



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

---

# Evaluating Lexicon-Based Models versus BERT for Sentence-Level Sentiment Analysis in Swedish

Master's thesis in Computer science and engineering

Ricardo Mansour

Erik Nilsson

---

Department of Computer Science and Engineering  
CHALMERS UNIVERSITY OF TECHNOLOGY  
UNIVERSITY OF GOTHENBURG  
Gothenburg, Sweden 2024



MASTER'S THESIS 2024

# Evaluating Lexicon-Based Models versus BERT for Sentence-Level Sentiment Ana-lysis in Swedish

Ricardo Mansour  
Erik Nilsson



UNIVERSITY OF  
GOTHENBURG

---



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering  
CHALMERS UNIVERSITY OF TECHNOLOGY  
UNIVERSITY OF GOTHENBURG  
Gothenburg, Sweden 2024

Evaluating Lexicon-Based Models versus BERT for Sentence-Level Sentiment Analysis in Swedish

Ricardo Mansour

Erik Nilsson

© Ricardo Mansour, Erik Nilsson 2024.

Supervisor: Dana Dannélls, University of Gothenburg

Niklas Zechner, University of Gothenburg

Justyna Sikora, KB-labs

Examiner: Richard Johansson, Department of Computer Science and Engineering  
& University of Gothenburg

Master's Thesis 2024

Department of Computer Science and Engineering

Chalmers University of Technology and University of Gothenburg

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Typeset in L<sup>A</sup>T<sub>E</sub>X

Gothenburg, Sweden 2024

# Abstract

This thesis explores the development and evaluation of different approaches to sentiment analysis for the Swedish language, focusing on sentence-level sentiment detection. The study compares traditional rule- and lexicon-based models with modern machine learning approaches, particularly the Bidirectional Encoder Representations from Transformers (BERT), as well as a hybrid model combining the rule-based model with Support Vector Machines SVM. Utilizing the Sparv pipeline for linguistic analysis and breakdown in tandem with the sentiment lexicon SenSALDO, we aim to enhance the existing research on Swedish rule-based models by inclusion of linguistic features. The research also involves expanding the lexicon with neutral, positive and negative entries in order to improve coverage and accuracy of sentence level sentiment analysis. The evaluation highlights the strengths and weaknesses of each model where the BERT model was the best performing overall, especially for neutral sentences, while the rule based and hybrid model were much better at positive sentences, for negative sentiment detection the hybrid SVM model was the best performing. Our thesis contributes to the ongoing discourse on effective sentiment analysis in non-English languages and offers insights for further advancements in natural language processing (NLP) for Swedish.

Keywords: Computer science, Sentiment analysis, BERT, Lexicon-based, Rule-based, Support Vector Machines (SVM), Swedish language, Natural language processing (NLP), Sparv, SALDO, SenSALDO, Machine learning.



## Acknowledgements

We would like to thank our supervisors Dana Dannélls, Niklas Zechner and Justyna Sikora for their help and guidance during the project. Their support and advice has helped us navigate during development and writing. We would also like to thank our friends and opponents for their support.

Ricardo Mansour, Gothenburg, Erik Nilsson, Gothenburg 2024-06-17





# Contents

<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xi</b>
<b>Concepts</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem statement . . . . .	1
1.2 Aim . . . . .	2
1.3 Limitations . . . . .	3
1.4 Challenges and ethical considerations . . . . .	3
<b>2 Theory</b>	<b>5</b>
2.1 Related work . . . . .	5
2.1.1 Lexicon-Based Sentiment Analysis . . . . .	5
2.1.2 Current state of Swedish language sentiment analysis . . . . .	6
2.1.3 State of the art of sentiment analysis for English . . . . .	6
2.1.4 Utilizing linguistic features . . . . .	7
2.1.5 Previous work within SVM . . . . .	7
2.1.6 Recent machine learning approaches using BERT models . . . . .	8
2.2 Sparv . . . . .	9
2.3 BERT . . . . .	10
2.3.1 Transformer architecture . . . . .	11
2.3.1.1 Attention mechanism . . . . .	13
2.3.1.2 Multi-head attention . . . . .	13
2.3.2 Pre-training of a BERT model . . . . .	14
2.3.3 Fine-tuning and Transfer learning . . . . .	15
2.3.4 KB-BERT . . . . .	15
2.4 SVM-model . . . . .	16
2.4.1 Theory of SVM . . . . .	16
2.4.2 Kernel Trick . . . . .	17
2.4.3 Multiclass SVM . . . . .	17
2.4.4 Advantages and Disadvantages . . . . .	17
2.4.5 Applications . . . . .	18
2.4.6 Performance Evaluation . . . . .	18
2.5 Evaluation methods . . . . .	18

