 'PERSON', 'MONEY', 'PERCENT']*, it is then prepossessed to remove URLs, emails, special characters and HTML tags. Finally if it isn't a empty string after being prepossessed, it is appended to the corpus. Each word is appended along with its sentence number, lemma, POS-tag, gender and polarity from the polarity dictionary, which is 0 if it is not found. The POS-tag is identified by spaCy.

6.2.3 Feature engineering

This section will cover all preprocessing methods used, when the *Sentiment corpus* is first loaded, before it is used to fit a model. From the corpus, consisting of 4.970.466 samples, 3.000.000 samples from the head is extracted for training dataset and 1.000.000 samples from the tail for test dataset.

	Sentence #	Word	Lemma	POS	Polarity	Gender
0	1	organizations	organization	NOUN	-0.223025	F
1	1	decided	decide	VERB	-0.039008	F
2	1	drop	drop	VERB	-0.058436	F
3	1	women	woman	NOUN	0.272143	F
4	1	accused	accuse	VERB	0.006282	F
...	...	...	...	...	...	...
2999995	404384	cool	cool	ADJ	0.037064	F
2999996	404384	factors	factor	NOUN	-0.055300	F
2999997	404384	like	like	ADP	-0.129606	F
2999998	404384	pore	pore	NOUN	0.000000	F
2999999	404384	refiner	refiner	NOUN	0.000000	F

Figure 21: The allocated training dataset, extracted from the *Sentiment corpus*. It is essential to extract test data from the tail to prevent misleading test results, since the polarity dictionary is computed from the first 75% of the *Sentiment corpus* to prevent overfitting. The training and test dasets will roughly go through the same preprocessing steps.

6.2.3.1 Data factorization

The sentiment algorithms should be trained on sequences of words, as described in *3.3 Requirement specification*. Therefore the *Sentiment corpus* is grouped by sentence index.

	Sentence #	Word	Lemma	POS	Polarity	Gender
0	1	[organizations, decided, d...	[organization, decide, dro...	[NOUN, VERB, VERB, NOUN, V...	[-0.223024829414031, -0.03...	F
1	2	[Women, previously, worked...	[woman, previously, work, ...	[NOUN, ADV, VERB, VERB, NO...	[0.27214283999099703, -0.0...	F
2	3	[response, allegations, Vi...	[response, allegation, Vis...	[NOUN, NOUN, PROPN, VERB, ...	[-0.028053754978966002, 0....	F
3	4	[aware, allegations, Mr]	[aware, allegation, Mr]	[ADJ, NOUN, PROPN]	[-0.01571544139416, 0.0046...	F
4	5	[point, Visa, suspending, ...	[point, Visa, suspend, mar...	[NOUN, PROPN, VERB, NOUN, ...	[-0.22872995655083203, 0.0...	F
...	...	...	...	...	...	...
362024	404378	[Philosophy, Help, Cream]	[philosophy, help, cream]	[NOUN, VERB, NOUN]	[0.0, -0.25124934299454604...	F
362025	404379	[Philosophy, gem, great, b...	[Philosophy, gem, great, g...	[PROPN, NOUN, ADJ, ADJ, NO...	[0.0, 0.0, -0.192702886367...	F
362026	404381	[Benefit, POREfessional, P...	[benefit, porefessional, p...	[VERB, NOUN, NOUN, VERB, N...	[-0.16728961099418502, 0.0...	F
362027	404383	[Pores]	[Pores]	[PROPN]	[0.0]	F
362028	404384	[Pore, Refiner, notch, pro...	[Pore, Refiner, notch, pro...	[PROPN, PROPN, NOUN, NOUN,...	[0.0, 0.0, 0.0, -0.1630039...	F

Figure 22: The training dataset grouped per *Sentence #*. Each sample consists of a sequence of words, lemmas, POS tags, genders and polarities. The gender label is transformed to a single value representative of the entire sentence for binary classification, and polarities are kept as a sequence for multi classification for each word in a sentence. Both the training and test dataset are subsequently shuffled to further prevent overfitting.

6.2.3.2 Partition of data

To prevent any bias towards one of the two genders, all samples with gender-label M (man/masculine) and W (woman/feminine) are counted, from which they are partitioned into equal size. Most sequences are of different lengths, but the shape of the input to a model is required to be constant. To solve this, sequences can be padded or truncated to match a fixed length. The length could be any positive integer, but it has been chosen to find the length, in which 80% of the sequences fit, called the decision boundary.

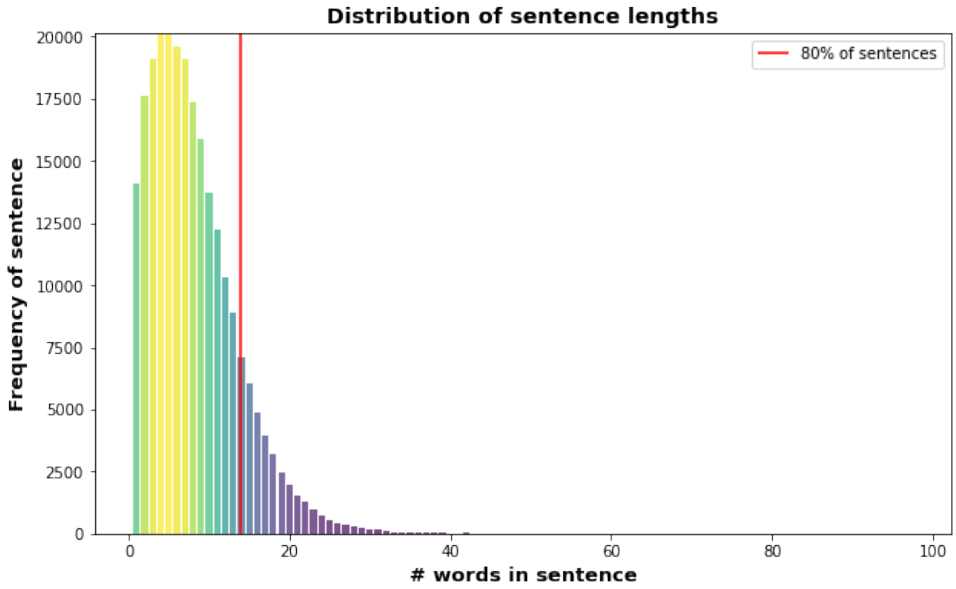


Figure 23: The distribution of sentence lengths and a vertical line marking the decision boundary at 14, in which 80% of the sequences fit.

6.2.3.3 Feature tokenization

A tokenizer is fit on all lemma sequences, which updates its internal vocabulary with each unique word, ranked by its term frequency. The closer to 0, the more frequent a word is, 0 is reserved for padding [54]. Each sequence is then transformed to a sequence of integers based on their index in the tokenizer's internal vocabulary. POS-tags are tokenized with same process. The sequences are then padded or truncated according to the decision boundary.

Base	['government', 'official', 'consider', 'game', 'vital', 'nation', 'morale']
Encoded	[371, 591, 209, 34, 2398, 596, 5200]
Padded	[371., 591., 209., 34., 2398., 596., 5200., 0., 0., 0., 0., 0., 0., 0.]