<b>3</b>	<b>Data</b>	<b>20</b>
3.1	Lexicons . . . . .	20
3.1.1	SALDO . . . . .	20
3.1.2	SenSALDO . . . . .	21
3.2	Assembling the dataset . . . . .	22
3.2.1	Annotation and its difficulties . . . . .	23
3.3	Train, Test, and Validation Split . . . . .	23
3.3.1	Rule-Based Model Test and validation data . . . . .	24
<b>4</b>	<b>Methods</b>	<b>25</b>
4.1	Saldo and linguistic-based sentiment analysis . . . . .	25
4.1.1	Rule-based sentiment analysis and bag-of-words . . . . .	25
4.2	Our baseline . . . . .	27
4.2.1	Negations . . . . .	29
4.2.2	Expansion of the lexicon . . . . .	30
4.3	The hybrid SVM model . . . . .	31
<b>5</b>	<b>Results</b>	<b>32</b>
5.1	Rule-based model . . . . .	32
5.2	Negations . . . . .	33
5.3	The expansion of the lexicon and its contributions . . . . .	34
5.4	The KB-BERT model . . . . .	36
5.5	SVM-Hybrid Model . . . . .	38
5.6	Overview of Results . . . . .	39
5.6.1	Class-Specific Performance Metrics . . . . .	39
5.6.2	Weighted Performance Metrics . . . . .	40
<b>6</b>	<b>Discussion</b>	<b>41</b>
6.1	Comparisons of the different approaches . . . . .	41
6.2	Reasons for the model's performance . . . . .	42
6.2.1	Difficulties with classifying negative sentences . . . . .	42
6.2.2	Drop in accuracy when using the expanded lexicon . . . . .	42
6.2.3	The hybrid approach . . . . .	43
6.2.4	Other reasons and overview . . . . .	44
6.3	This thesis in context of previous studies . . . . .	44
6.3.1	Lexicon-Based Models and Their Expansion . . . . .	45
6.3.2	Simple Machine Learning Approaches . . . . .	45
6.3.3	Advanced Deep Learning Models . . . . .	45
<b>7</b>	<b>Conclusion</b>	<b>46</b>
7.1	Future work . . . . .	47
	<b>Bibliography</b>	<b>48</b>

# List of Figures

1.1	Example sentence of Sparv pipeline . . . . .	2
2.1	Example output of Sparv for a short sentence containing a negation. .	9
2.2	Part of a longer example sentence breakdown and analysis by Sparv. .	10
2.3	The Transformer architecture. The encoder on the left is in charge of processing input data, and the decoder on the right is in charge of generating the output sequence. [22] . . . . .	11
2.4	Output and input have the same sequence length and dimension. Weighs each value by similarity of the corresponding query and key. For each sequence position output, sum up the weighted values. [23] .	12
3.1	The structure of the SALDO lexicon shown by the example word yxa (axe). . . . .	21
3.2	Structure of the SenSALDO lexicon. . . . .	22
4.1	Breakdown of an example sentence with a negation . . . . .	27
4.2	The entry in SALDO of the word "adventsstjärna" (advent star) . . .	30
5.1	The confusion matrix of the baseline model . . . . .	33
5.2	The confusion matrix of the model with negations handling added . .	34
5.3	The confusion matrix of the model using the expanded lexicon . . . .	35
5.4	The confusion matrix of model using the expanded lexicon and negation handling . . . . .	36
5.5	The confusion matrix of KB-BERT model . . . . .	37
5.6	The confusion matrix of SVM-Hybrid model . . . . .	39

# List of Tables

3.1	The different sources of our dataset . . . . .	22
5.1	Class-specific performance metrics for different models . . . . .	40
5.2	Weighted performance metrics for different models . . . . .	40

# Concepts

**Natural language processing (NLP)** - A field of computer science that focuses on making computers able to understand and use human language.

**Sparv** - A text analysis tool for Swedish that can break down sentences and annotate them with information about word classes and sentence structure [1][2].

**SALDO** - An electronic lexical resource for the Swedish language that contains a plethora of words with corresponding semantic and morphological information [3].

**SenSALDO** - A lexicon for Swedish based on SALDO containing sentiment values for each word (negative -1, neutral 0, positive 1) [4][5][6].

**Sentence-based analysis** - Analysis on a sentence based level taking into account how words link to each other and syntax instead of word-based where you often just add together all the individual word sentiment scores in the sentence.

**Linguistic based model** - In the context of natural language processing (NLP) and computational linguistics, a linguistic-based model is a method or system that processes and comprehends human language primarily through the application of linguistic theories and expertise.

# 1

## Introduction

This thesis presents the innovative development of a sentiment model that employs linguistic features for sentence-level analysis and compares it to other approaches to sentiment analysis like machine-learning models.

The field of NLP witnessed a paradigm shift with the introduction of Google AI’s revolutionary Bidirectional Encoder Representations from Transformers (BERT) model, as highlighted by Devlin et al. [7]. BERT’s novel bidirectional processing, encompassing both right-to-left and left-to-right contexts, enables the recognition of complex semantic linkages and contextual nuances within text. This capability has propelled BERT to a leading position in NLP tasks, particularly in sentiment analysis, underscoring its unmatched effectiveness in comprehending and interpreting the intricacies of human language. BERT’s transformative influence in sentiment analysis epitomizes the ongoing quest for models that adeptly navigate language nuances, a quest that continuously propels NLP forward.