Table 18: Example of a sequence of lemmas being preprocessed. Firstly we have a sequence of lemmas, from sentence number 26809, which are encoded using a tokenizer, mapping each to their respective unique integer value. Lastly the sequence is padded or truncated at the end of sequence to match the decision boundary passed as an argument. After all lemma and POS-tag sequences in the training dataset are preprocessed in the same manner, it becomes the final input for the model, X_train_lemma, X_train_pos, X_test_lemma and X_test_pos.

For binary classification the gender is encoded to integers, -1 if the gender-label is M and 1 if F, and then one-hot encoded.

	M	**F**
Base	'M'	'F'
Encoded	0	1
One-hot	[1., 0.]	[0., 1.]

Table 19: Gender values encoded using a labelencoder, and one-hot-encoded to binary class matrices. The labelencoder is fit on a array with the gender labels, *['M', 'F']*, mapping each value to an unique integer. All gender rows are preprocessed using these steps, resulting in training target matrix y_train and test target matrix y_test for binary sentiment classification.

For multi-classification every polarity is rounded to one decimal, resulting in a maximum of 21 classes (-1 to 1 with stepsize 0.1). Every polarity is tokenized to an unique integer value and one-hot encoded to binary class matrix, as shown in table 20.

Tag	Tokenized tag	Binary class matrix
PAD	0	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
-1.0	1	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
-0.9	2	[0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
-0.8	3	[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
-0.7	4	[0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
-0.6	5	[0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
-0.5	6	[0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
-0.4	7	[0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
-0.3	8	[0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0]
-0.2	9	[0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0]
-0.1	10	[0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0]
0.0	11	[0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0]
0.1	12	[0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0]
0.2	13	[0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0]
0.3	14	[0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0]
0.4	15	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0]
0.5	16	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0]
0.6	17	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0]
0.7	18	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0]
0.8	19	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0]
0.9	20	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0]
1.0	21	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]

Table 20: Visualization of encoding and binary class matrix transformation of polarities. The sequence of polarities are padded using the PAD binary class matrix shown in the figure.

6.2.3.4 Token embedding

The tokenized sequences are embedded to a fixed size, stretching each integer to a vector of that size, thereby changing each sequence from 1D to 2D. Embedding is an alternative to one-hot encoding, that represents words in a better way with reduced dimensions. It converts each word to a vector of a defined size with real values, instead of 0's and 1's, which reduces need for large storage capacity and increases model efficiency. To clarify, if one was to one-hot encode all words in the *Sentiment corpus*, the resulting vector size of each would be equal to the number of unique words in the corpus.

6.2.4 Implementation

At this point all input is set, and all four sentiment algorithms are composed. One model is composed based on the research in *4.3.3 Comparison*, and used for all 4 sentiment algorithms, only varying in input and output dimensions. The composed model will use biLSTM, which has properties to understand complex contextual correlations. Only *Sentiment algorithm 4* will covered, but the other algorithms and more in-depth details can be found in *Annex - 6.4 Implementation of neural networks.*

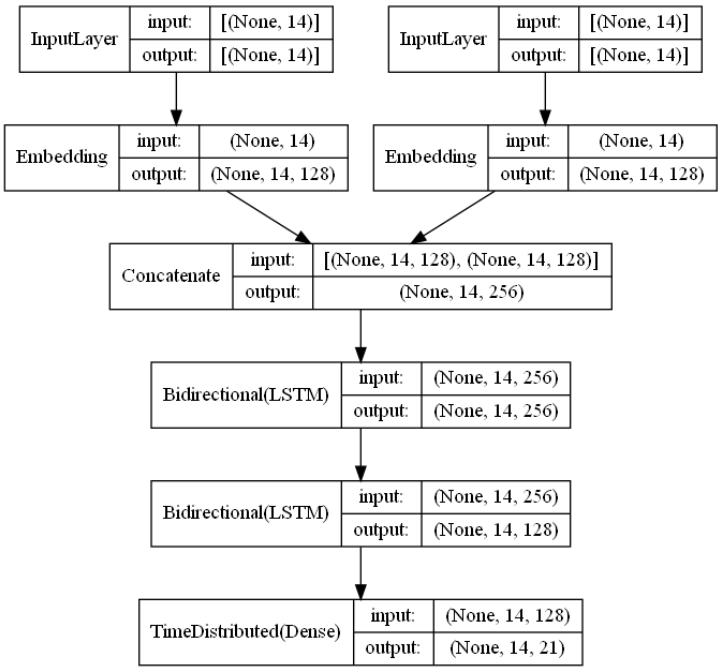


Figure 24: Layer architecture of the multi sentiment classification *Sentiment algorithm 4* BiLSTM neural network. The output dimensions, *(None, 14, 21)*, of the last layer is the 21 unique polarities (-1.0 to 1.0 with stepsize 0.1).

Architectural composition of multi-input multi sentiment classification network		
Multi-input: Lemma and POS-tag		
Concatenate layer:		
	Output dimensionality:	256
Bidirectional layer (LSTM):		
	Dropout:	0.3
	Recurrent dropout:	0.3
	Return sequences:	true
	Output dimensionality:	128
Bidirectional layer (LSTM):		
	Dropout:	0.1
	Recurrent dropout:	0.1
	Return sequences:	true
	Output dimensionality:	64
Time distributed layer (dense):		
	Output dimensionality:	21
	Activation function:	softmax

Table 21: Architectural composition of the *Sentiment algorithm 4* BiLSTM network for multi sentiment classification. Displaying actively set hyper parameters, whereas many hyper parameters are default values for the layers provided by Keras.

Furthermore a first iteration of XAI is implemented, which calculates the total polarity for a sentence. No publicly available equation to calculate a total polarity for a sentence could be found. To solve this issue an equation has been self-developed, equation (6), that calculates a total weight of a sentence's bias. As part of equation (6), equation (5) is defined, which is a first-order logic predicate symbol.

Put in natural language, equation (6) calculates the following: the polarity of a sentence p_s is given by the sum of the predicted polarities of the lemmas at the specified indexes in the sentence s, divided by the amount of lemmas in the sentence s, that has a polarity.

$$P(t) = \{t | t \neq 0.0\} \tag{5}$$

$$p_s = \frac{\sum_{s_i=0}^{|s|} p_{t_{s_i},c}}{\sum_{s_i=0}^{|s|} \exists s_i(P(s_i))} \tag{6}$$

If the overall polarity exceeds a fixed threshold, the XAI will present the lemmas with the biggest spikes in polarity.

6.2.5 Evaluation

The methodology of the sentiment algorithms and the selected features is nothing new, and is used in established classic sentiment analyses, determining positive, negative or neutral. Thereby the approach is using tested and iterated methods, known to be effective in classic sentiment analysis, and thereby adapted to gender sentiment analysis.

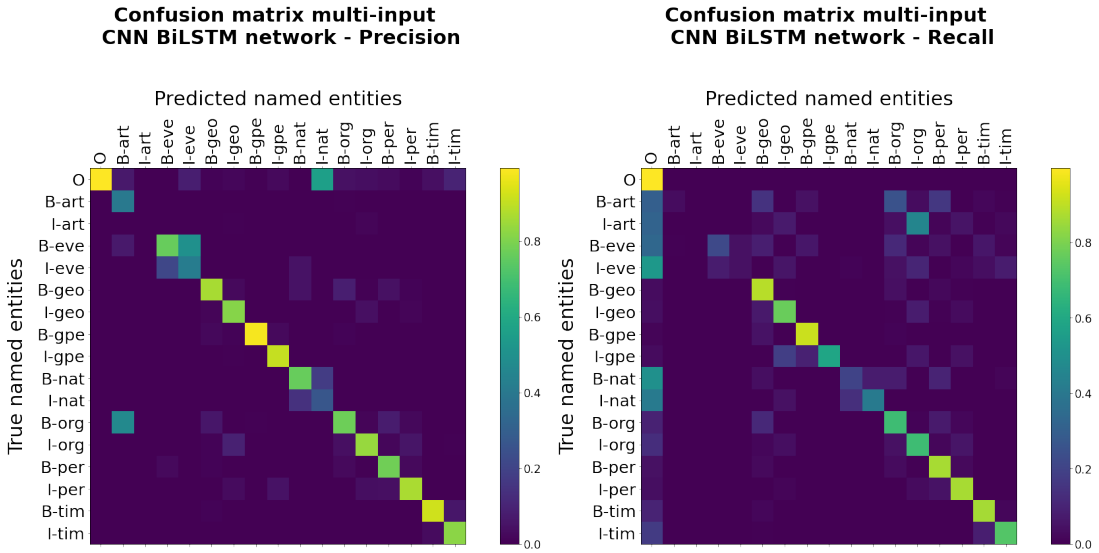
The methodology of the data collection and processing is mostly self-developed, which was a necessity, since no appropriate data could be found for gender sentiment analysis. All choices made in the process were first iterations of the Sentiment corpus' first iteration, and made based on the research and knowledge gained by reading studies of other gender sentiment analyses.

7 Results

This section will present highlighted results for the composed networks for both the LA and SA. The results have been obtained by use of the methodological approaches in *6 Methodology* as a template for preprocessing of data and implementation of various neural networks. The results are presented separately for the LA and SA, and in the form of score measurements of the composed networks, to measure their applicability regarding their individual key points from *1.2 Problem statement*. For the best performing network, confusion matrices are presented as well. The results of all neural networks both during training and testing can be found in *Annex - 7 Results*.

7.1 Linguistic analysis: Results

The overall best performing neural network was the multi-input CNN BiLSTM, which confusion matrices can be seen in figure 25, where precision and recall measurements are visualized. For confusion matrices, the contrast of the diagonal is important, since it gives a visual idea of how well predicting the network is. Confusion matrices for all neural networks for the LA are to be found in *Annex - 7.1.2.2 Confusion matrix*.



(a) Confusion matrix showing the precision values for the multi-input CNN BiLSTM network.

(b) Confusion matrix showing the recall values for the multi-input CNN BiLSTM network.

Figure 25: Confusion matrices for multi-input CNN BiLSTM network, where 25a is a visualization of precision and 25b is a visualization of recall. The values of each square can range between 0-1, and indicates how well the network predicted the actual NE, with 1 being excellent and 0 being poor.

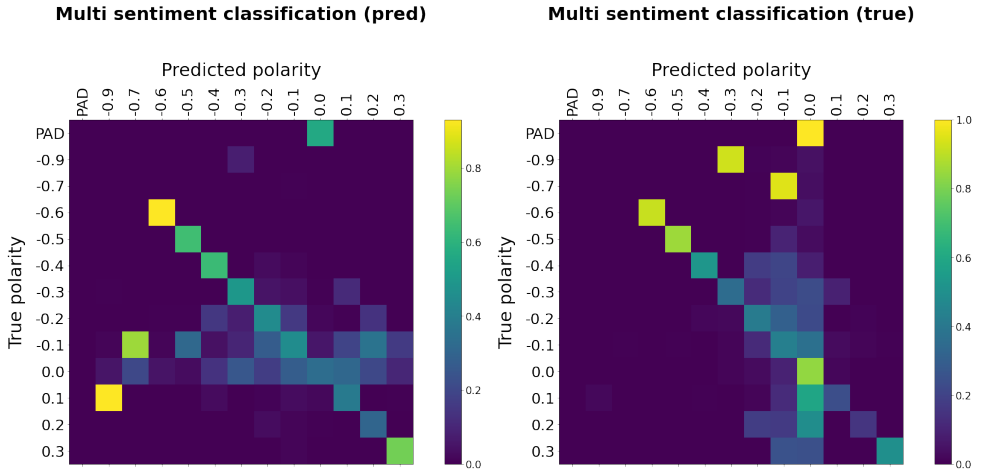
Additionally, the F1-score of all single-input and multi-input networks are shown, to provide a collective comparison across all networks. The specific results on precision and recall are to be found in *Annex - 7.1.2.1 Score metrics*, but since the F1-score is dependent on both precision and recall, it gives an indication of how well performing each neural network is. In table 22, the F1-scores for all single-input and multi-input networks are shown in a color-coded fashion.

	Score metrics for single-input and multi-input networks							
	F1-score for single-input networks				F1-score for multi-input networks			
	LSTM	BiLSTM	CNN	CNN BiLSTM	LSTM	BiLSTM	CNN	CNN BiLSTM
O	0.99	0.99	0.99	0.99	0.99	1.00	0.98	1.00
B-art	0.09	0.00	0.00	0.00	0.11	0.07	0.00	0.06
I-art	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00
B-eve	0.36	0.29	0.00	0.35	0.07	0.36	0.00	0.34
I-eve	0.12	0.17	0.00	0.12	0.25	0.30	0.00	0.08
B-geo	0.83	0.87	0.85	0.86	0.85	0.88	0.71	0.88
I-geo	0.77	0.78	0.73	0.77	0.76	0.81	0.65	0.79
B-gpe	0.94	0.95	0.94	0.95	0.95	0.96	0.92	0.95
I-gpe	0.67	0.69	0.62	0.55	0.71	0.72	0.65	0.71
B-nat	0.32	0.02	0.00	0.02	0.40	0.42	0.00	0.32
I-nat	0.00	0.00	0.00	0.00	0.28	0.17	0.00	0.32
B-org	0.66	0.73	0.68	0.71	0.69	0.75	0.56	0.73
I-org	0.75	0.77	0.71	0.76	0.75	0.78	0.61	0.76
B-per	0.80	0.82	0.79	0.80	0.82	0.83	0.68	0.82
I-per	0.86	0.85	0.83	0.85	0.86	0.87	0.53	0.87
B-tim	0.85	0.87	0.87	0.88	0.86	0.90	0.80	0.89
I-tim	0.71	0.72	0.67	0.72	0.73	0.77	0.53	0.77

Table 22: F1-score for both single-input and multi-input LSTM, BiLSTM, CNN and CNN BiLSTM networks. The colors of each cell can be one of four dependent on the value of the cell. Red is [0;0.25[, orange is [0.25;0.50[, yellow is [0.50;0.75[and green is [0.75;1.00].

7.2 Sentiment analysis: Results

The overall best performing neural network for binary sentiment classification was *Sentiment algorithm 2*, but the results of greatest interest are from multi-input multi-output *Sentiment algorithm 4*, whose confusion matrices can be seen in figure 26. Confusion matrices for all sentiment algorithms can be found in *Annex - 7.2.2.2 Confusion matrix*.



(a) Confusion matrix showing the precision values for *Sentiment algorithm 4*.

(b) Confusion matrix showing the recall values for *Sentiment algorithm 4*.