Our thesis sets forth two primary objectives: firstly, to develop and refine a sentiment model specifically designed for sentence-level analysis using a linguistic-based approach for Swedish that is based on the the text analysis pipeline Sparv developed by Språkbanken [1] [2]; and secondly, to conduct a comprehensive comparison of these models against advanced methodologies like BERT. Moreover, as the field has been shifting with the rise of advanced machine learning models like BERT there is a need for comparison and testing the performance of a novel sentence-level lexical model based on Sparv compared with a BERT one, especially in the context of non-English languages. There is also a general scarcity of such models for Swedish and other non-English languages.

This thesis is aligned with the evolving discourse on sophisticated NLP methods and seeks to contribute meaningfully to the broader debate within this burgeoning field. By providing empirical comparisons, our research aims to place our findings within a wider NLP context, highlighting the unique challenges and opportunities presented by the Swedish language.

### 1.1 Problem statement

Our research initially focuses on addressing the current limitations of the existing sentiment model in Sparv, which operates primarily at a word-based level of

sentiment analysis and is partly constrained by a lexicon, SenSALDO [4], with a somewhat limited amount of annotated lexical entries [4][6][5]. This calls for a substantial expansion and enhancement of this particular lexical resource to enrich the depth and accuracy of sentiment analysis.

Additionally, it is important to note that, as of now, there are a lot fewer models that perform sentiment analysis in Swedish than there are for English, especially sentence-level non-BERT models, although there are some rule-based models like VADER [8]. This often necessitates the use of models developed for languages like English. While there is a Swedish version of VADER and of the BERT model, known as KB BERT, it inherits the general limitations of BERT models. These limitations include potential suboptimal performance on languages with different grammatical structures and idiomatic expressions, such as Swedish.

Multilingual versions of BERT often struggle to capture the nuances of less commonly spoken languages due to limited training data, where more data is correlated with better performance [9]. This could lead to less accurate sentiment analysis in Swedish, especially for tasks requiring a deep understanding of language-specific subtleties.

☒ Lexikalanalys ☐ Sammansättningsanalys ☒ Dependensanalys ☒ Attitydanalys ☒ Namntaggar ☒ Läsbarhetsvärden

[Visa avancerade inställningar](#) [Återställ standardinställningar](#) [Kort](#)

[XML](#)

```
<text lix="4.00" ovix="inf" nk="0.00"> [Visa XML]
<paragraph> [Visa XML]
<sentence id="8f7-8c4"> [Visa XML]
  Jag är inte glad .
  token      msd      lemma  lex      sense  complemgram  compwf  sentimentclass  deprel
  Jag        PN. UTR. SIN. DEF. SUB  jag     jag..pn.1  jag..1                neutral         SS
  är         VB. PRS. AKT          vara    vara..vb.1  vara..1                positive        ROOT
  inte       AB                    inte    inte..ab.1  inte..1                neutral         NA
  glad       JJ. POS. UTR. SIN. IND. NOM  glad    glad..av.1  glad..1                positive        SP
  .          MAD                    .       .           .                      neutral         IP
</sentence>
</paragraph>
</text>
```

Figure 1.1: Example sentence of Sparv pipeline

## 1.2 Aim

The architecture behind Språkbanken’s corpus annotation pipeline is called Sparv [1]. Using a plain text document and the pipeline’s web interface is the most straightforward method of using it. The pipeline, as can be seen in Figure 1.1, breaks up the text into sentences and paragraphs, tokenizes, tags parts of speech, looks up terms

in dictionaries, and analyses compounds using both internal and external tools. By looking at the example, we can see that some of the words are not annotated, as mentioned before.

Another thing to consider is when negations are in play, such as when a sentence has both negative and positive semantics. To address these challenges, our next step is to implement a nuanced sentiment analysis approach, positive/negative/neutral at the sentence level. This approach aims to enhance the accuracy of sentiment detection in Swedish texts, particularly by accounting for the complexities of sentence-level semantics and the interplay of negations and mixed sentiments within a single sentence.

Research questions:

1. Can we develop a sentence-level sentiment model that exploits linguistic analysis, specifically for identifying semantic links and contextual nuances in the Swedish language?
2. How well does a linguistically motivated model perform on Swedish text as compared to a more recent BERT model?

### 1.3 Limitations

For this thesis, only the models mentioned in the methodology and theory chapters will be used, in order to limit the scope of the project to something feasible for a master thesis. There might be better models or techniques that could possibly perform better, and this is something that could be explored in further studies. Additionally, we will constrict ourselves to using only the lexicon SenSaldo as well as the expanded version of it that create.

### 1.4 Challenges and ethical considerations

A leading challenge for this project is the handling of semantic links and contextual nuances present in human language such as the negation of positive words yielding a negative meaning, as in not good being a negative sentiment about something. This is something that word-based sentiment analysis models are not able to take into account and is hoped to be a strength for the Sparv-based model.

Additionally, a challenge will be ensuring that the Sparv-based model performs well on texts from a multitude of different domains and thus a varied dataset is needed in order to accurately ascertain the versatility of the model as well as observing its ability to adapt to different linguistic contexts. The acquisition of such a varied dataset in Swedish of good quality might in and of itself pose a challenge, and methods such as translation might have to be resorted to. Furthermore, Sparv uses SenSALDO to get word sentiment scores and the fact that SenSALDO is not complete with all words, although containing a substantial amount (around 12 000),



means that there will be words with no sentiment alignment available for them and thus hampering the effectiveness of the model.

Although the rule-based model most likely will not need any powerful computer resources in order to run, and this is often held as a benefit of rule-based models, the KB-BERT-model that will be used for comparison might require a more powerful computer to run on for tuning and testing. Although it, the KB-BERT model also might not need it and that is something that will be ascertained fully when the project has commenced, and we could most likely obtain access to such resources through the university should the need arise.

As for ethical concerns, one is the risk of leaking personal data, however, from the information available the datasets that will be used for expanding the lexicon as well as training and testing are anonymized which helps mitigate this risk. Furthermore, the potential future usage of the model after development could pose some ethical concerns. If the model is implemented for public use and produces many incorrect results, there is a risk of it giving false information if it were to be used for something like opinion polling or comment filtering.

Another ethical concern can be that the KB-BERT model is pre-trained and tested on data that is not all publicly available, leading to there being a potential data bias such as over and underrepresentation of certain words in certain contexts. However, this risk is not specific to the BERT model itself but rather something to consider for any other type of black box model [10].

# 2

## Theory

In this chapter, the theory needed to understand the thesis will be covered.

### 2.1 Related work

In this subchapter, related work and previous studies will be discussed and analysed.

#### 2.1.1 Lexicon-Based Sentiment Analysis

There are many approaches to lexicon-based sentiment analysis but to perform it one needs a lexicon. An important lexicon that has more often been used by previous researchers who work on English is SentiWordNet [11], which is a lexicon that is based on WordNet [12][13]. WordNet is a database consisting of 117 000 synsets, where a synset can be described as groups of synonyms that describe distinct concepts [13]. SentiWordNet is then an extension of WordNet where each synset automatically has been assigned a sentiment annotation that ranges from 0.0 to 1.0 in the three categories: positivity, neutrality (objectivity), negativity [11]. One can use SentiWordNet in multiple ways in order to calculate the sentiment of a text, however, a common way is to sum up all the different scores and see if they cross a certain threshold and then assign the corresponding sentiment value [14].

Another lexicon-based or rule-based approach is VADER (Valence Aware Dictionary and Sentiment Reasoner)[8]. Vader uses a lexicon and a set of rules to capture sentiment conveyed in text, which is especially useful for social media material. It uses a predetermined set of lexical characteristics (words) annotated with sentiment values indicating polarity and intensity. The tool has multiple rules that account for different elements of language. Punctuation, such as exclamation marks, heightens emotional intensity. Capitalization of words indicates a greater feeling. Degree modifiers, such as "extremely" or "barely," change the strength of the feeling. Conjunctions such as "but" can change the sentiment of a phrase, and the context within a brief word window (preceding trigram) is used to refine sentiment ratings.[8]

VADER calculates four sentiment metrics for each text: positive, negative, neutral, and compound. The compound score is a normalized value between -1 (most severe negative) and +1 (most extreme positive) that represents the text's overall emotion. It is specifically designed to handle the peculiarities of social media communication, such as slang, emoticons, and acronyms.

In reality, VADER may be simply implemented in Python using the vaderSentiment module. The primary method, *polarity\_scores()*, returns the sentiment scores for the given text. This combination of a curated sentiment vocabulary and heuristic rules enables VADER to efficiently assess sentiment in a variety of text forms, with a special focus on informal and social media situations[8].

### 2.1.2 Current state of Swedish language sentiment analysis

Currently, for Swedish there is a severe paucity of both, research on different approaches to sentiment analysis, and of available models for sentiment analysis. There are however lexicons that can be used in the development of models for Swedish, like SenSALDO, which contain words labelled with the sentiments: negative, neutral, positive; corresponding to the labels -1, 0, 1 [4][6][5].

There has also been some previous research exploring machine-learning-based as well as simple word-based, bag-of-words, approaches. In one relevant project, the author used the English resource VADER and used an API to translate the lexicon's words into Swedish and used a bag-of-words approach to classify tweets as positive, neutral or negative [15]. The bag-of-words approach in this paper was searching up and getting a polarity score for each word in the tweet and then summing all scores as well as dividing by the number of polarity words; yielding the resulting polarity score. Noted however is the problem of translation and especially the automatic approach since translation is not precise which leads to errors and meanings getting lost in translation. Also noteworthy was that the accuracy of this model was between 0.4 and 0.5 which is not that accurate.

In another related project more simple machine learning models for sentiment analysis, also done on tweets, were explored, like SVM, random forests and multinomial logistic regression, and they found SVM to be the most accurate achieving an accuracy score of around 0.7 [16].