Figure 26: Confusion matrices showing prediction results for *Sentiment algorithm 4*, multiple features (sequence of lemmas and POS-tags) as input and multi sentiment classification network. It presents the prediction results on the test dataset as with a pair of confusion matrices, one visualizing the precision and the other visualizing recall. The precision and recall is measured between predicted polarity and true polarity of each word index in a sequence. Confusion matrix (a) normalized over the predicted (columns), (b) normalized over the true (rows).

In table 23 the score metrics; precision, recall and F1-score are compared for the two mutli sentiment classification networks.

	Score metrics for multi classification					
	Precision		Recall		F1-score	
	SF	MF	SF	MF	SF	MF
-0.9	0.00	0.00	0.00	0.00	0.00	0.00
-0.7	0.00	0.00	0.00	0.00	0.00	0.00
-0.6	0.89	0.93	0.91	0.91	0.90	0.92
-0.5	0.93	0.64	0.94	0.86	0.94	0.74
-0.4	0.60	0.63	0.66	0.53	0.63	0.58
-0.3	0.35	0.49	0.29	0.35	0.32	0.41
-0.2	0.46	0.45	0.47	0.42	0.47	0.43
-0.1	0.42	0.45	0.45	0.43	0.43	0.44
0.0	0.33	0.33	0.80	0.84	0.47	0.48
0.1	0.33	0.38	0.21	0.24	0.26	0.30
0.2	0.29	0.30	0.18	0.15	0.22	0.20
0.3	0.84	0.73	0.40	0.50	0.54	0.59

Table 23: Precision, recall and F1-score for multi sentiment classification single and multiple feature input, labeled SF and MF respectively. SF: *Sentiment algorithm 2*. MF: *Sentiment algorithm 4*. The colors of each cell can be one of four dependent on the value of the cell. Red is [0;0.25[, orange is [0.25;0.50[, yellow is [0.50;0.75[and green is [0.75;1.00].

Additionally the binary sentiment classificaiton networks' score metrics can be seen in table 24.

	Score metrics for binary classification					
	Precision		Recall		F1-score	
	SF	MF	SF	MF	SF	MF
M	0.54	0.57	0.54	0.52	0.54	0.54
F	0.54	0.56	0.54	0.60	0.54	0.58

Table 24: Precision, recall and F1-score for binary sentiment classification single and multiple feature input, labeled SF and MF respectively. SF: *Sentiment algorithm 1*. MF: *Sentiment algorithm 3*. The colors of each cell can be one of four dependent on the value of the cell. Red is [0;0.25[, orange is [0.25;0.50[, yellow is [0.50;0.75[and green is [0.75;1.00].

Two XAI explanations from *Sentiment algorithm 4* has been cherry-picked, one to present a sentence predicted to be masculine and one feminine. All four algorithms have XAI explanations, which can be found in *10.5 Appendix E*.

```
4   ----------------------------------------------------------------
5
6   The sentence: ""A further order will be entered contemporaneous with
7   the formal decision," the temporary order said.".
8
9   Was predicted to be masculine (-0.3).
10
11  Most masculine words in sentence are:
12
13  'order' with a polarity of -0.6
14
15  'formal' with a polarity of -0.3
16
17  ----------------------------------------------------------------
18
19  The sentence: "In 2010, 27 percent of millionaires worldwide were women.".
20
21  Was predicted to be feminine (0.3).
22
23  Most feminine words in sentence are:
24
25  'women' with a polarity of 0.3
26
27  'worldwide' with a polarity of 0.0
```

Figure 27: A screenshot of two XAI explanations made by *Sentiment algorithm 4*. The multi sentiment classification algorithms' XAI works independently of the polarity dictionary, only using the predicted polarities. If the overall sentiment, computed by equation 6, of the sentence exceeds the threshold (-0.1 and 0.1), XAI presents the most masculine or most feminine words, dependent of which way the overall polarity leans.

8 Discussion

This section will start off with discussing the results in *7 Results* and the underlying process of attaining the results of the LA and SA. The results of the LA and SA will revolve around the measured score metrics and confusion matrices. A comparison between networks will be conducted and tendencies will be stated. Afterwards, the discussion will move on towards the social science aspect of the project, also with an ethical perspective in mind.

8.1 Linguistic analysis: Discussion

The sole purpose of the LA was to analyze potential machine learning solutions, specifically with the use of neural networks, to *Key point 1.* in the problem statement:

"[To solve] discrimination of A in regard to demographic characteristics and socioeconomic status (...)"

It was assessed that the linguistic feature NER could be used to recognize demographic characteristics of A with the intended purpose of censoring these characteristics in the application. As stated in *3.3 Requirement specification* the dataset of the LA didn't consist of job applications, which therefore already decouples it from the intended domain, job recruitment. So how can it be possible to draw conclusions on a perhaps inadequate dataset, that has no relation to the context the project is intended to?

8.1.1 Inadequate dataset?

Since the LA is only proof of concept as stated in *2 Project description*, it is arguably sufficient to prove the concept based on a dataset with no relation to the intended domain context. The only two major differences are, that the data is labelled with NE tags mostly unrelated to job applications, and that the language usage probably is not alike, meaning that the acquired behaviour of each neural network is targetting articles rather than job applications. For that reason, future work is also proposed, so that a dataset consisting of diverse job applications can be gathered.

8.1.2 Empirical observations from testing results

Valid for both precision and recall measurements of both single-input and multi-input networks is, that predictions for NE tags such as *Artifacts, Events* and *Natural Phenomena* are poor. On the other hand, predictions for NE tags such as *Geographical entities, Geopolitical entities, Persons, Organizations* and *Time* are between satisfying and excellent, which is optimal since these are the demographic characteristics the project aims to find and censor. The representation of the NE's that was poorly predicted ranges between 51 and 402, whereas the representation of NE's that was predicted satisfactorily and better ranges between 6,528 and 37,644. This strongly suggests that there is a correlation between the total amount of a NE tag and the precision and recall of the prediction of that NE tag. Table 22 in *7.1 Linguistic analysis: Results*, displaying F1-scores for all composed neural networks, also suggests that the NE tags clearly can be segmented based on their representative amount. Since the F1-score is measured on the basis of precision and recall, this makes it a good indication for the correlation for both mentioned score metrics.

Observations for confusion matrices representing precision

Considering the confusion matrices representing precision in *Annex - 7.1.2.2 Confusion matrix*, a few observations can be made. In general terms, the CNN's are good at not falsely predicting NE's, whereas The LSTM's and BiLSTM's (from now on contracted as RNN's) are better at predicting the NE's that are vaguely represented in the dataset, but as a compromise introduces more falsely predicted NE's. This is both valid for single-input and multi-input networks. Therefore, two observations are deduced; the CNN's work great as filters for false positives and the RNN's cover a broader range of NE tags, but introduces more false positives. So with the two observations in mind, looking at the CNN BiLSTM's, especially for the single-input CNN BiLSTM, many false positives are discarded and the precision is higher for various NE tags than for regular CNN's. So a combination of CNN's and RNN's seems profitable as for precision.