Finally, a very recent model for Swedish sentiment analysis is a BERT model developed by KBLab. Noteworthy is that it is accessible through Hugging Face and that it is trained on a very large dataset not only containing reviews but also other texts. The accuracy of the model is also reported to be quite high: 0.8 for multiclass (positive, neutral, negative) and 0.88 for binary (positive and negative) [9].

### 2.1.3 State of the art of sentiment analysis for English

The field of sentiment analysis for English is much more explored and developed, and it is the case that often people translate texts into English, or translate English sentiment lexicons into other languages in order to perform sentiment analysis for non-English languages. There are traditionally three approaches to sentiment analysis: lexicon-based, machine-learning-based and a hybrid of the two.

A paper comparing lexicon-based approaches for sentiment analysis on movie reviews found that the analysers Vader and Textblob were the best performing, reaching accuracies of 77% and 74% respectively [8].

Machine-learning-based approaches have soared in popularity recently and often achieve very high accuracy results. In a paper testing and comparing a plethora of different machine learning models for sentiment analysis on tweets in the English language, the authors found the CNN, RNN, BiLSTM to have scored accuracies of 89%, 90%, 90%. They also tested different BERT model combinations, BERT-CNN, BERT-RNN, BERT-BiLSTM, and it was found that this combination of having BERT and a machine learning neural network together surpassed the accuracy of just using the neural networks alone, where all of the three combination BERT models achieved an accuracy score of 93% [17].

#### 2.1.4 Utilizing linguistic features

Sentiment analysis has traditionally often been done on a word-based level and thus misses meaning and nuance that is spread throughout the sentences, and in developing a model with Sparv such intricacies could be taken into account. Studies investigating the use of linguistic features in aiding sentiment analysis models are quite scarce. However, a study examining the impact of utilizing linguistic features in Korean for sentiment analysis using an SVM-light model on news articles found that the inclusion of linguistic features gave a positive increase in accuracy of around 2 to 28 percentage points, which is quite significant [18].

#### 2.1.5 Previous work within SVM

Sentiment analysis, often known as opinion mining, is a branch of natural language processing (NLP) that seeks to understand the opinions and subjectivity contained in text. This section examines the use of Support Vector Machine (SVM) for sentiment analysis, building on the findings of Patil, Galande, Kekan, *et al.* [19].

Patil et al. (2014) used SVM to create a classifier for sentiment analysis of user opinions on political candidates in comments and tweets. The goal was to mark user comments as good or negative, which could subsequently be used to categorize text into sentiment classes. The study emphasizes the usage of SVMs due to their known efficacy in text categorization.

The study included many critical processes to successfully preprocess and analyze text data. Initially, the text was tokenized, which involved breaking it down into individual tokens (words or phrases) and deleting extraneous characters like punctuation marks. Following tokenization, common stop words such as "a," "the," and "and" were deleted to minimize noise and improve data interpretation. Stemming was then used to reduce words to their base forms, so standardizing variations of the same word (for example, "developed" and "developing" to "develop"). The study used Term Frequency-Inverse Document Frequency (TF-IDF) to convert the text into a machine learning-ready format. This method determines the relevance of each word in the corpus, selecting terms that are more important for processing by comparing their frequency in specific documents to their rarity across all documents.

Feature selection was critical in refining the dataset since it identified and focused on terms that have the greatest impact on positive and negative sentiment. This

stage aims to improve the classifier’s efficiency and accuracy by focusing on relevant features. The classification phase required using the SVM technique, which creates a hyperplane in an N-dimensional space to divide data points into two categories: positive and negative attitudes. The study chose a linear kernel, which is appropriate for the high-dimensional feature space found in text classification tasks.

## Results and Conclusions

The performance of the SVM classifier was assessed using precision and recall metrics. Precision quantifies the accuracy of positive predictions, whereas recall checks their completeness. The study concluded that SVM is quite useful for sentiment analysis, particularly in text categorization tasks. SVM’s capacity to handle high-dimensional feature spaces without considerable feature selection makes it ideal for huge datasets. The evaluation found that SVM has great precision and recall, demonstrating its resilience in sentiment categorization. Patil et al.’s paper gives a core understanding of SVM’s applicability in sentiment analysis. It emphasizes the relevance of preprocessing, feature selection, and TF-IDF in improving the performance of SVM classifiers. The findings of this study are critical for comparing with current research on sentiment analysis utilizing a hybrid SVM model.

The findings of this study are critical for comparing with current research on sentiment analysis utilizing a hybrid SVM model. This thesis attempts to demonstrate advancements in sentiment analysis, particularly in multilingual environments, by comparing the results of the hybrid SVM model to those of Patil et al. This comparison will assist in understanding the gains and advancements made in the use of SVM for sentiment analysis in the Swedish language. It will also emphasize the effectiveness of hybrid models in boosting sentiment classification accuracy and resolving constraints found in prior studies [19].

### 2.1.6 Recent machine learning approaches using BERT models

Recently, the NLP field has been inundated with machine-learning models and especially Bidirectional Encoder Representations from Transformers (BERT) models [20]. Such models offer both benefits and drawbacks and a study looking at both the advantages and disadvantages of them found that an advantage is the enabling of end-to-end training where one can skip a lot of the middle steps of model development, thus facilitating for developers.

However, previous studies also pointed out that natural language data follows a power law distribution, meaning that the vocabulary increases as the dataset increases and there will thus always be words not represented in the dataset, which is a problem. Other problems discussed were the need for large datasets and powerful computers, as well as deep machine learning models lack of interpretability [21].

Lexicon-based or rule-based approaches are well interpretable and quite fast, and generally do not need a lot of computing power to run. A study comparing lexicon-based models to BERT ones for the Russian language found that although Ru-

BERT outperformed the best-performing lexicon-based method with SO-CAL on most datasets, SO-CAL was not far behind and even outperformed RuBERT on four of the datasets used [20]. This thus indicates that there is a promising path to explore further, especially if also combining with the inclusion of linguistic features that can be done with Sparv.

## 2.2 Sparv

Sparv is an annotation pipeline and text analysis tool that has the capacity to break down sentences and give annotations on the different parts of it [2][1]. It currently supports 21 different language options, where it has two for Swedish: Swedish and 1800's Swedish. Swedish is also the language that has the most developed support [1].

As seen in Figure 2.1 and Figure 2.2 below, the annotation is composed of nine different categories that the user can choose from where the most important for this work are: token (the word or symbol), msd (morpho-syntactic tag), lemma (the root form of the word, for example without the inflection for plural or definiteness), lex, part of speech (noun, adjective verb etc.), sense id (which link to the word in lexicons and gives its meaning), prefix, suffix, sentimentclass (the sentiment value of the word as positive, negative or neutral), deprel (dependency relation which tells us how the word is related to the others in the sentence).

☒ Lexikalanalys 
 ☐ Sammansättningsanalys 
 ☒ Dependensanalys 
 ☒ Attitydanalys 
 ☒ Namntaggare 
 ☒ Läsbarhetsvärden

```

<text lix="5.00" ovix="inf" nk="0.33"> [Visa XML]
<paragraph> [Visa XML]
<sentence id="8f7-8fd"> [Visa XML]

```

token	msd	lemma	lex	sense	complemgram	compwf	sentimentclass	deprel
Han	PN. UTR. SIN. DEF. SUB	han	han..pn.1	han..1				SS
gillar	VB. PRS. AKT	gilla	gilla..vb.1	gilla..1 (0.907), gilla..2 (0.093)			positive	ROOT
inte	AB	inte	inte..ab.1	inte..1				NA
denna	DT. UTR. SIN. DEF	denna	denna..pn.1	denna..1				DT
boken	NN. UTR. SIN. DEF. NOM	bok	bok..nn.1, bok..nn.2	bok..1 (0.908), bok..2 (0.092)			neutral	OO
.	MAD							IP

```

</sentence>
</paragraph>
</text>

```

Figure 2.1: Example output of Sparv for a short sentence containing a negation.

PP: Preposition

token	msd	lemma	lex	sense	complemgram	compwf	sentimentclass	deprel
<b>&lt;ne ex="TIMEX" type="TME" subtype="DAT"&gt; [Visa XML]</b>								
I	PP	i, i fredags	i..pp.1, i_fredags..abm.1	i..2, i_fredags..1				TA
fredags	NN. UTR. SIN. IND. GEN	fredag, i fredags:01	fredag..nn.1, i_fredags..abm.1	fredag..1, i_fredags..1	freda..vb.1+ag..nn.1	fred+ags	neutral	HD
<b>&lt;/ne&gt;</b>								
ändrade	VB. PRT. AKT	ändra, ändra sig	ändra..vb.1, ändra_sig..vbm.1	ändra..1, ändra_sig..1			neutral	ROOT
sig	PN. UTR+NEU. SIN+PLU. DEF. OBJ	sig, ändra sig:03	sig..pn.1, ändra_sig..vbm.1	sig..1, ändra_sig..1				SS
<b>&lt;ne ex="ENAMEX" type="PRS" subtype="HUM"&gt; [Visa XML]</b>								
Lars	PM. NOM	Lars	Lars..pm.1	Lars..1				XX
Ohly	PM. NOM							HD
<b>&lt;/ne&gt;</b>								
,	MID							IK
<b>&lt;ne ex="TIMEX" type="TME" subtype="DAT"&gt; [Visa XML]</b>								
Nu	AB	nu	nu..ab.1	nu..1				TA
<b>&lt;/ne&gt;</b>								
vill	VB. PRS. AKT	vilja	vilja..vb.1	vilja..1			neutral	TA
han	PN. UTR. SIN. DEF. SUB	han	han..pn.1	han..1				SS
bli	VB. INF. AKT	bli	bli..vb.1	bli..1 (0.878), bli..2 (0.122)			neutral	VG
<b>&lt;ne ex="ENAMEX" type="ORG" subtype="PLT"&gt; [Visa XML]</b>								
vänsterpartiets	NN. NEU. SIN. DEF. GEN	vänsterparti	vänsterparti..nn.1	vänsterparti..1	vänster..nn.1+parti..nn.1, vänster..nn.2+parti..nn.1, vänster..av.1+parti..nn.1, vänster..nn.2+parti..nn.1+i..nn.1, vänster..nn.1+parti..nn.1+i..nn.1, vänster..av.1+parti..nn.1+i..nn.1	vänster+partiets, vänster+part+iets	neutral	DT
<b>&lt;/ne&gt;</b>								
ledare	NN. UTR. SIN. IND. NOM	ledare	ledare..nn.1	ledare..1 (0.978), ledare..3 (0.019), ledare..2 (0.003)			neutral	SP
.	MAD							IP
<b>&lt;/sentence&gt;</b>								