Observations for confusion matrices representing recall

Considering the confusion matrices representing recall in *Annex - 7.1.2.2 Confusion matrix*, the tendency among the networks are harder to spot. However, a general observation is, that for all vaguely represented NE-tags, many false negatives are introduced. Since the confusion matrices representing recall are not easy to interpret, looking in the tables for single-input and multi-output metrics in *Annex - 7.1.2.1 Score metrics* for assistance seems like a better option. Here it becomes apparent that the CNN BiLSTM and the BiLSTM have the best recall score measurements, for single-input and multi-input respectively compared to the rest. Therefore, one observation from the confusion matrices representing recall is deduced; the recall measurements for the vaguely represented NE tags are all over the place, making it hard to evaluate the result. What subsequently can be stated from this observation is, that if the dataset consisted of a more homogeneous representation of NE tags, with at least a five-digit representative of each NE-tag, the recall measurements could have scored better.

8.2 SA: Discussion

The purpose of the SA was to analyze potential machine learning solutions, specifically with the use of neural networks, to *Key point 2.* in the problem statement:

"[To solve] discrimination of A in regard to gendered wording, both in terms of wording in job applications, potentially leading to implicit cognitive perceptions of A, as well as in job advertisements, potentially discouraging A from applying for the job."

It was quickly concluded that no public datasets exists to even start tackling this problem, which requires data suitable to learn tendencies and hidden patterns in how males and females communicate. Hence the SA had to put more emphasis on creating a qualified dataset, than examining multiple suitable neural networks.

The resulting dataset created, the *Sentiment corpus*, contains all features thought to be required and contains 4.9 million samples, in which each sample contains 'Sentence #', 'Word', 'Lemma', 'POS', 'Polarity' and 'Gender'. It can be manipulated into sequences, grouped by sentence index, to be used for binary or multi sentiment classification, and the sentiment for each word and sentence are computed based on a gender.

To apply the corpus four algorithms were developed, which all include the preprocessing, training, prediction and XAI of test data. All algorithms uses the RNN neural network biLSTM model architecture, which is strong on contextual awareness and commonly used on difficult NLP problems. The four algorithms are grouped by classification problem (binary and multi), which were described in *6.2.1 Features and classification.*
The self-developed *Sentiment corpus* seems like a suitable dataset, used with the suitable neural network biLSTM, but do the results, in *7.2 Sentiment analysis: Results*, implicate otherwise?

8.2.1 Empirical observations from training and testing results

Looking at the test results for binary sentiment classification in *7.2 Sentiment analysis: Results*, specifically score metrics in figure 24 in *7.2 Sentiment analysis: Results*, it is apparent that the models have equal performance when used on test data, although the multi features input algorithm has slightly better precision.

Looking at the training results for multi sentiment classification in *Annex - 7.2.1 Training results*, it is evident that the models almost instantly become overfitted, almost reaching an accuracy score of 0.99 by the end of the first epoch. The overfitting makes sense though, since the polarity dictionary was computed using the training data. Presumably the model just tries to guess all keys and values within the dictionary, and once it has, the values are static and never change according to a context, which means it quickly reaches close to 100% accuracy.

Looking at the test results for multi sentiment classification, specifically the score metrics in table 23 displaying precision, recall and F1-score, it can be seen that both models perform extremely poorly in all sections. The majority of the precision scores are well below 0.50, which means under 50% of polarities the models predict are correct. It does have high precision predicting the strongest polarities, the ones near -1.0 or 1.0, but that is expected to be true, when the support for such are low. The main part of the polarities are neutral or close to neutral, meaning most words in the corpus do not hold great gender bias.

It is evident that not a single on of the four models meet the precision score requirements set in *3.3 Requirement specification*. What could be causing the lacking results? Does *Sentiment corpus* have any fatal flaws?

8.2.2 A flawed sentiment corpus

The *Sentiment corpus* was built using presumed genders of presumed target audiences of multiple news articles' categories. The model is asked to find correlations in texts provided for each gender, but if the genders aren't true to any of genders' language usage, it is evident that no correlations will be found. That raises the question, is grouping texts' gender by news categories' presumed audience the best approach? A lot of bias is possibly introduced already at that point, since the genders are selected by this project's authors. The reason that approach was picked initially was, because it was thought that for example a category like *WOMEN* would be heavily nuanced by a feminine language usage. That may be true, but it seems to be true for categories like *MONEY* also, which is evident in *Annex - 7.2.2.3 XAI*. The XAI logs for *Sentiment algorithm 1* shows that the sentence was predicted masculine due to the words 'fund' and 'returns', which seems heavily influenced by the money/finance category's language usage.

A better solution could be to develop an algorithm to predict the gender of an author by their name, using a public name list of each gender [62]. Using this approach it would be possible use all news categories from *News Category Dataset* by *Rishabh Misra* [34], increasing the article count from an intial 25.667 to 200.853. Quantity of samples of each category could then be equalled to reduce significance of lingo related language usage. This could possibly also solve the dataset's limited span of polarities, the present polarities in the dataset are -0.7 to 0.3, missing a lot of feminine polarities.

The polarities for each word are always static, no matter the context of a sentence, since it just gets its polarity from the polarity dictionary. A more dynamic dataset with polarities representative of contextual setting of each instance could be used to prevent the tendency discussed in *8.2.1 Empirical observations from training and testing results*, in which the multi sentiment classification models quickly become overfitted, as they simply have to guess the polarity.

8.3 Ethical dilemmas

The project introduces and chooses to ignore many humanistic and ethical dilemmas, since it just is not the project's focus as it is proof-of-concept, but that does not make them any less important.

A lot of bias is introduced in early stages of creating the *Sentiment corpus*, when it is chosen that the news article categories; *SPORTS*, *MONEY* and *BUSINESS* are chosen to be representative of masculine phrasing and wording, and *WOMEN* and *STYLE & BEAUTY* are chosen to represent feminine phrasing and wording. The categories are chosen by the authors of this report, and it is most likely that a different group with different demographic characteristics or socioeconomic status would perceive it differently. Initially it is probably ethical wrong to choose texts to be true of a gender's language usage by what news categories they fall under. It can be difficult to be ethical clean, even if the approach proposed in *8.2.2 A flawed sentiment corpus*, using the author's name to determine the gender, was used, it is not certain that it would correctly reflect the gender that the author identifies as. Genders aren't binary anymore and should be treated accordingly, to reduce the bias and prevent discrimination, the diversity of the group could be increased, allowing for more diverse opinions.

The survey in *4.4.2 Study of public demand* seeks to gain insight on the public's personal opinions and experience with problems described in *1.1 Description of problem*, and whether there is any demand for a solution, like the one the project aims to be a proof-of-concept of. It seems that neither A or E wants to discriminate towards anybody and the opinion of whether to include an algorithm into the job recruitment process to prevent discrimination is split evenly. E almost in unison agrees that demographic characteristics may influence their opinion of a candidate subconsciously and it shouldn't unless for practical reasons, but contradictory believe that they can make objective opinions. The questionnaire's credibility should also be questioned, due to the low numbers and diversity of the participants.

9 Conclusion

Having discussed the results of the LA and SA, conclusions can be drawn as to whether or not the findings honor the key points stated in *1.2 Problem statement*. Considering the results for the LA in terms of it being a valid solution to censoring demographic characteristics and socioeconomic status, it proved that for strongly represented NE's in the dataset (>6000), it mostly succeeded in meeting the required recall measurement of 75%. However for vaguely represented NE tags in the dataset (<1000), the LA failed to meet the requirements. So from the score measurements of the LA it can be concluded, that with a uniformly and strongly represented dataset in terms of named entities, neural networks composed of RNN and CNN performing NER, can be used to recognize named entities in natural language. The question is if the methodological approach can be translated to the job recruitment context, since the LA didn't operate on domain-related data. The analysis indicates that with a domain-specific dataset similar to the one trained upon for the LA, it should be possible to use NER to recognize demographic features and socioeconomic status in job applications.