Figure 2.2: Part of a longer example sentence breakdown and analysis by Sparv.

## 2.3 BERT

BERT is a language model developed by Google in 2018. It is based on the Transformer architecture, which is a neural network architecture that is particularly well-suited for natural language processing tasks. With directional models, the text input is read either left-to-right or right-to-left, but with the Transformer encoder, all words are read simultaneously. It would be more accurate to describe it as non-directional, but as a result, it is seen as bidirectional. Because of this feature, the model is able to understand a word's context by taking into account both the word's left and right surroundings.

BERT uses two training techniques: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In MLM, a portion of words in a sentence are randomly masked, leaving blank spaces in their place. The model is then tasked with predicting the original, masked words. In NSP, the model is presented with pairs of sentences and asked to determine whether the second sentence logically follows the first. Token, segment, and positional embeddings are just a few of the many embedding layer kinds that are used to encode the information in order to comprehend the context of words. The original BERT model, also known as BERT-Base, contains 12 layers, but the bigger BERT model, known as BERT-Large, has 24 levels.

BERT is trained on a massive dataset of text, and it can access the entire context of a sentence when generating its predictions. As a result, there is no need for training an entirely new BERT model from scratch. This approach is commonly known as transfer learning, and it allows the system to leverage previously acquired knowledge to solve other specific tasks.

### 2.3.1 Transformer architecture

Transformer is a neural network architecture and was the first of its kind to model contextual language without the need of recurrent neural network models also known as RNNs, such as long short-term memory (LSTM) and gated recurrent unit (GRU).

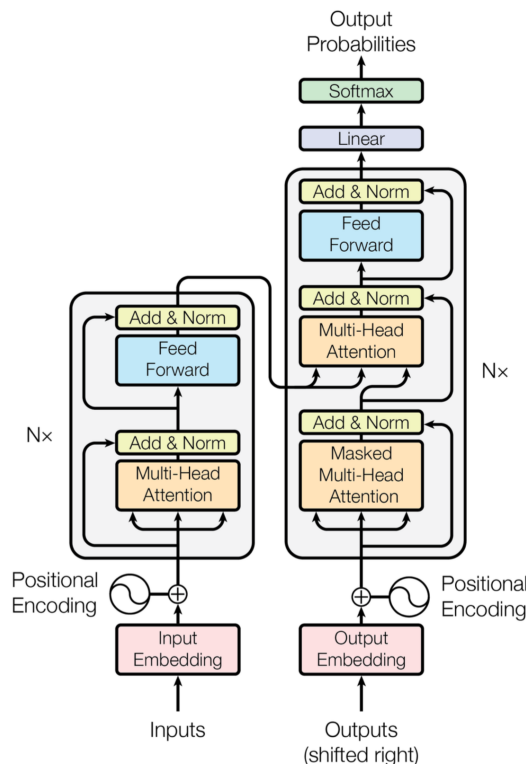


Figure 2.3: The Transformer architecture. The encoder on the left is in charge of processing input data, and the decoder on the right is in charge of generating the output sequence. [22]



There are encoder and decoder pieces in a transformer. While the decoder performs the opposite function, the encoder, as its name suggests, converts phrases and paragraphs into an internal structure (a numerical matrix) that comprehends the context.

Sequence-to-sequence activities, like translation, can be carried out by a transformer by combining the encoder and decoder. Eliminating the transformer's encoder section can provide you with interesting context-related information. By utilizing the attention model, the Bidirectional Encoder Representation from Transformer (BERT) is able to obtain a more profound comprehension of the language environment. A stack of several encoder blocks makes up BERT. As with the transformer model, the input text is divided into tokens, and at the BERT's output, each token is converted into a vector.

When looking at 2.4 one of its integral parts is the self-attention mechanism, the self attention mechanism allows the model to focus on different parts of the input sequence as it processes each token. Self-Attention compares all input sequence members and alters the appropriate output sequence locations as a result. In other words, the self-attention layer searches the input sequence for each input and adds the results to the output sequence [22].