Alternatively, one could consider if an inverse approach could be used, where instead of recognizing demographic characteristics to be censored, neural networks perfoming NER could be used to recognize entities regarding skills and experience. This would raise the margin of error allowed, since a false negative of experiences and skills would be less fatal than a false negative demographic characteristic, since slip-ups of missing classifications of demographic characteristics would defeat the purpose of anonymizing A.

Considering the results of the SA in terms of it being a valid solution to classifying gendered wording based on a computed dataset of polarities, it proved that binary classification of entire sentences was ineffective, since the network practically just guessed the polarity of the sentence, resulting in a 50% success rate at best. Classification of gender polarity of each word also proved ineffective, with average precision measurements at approximately 35%. The ultimate bottleneck for the measurements, results from the computed polarity dictionary, that was based on articles from selected topics from a specific news media. This introduced a biased lingo originating from the articles of the selected topics. Additionally, the extended *tf-idf weighting scheme* approach to calculate polarities of all words across all articles lacked diversity in polarities, meaning the span of polarity of all words were too narrow.

10 Appendix

10.1 Appendix A

Appendix A is the the source code, and it is enclosed in the attached *Appendixes* zip-folder.

10.2 Appendix B

Text from first sample of *3_text_and_gender.json*, which is the web scraped body text from an article with category *WOMEN*.

At least two organizations have decided to drop Morgan Freeman after eight women accused him of inappropriate behavior and sexual harassment. Women who previously worked with the Oscar-winning actor told CNN that he repeatedly made comments about their bodies or their clothing and frequently engaged in inappropriate touching. In response to the allegations, Visa announced it had suspended him from its marketing campaign. We are aware of the allegations that have been made against Mr. Freeman. At this point, Visa will be suspending our marketing in which the actor is featured, Visa said in a statement. TransLink, Vancouver's public transit system, also decided to stop using Freeman's voice as part of an ad campaign to promote its Visa credit card and mobile payments on bus and Skytrain operations. In light of information we've learned ... of allegations regarding actor Morgan Freeman, TransLink has decided to pause his voice announcements as part of a Visa ad campaign on our transit system, the transit system said a statement. We will be reaching out to Visa to discuss further. A few hours after the report was released, Freeman, 80, issued an apology: Anyone who knows me or has worked with me knows I am not someone who would intentionally offend or knowingly make anyone feel uneasy, I apologize to anyone who felt uncomfortable or disrespected — that was never my intent. A production assistant working on the 2015 film Going in Style said Morgan made numerous comments about her body, asked her several times if she was wearing underwear and tried to lift her skirt. Another worker on the set said women wore loose-fitting clothing in an effort to avoid Freeman's attention. Entertainment Tonight released footage on Thursday evening of two on-air interviews with Freeman that also revealed his attitude toward women. During an interview for the 2016 film London Has Fallen, Freeman asked a young female reporter, My goodness, are you married? Fool around with other guys? I'm just asking. In 2015, he told author Janet Mock: You got a dress halfway between your knee and your hips, and you sit down right across from me and you cross your legs. Mock, who was a special correspondent for the interview, said Freeman's treatment of her was an exhibition of the casual nature at which men in positions of power believe that everything belongs to them, including women's bodies. Freeman's union, SAG-AFTRA, is considering what to do about Freeman's Life Achievement Award. The award is given to actors who represent the finest ideals of the acting profession, CNN reported. These are compelling and devastating allegations which are absolutely contrary to all the steps that we are taking to ensure a safe work environment, the union said in a statement. Any accused person has the right to due process, but it is our starting point to believe the courageous voices who come forward to report incidents of harassment ... we are reviewing what corrective actions may be warranted at this time.

10.3 Appendix C

Study of public demand, questions and answers grouped by the survey group.

Applicants

Opsætning:

Du er job ansøger, og skal til at sende dit CV og din motiverede ansøgning.

	Spørgsmål	Ja	Nej	Ved ikke
1	Ville du lade din ansøgning køre igennem en algoritme, som censurerer dine personlige oplysninger (køn, alder, etnicitet, race og religion), for at du ikke bliver valgt/fravalgt på baggrund af disse, men derimod dine kompetencer og erfaring?			
2	Tror du at man kunne mindske diskrimination i ansøgningsprocessen ved at censurere disse personlige oplysninger?			
3	Har du tidligere været nervøs for at blive diskrimineret i job-ansøgningsprocessen på baggrund af din køn, alder, etnicitet, race og/eller religion?			
4	Ville du lade din ansøgning køre igennem en algoritme, som gør dig opmærksom på kønsladede ord, som kan røbe dit køn? (eksempel: statistisk set er følgende sætning feminint ladet, grundet understregede ord: "Jeg er en sød og hjælpende person...")			
5	Ville du være villig til at gøre din ansøgning mere kønsneutral, hvis du bliver gjort opmærksom på at den indeholder kønsladede ord?			
6	Tror du at man kunne mindske diskrimination i ansøgningsprocessen ved at gøre ens ansøgning mere kønsneutral?			

Opsætning:

Du er job ansøger, og leder efter relevante jobs, du vil ansøge.

	Spørgsmål	Ja	Nej	Ved ikke
7	Ville du være *mindre* tilbøjelig til at sende en ansøgning til et job, hvor stillingsopslaget benyttede kønsladede ord, der *ikke* svarede til dit køn?			
8	Ville du være *mere* tilbøjelig til at sende en ansøgning til et job, hvor stillingsopslaget benyttede et neutralt sprog uden kønsladede ord?			

Figure 28: Questions asked in the applicant survey. The survey was printed and handed out to 10 students around the campus to be answered "anonymously". 5 males and 5 females answered, the group was primarily white.

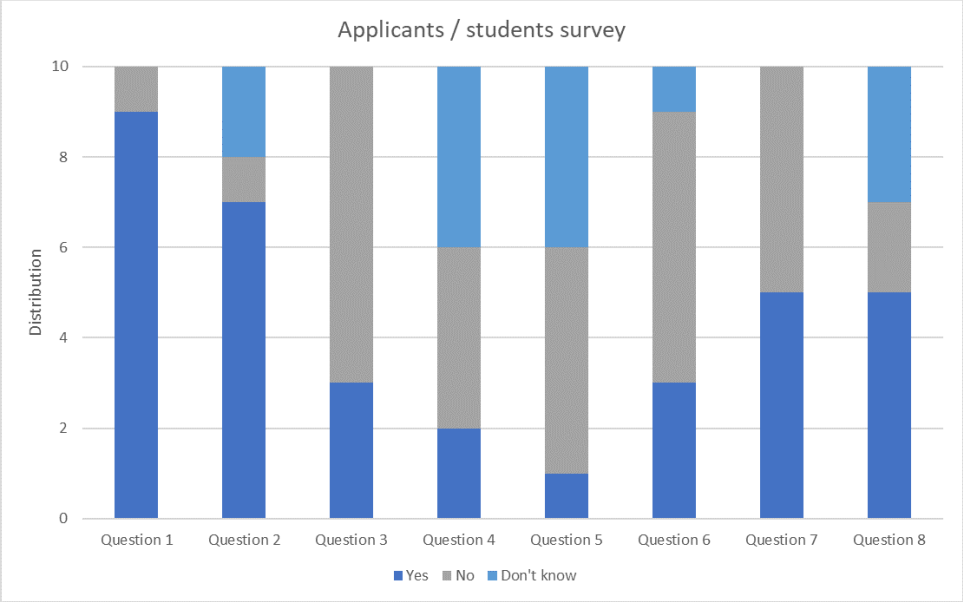


Figure 29: Distribution chart showing the answers of the applicant survey, shown in figure 28.

Employers

Opsætning:

Du er ansætter i en job ansøgningsproces, hvor du skal lave et job opslag og screene ansøgere til en første samtale.

	Spørgsmål	Ja	Nej	Ved ikke
1	Er ansøgerens køn en del af dine overvejelser ift. screeningen?			
2	Er ansøgerens etnicitet en del af dine overvejelser ift. screeningen?			
3	Er ansøgerens race en del af dine overvejelser ift. screeningen?			
4	Er ansøgerens alder en del af dine overvejelser ift. screeningen?			
5	Er ansøgerens religion en del af dine overvejelser ift. screeningen?			
6	Kunne du forestille dig, at du underbevidst lader ansøgerens køn, etnicitet, race eller religion påvirke din beslutning?			
7	Kunne du forestille dig, at du er mere tilbøjelig til at vælge kandidater af samme køn, etnicitet, race, alder og religion som dig selv?			
8	I en situation med mange ansøgere, ville du anvende algoritmer til at filtrere kandidater fra?			
9	I en situation med mange ansøgere, ville du anvende algoritmer til at fremhæve de bedst egnede kandidater? (baseret på algoritmens filtreringsparametre)			
10	Ville du anvende en algoritme til at skjule personlige oplysninger som ansøgerens køn, etnicitet, race, alder og religion?			
11	Ville du anvende en algoritme, som fremhæver ansøgerens kompetencer og erfaringer?			
12	Synes du, at det er en god idé at censurere ansøgerens personlige information i en ansøgning for at lægge fokus på kompetencer og erfaring og dermed mindske diskrimination af ansøgeren?			
13	Ville du være tilbøjelig til at anvende en algoritme, der markerede kønsladede ord i et job opslag? (eksempel: Følgende sætning indeholder ord der statistisk set er maskulint ladede: "Vi søger en stærk leder til...")			
14	Ville du være villig til at gøre dit job opslag mere kønsneutralt, hvis du bliver gjort opmærksom på at det indeholder kønsladede ord?			

Figure 30: Questions asked in the employers survey. The survey was printed and handed out to 10 teachers at their offices in the AU building Edison to be answered "anonymously". The group was primarily white males.

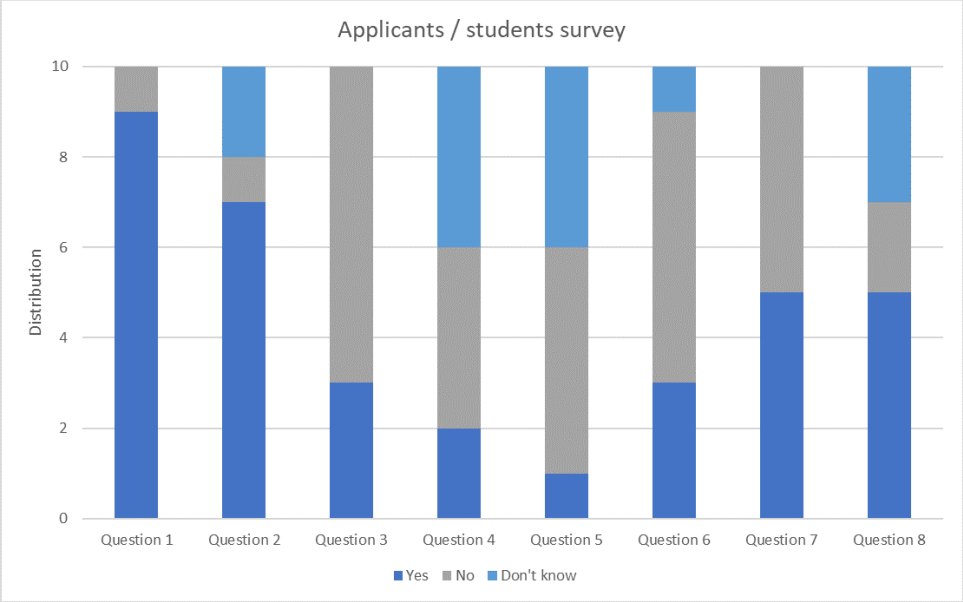


Figure 31: Distribution chart showing the answers of the employer survey, shown in figure 30.

10.4 Appendix D

Appendix D is the process report and is enclosed in the attached *Appendixes* zip-folder.

10.5 Appendix E

Appendix E is the XAI log txt files for each sentiment algorithm, and are enclosed in a folder in the attached *Appendixes* zip-folder.

List of Figures

List of Tables

References

[1] Signature Recruitment. *Gendered Words Dataset*. https://www.signaturerecruitment.co.uk/why-gender-neutral-language-improves-recruitment-ads/. [Online; Accessed 24/11-21]. 2021.

[2] Malte Dahl and Niels Krog. *Experimental evidence of discrimination in the labour market: Intersections between ethnicity, gender, and socio-economic status*. https://menneskeret.dk/sites/menneskeret.dk/files/media/dokumenter/malte_dahl_forskning.pdf. [Online; Accessed 30/08-21; Pages 83 - 114]. 2018.

[3] Marianne Bertrand and Sendhil Mullainathan. *ARE EMILY AND GREG MORE EMPLOYABLE THAN LAKISHA AND JAMAL? A FIELD EXPERIMENT ON LABOR MARKET DISCRIMINATION*. https://www.nber.org/system/files/working_papers/w9873/w9873.pdf. [Online; Accessed 30/08-21]. 2003.

[4] Malte Dahl. *Alike but different: How cultural distinctiveness shapes immigrant-origin minorities' access to the labour market*. https://menneskeret.dk/sites/menneskeret.dk/files/media/dokumenter/malte_dahl_forskning.pdf. [Online; Accessed 30/08-21; Pages 125-155]. 2019.

[5] Jeffrey Dastin. *Amazon scraps secret AI recruiting tool that showed bias against women*. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G. [Online; Accessed 03/11-21].

[6] Danielle Gaucher, Justin Friesen, and Aaron C. Kay. *Evidence That Gendered Wording in Job Advertisements Exists and Sustains Gender Inequality*. https://www.hw.ac.uk/documents/gendered-wording-in-job-ads.pdf. [Online; Accessed 30/08-21; Pages 4-5]. July 2011.

[7] Society for Human Resource Management. *Using Social Media For Talent Acquisition - Recruitment and Screening*. https://www.shrm.org/hr-today/trends-and-forecasting/research-and-surveys/pages/social-media-recruiting-screening-2015.aspx. [Online; Accessed 30/08-21; Slide 3]. Jan. 2016.

[8] The London School of Economics and Political Science. *'Big data' from online recruitment platforms show discrimination against ethnic minorities and women – and sometimes men*. https://www.lse.ac.uk/News/Latest-news-from-LSE/2021/a-Jan-21/Big-data-from-online-recruitment-platforms-show-discrimination. [Online; Accessed 30/08-21;] Jan. 2021.

[9] Kumba Sennaar. *Machine Learning for Recruiting and Hiring – 6 Current Applications*. https://emerj.com/ai-sector-overviews/machine-learning-for-recruiting-and-hiring/. [Online; Accessed 03/11-21].

[10] James Vincent. *Automated hiring software is mistakenly rejecting millions of viable job candidates*. https://www.theverge.com/2021/9/6/22659225/automated-hiring-software-rejecting-viable-candidates-harvard-business-school?fbclid=IwAR2Kie4rlDVBQumbJ9iM-8PKLbUyTrOF5e2EspLquq8pUGHysDScoqtRCBk. [Online; Accessed 03/11-21].

[11] Tim Halloran. *Watch your (gender) tone*. https://textio.com/blog/watch-your-gender-tone/13035166463. [Online; Accessed 03/11-21].

[12] Ayn-Monique Klahre. *3 Ways Johnson & Johnson Is Taking Talent Acquisition to the Next Level*. https://www.jnj.com/innovation/3-ways-johnson-and-johnson-is-taking-talent-acquisition-to-the-next-level. [Online; Accessed 03/11-21].

[13] *Annex - 3.3.1 MoSCoW analysis*.

[14] *Annex - 4.2 Datasets*.

[15] *Annex - 4 Analysis*.

[16] *Annex - 4.3 Open-source NLP libraries.*

[17] *Annex - 4.4 Machine learning frameworks.*

[18] *Annex - 4.1.1 Named entity recognition.*

[19] Wikipedia. *Part-of-speech tagging*. https://en.wikipedia.org/wiki/Part-of-speech_tagging. [Online; Accessed 29/11-21].

[20] *Annex - 4.1.2 Part-of-speech tagging.*

[21] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf. [Online; Accessed 17/11-21; Chapter 2.2.4 Stemming and lemmatization]. 2009.

[22] Hunter Heidenreich. *Stemming? Lemmatization? What?* https://towardsdatascience.com/stemming-lemmatization-what-ba782b7c0bd8. [Online; Accessed 29/11-21]. 2018.

[23] Jabir Jamal. *NLP-03 Lemmatization and Stemming using spaCy*. https://medium.com/mlearning-ai/nlp-03-lemmatization-and-stemming-using-spacy-b2829becceca. [Online; Accessed 29/11-21].

[24] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf. [Online; Accessed 17/11-21; Chapter 2.2.2 Dropping common terms: stop words]. 2009.

[25] *Annex - 4.2.1.1 Requirements.*

[26] *Annex - 4.2.1.3 Availability.*

[27] Groningen Meaning Bank. *Dataset for linguistic analysis*. https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus. [Online; Accessed 08/11-21].

[28] *Annex - 4.2.2.1 Requirements.*

[29] *Annex - 4.2.2.3 Availability.*

[30] *Annex - 4.2.2.1 Considerations.*

[31] Usman Malik. *Python for NLP: Movie Sentiment Analysis using Deep Learning in Keras*. https://stackabuse.com/python-for-nlp-movie-sentiment-analysis-using-deep-learning-in-keras/. [Online; Accessed 17/11-21].

[32] Sergio Virahonda. *An easy tutorial about Sentiment Analysis with Deep Learning and Keras*. https://towardsdatascience.com/an-easy-tutorial-about-sentiment-analysis-with-deep-learning-and-keras-2bf52b9cba91. [Online; Accessed 17/11-21].

[33] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, 2nd edition*. https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/. [Chapter 16: Sentiment Analysis]. 2019.

[34] Rishabh Misra. *News Category Dataset - Identify the type of news based on headlines and short descriptions - Version 2*. https://www.kaggle.com/rmisra/news-category-dataset. [Online; Accessed 17/11-21]. 2018.

[35] Shipra Saxena. *Introduction to Long Short Term Memory (LSTM)*. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/. [Online; Accessed 22/11-21]. Mar. 2021.

[36] Christopher Olah. *Understanding LSTM Networks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Online; Accessed 22/11-21]. Aug. 2015.

[37] Xuan Hien Le et al. *Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting*. https://www.researchgate.net/figure/The-structure-of-the-Long-Short-Term-Memory-LSTM-neural-network-Reproduced-from-Yan_fig8_334268507. [Online; Accessed 22/11-21]. 2019.

[38] Derrick Mwiti. *NLP Essential Guide: Convolutional Neural Network for Sentence Classification*. https://cnvrg.io/cnn-sentence-classification/. [Online; Accessed 22/11-21].

[39] Marc Moreno Lopez and Jugal Kalita. *Deep Learning applied to NLP*. https://arxiv.org/pdf/1703.03091.pdf. [Online; Accessed 22/11-21].

[40] *Annex - 4.5.2 Convolutional nerual networks.*

[41] Anja Bechmann and Geoffrey C Bowker. *Unsupervised by any other name: Hidden layers of knowledge production in artificial intelligence on social media*. https://journals.sagepub.com/doi/pdf/10.1177/2053951718819569. [Online; Accessed 13/12-21]. 2019.

[42] The Colour Works. https://www.thecolourworks.com/what-colour-preference-are-you/. [Online; Accessed 14/12-21].

[43] *Annex - 5 Linguistic analysis.*

[44] *Annex - 5.2.2 Word tokenization.*

[45] Harsha Bommana. *Deep NLP: Word Vectors with Word2Vec*. https://medium.com/deep-learning-demystified/deep-nlp-word-vectors-with-word2vec-d62cb29b40b3. [Online; Accessed 10/12-21].

[46] *Annex - 5.3.1 Prerequisites.*

[47] *Annex - 5.3 Implementation of neural networks.*

[48] *Annex - 6 Sentiment analysis.*

[49] Adem Akdogan. *Word Embedding Techniques: Word2Vec and TF-IDF Explained*. https://towardsdatascience.com/word-embedding-techniques-word2vec-and-tf-idf-explained-c5d02e34d08. [Online; Accessed 30/11-21]. 2021.

[50] Akiko Aizawa. *An information-theoretic perspective of tf–idf measures*. https://www.sciencedirect.com/science/article/pii/S0306457302000213. [Online; Accessed 30/11-21; Chapter 2. A brief look at conventional statistical measures]. 2003.

[51] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf. [Online; Accessed 30/11-21; Chapter 6.2 Term frequency and weighting]. 2009.

[52] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf. [Online; Accessed 30/11-21; Chapter 6.2.1 Inverse document frequency]. 2009.

[53] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf. [Online; Accessed 30/11-21; Chapter 6.2.2 Tf-idf weighting]. 2009.

[54] Keras. *Keras Tokenizer source code*. https://github.com/keras-team/keras-preprocessing/blob/master/keras_preprocessing/text.py. [Online; Accessed 03/12-21]. Last update in 2020.

[55] *Annex - 6.4 Implementation of neural networks.*

[56] *Annex - 7 Results.*

[57] *Annex - 7.1.2.2 Confusion matrix.*

[58] *Annex - 7.1.2.1 Score metrics.*

[59] *Annex - 7.2.2.2 Confusion matrix.*

[60] *Annex - 7.2.1 Training results.*

[61] *Annex - 7.2.2.3 XAI.*

[62] Social Security of the American government. *Top Names Over the Last 100 Years*. `https://www.ssa.gov/oact/babynames/decades/century.html`. [Online; Accessed 13/12-21].