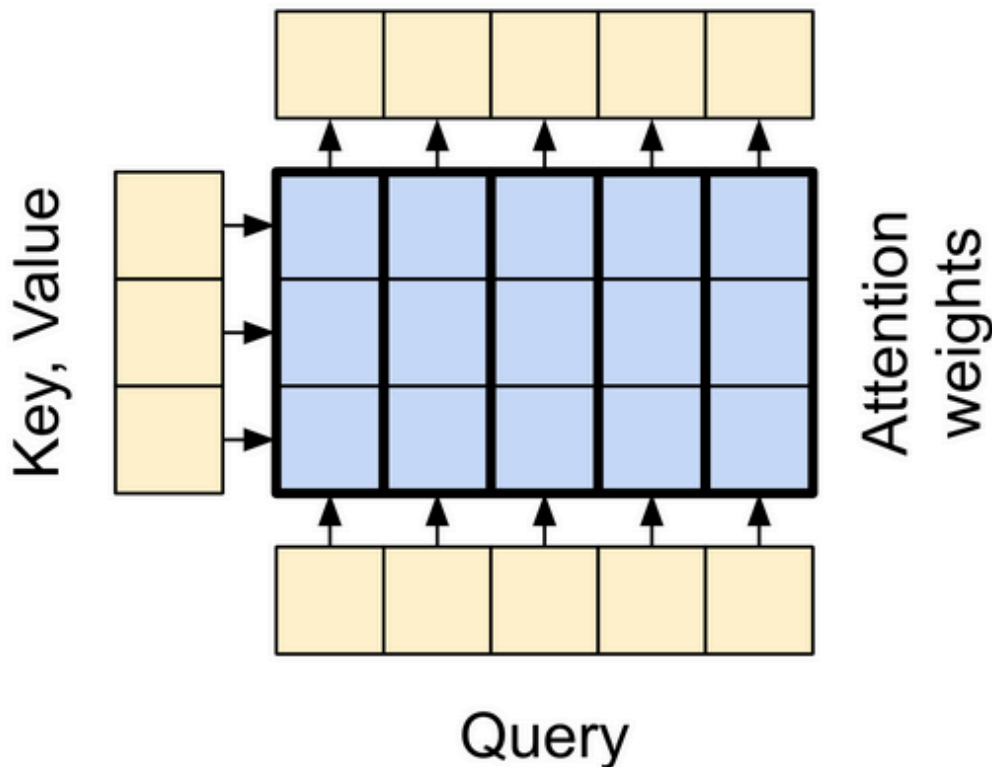


Figure 2.4: Output and input have the same sequence length and dimension. Weights each value by similarity of the corresponding query and key. For each sequence position output, sum up the weighted values. [23]

### 2.3.1.1 Attention mechanism

The transformer eliminates irrelevant information and prioritizes relevant information. Attention maps a query and associated key-value pairs to an output. The output is calculated as a weighted total of values, indicating the significance of each in the context of the query [22]. The encoder and decoder are combined in the transformer model to build a seq2seq model, allowing you to conduct translations, such as from English to Swedish, as previously demonstrated. We compute the attention function on many queries at the same time, resulting in a matrix  $Q$ . The keys and values are also organized into matrices  $K$  and  $V$ . We compute the output matrix as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_K}})V \quad (2.1)$$

In this equation,  $Q$ ,  $K$ , and  $V$  denote Query, Key, and Value, respectively. The dimensionality of the key vectors,  $d_k$ , is utilized to scale down the dot products of the query and key vectors. Scaling down the dot product is crucial when it exceeds a certain threshold. Otherwise, the softmax function will result in vanishing gradients, which can inhibit effective learning.

### 2.3.1.2 Multi-head attention

In the self-attention mechanism, the transformer employs an extension of the process known as Multi-head attention. The Multi-head attention model captures multiple characteristics of the input sequence. Multi-head attention involves operating multiple self-attention processes in parallel. Each head focuses on a distinct segment of the input sequence and generates a unique set of output vectors. These output vectors are then concatenated and multiplied with a learnt matrix  $W^O$  to form the final multi-head attention output. It can be shown in this equation:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \quad (2.2)$$

Where

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (2.3)$$

Based on the equation  $head_i$  there are three methods for extracting the matrices  $Q$ ,  $K$ , and  $V$ . The transformer employs the multi-head attention mechanism, which is further explained below.

#### Encoder-decoder attention

In "encoder-decoder attention" layers, queries are sourced from the previous decoder layer, whereas memory keys and values are derived from encoder output. This permits each place in the decoder to cover all positions in the input sequence. This behaviour is similar to the attention mechanisms used in sequence-to-sequence models.

#### Encoder self-attention

The encoder uses this attention head to grasp the context and attend to all places in the preceding layer. Matrices  $Q$ ,  $K$ , and  $V$  from the previous encoder layer are used for this form of attention.

**Decoder self-attention**

Self-attention layers in the decoder ensure that each position attends to all positions, up to and including its own. This restricts attendance to current and former roles, preventing future opportunities. The transformer’s important feature ensures that predictions for certain positions are purely dependent on prior information. Matrices  $Q$ ,  $K$ , and  $V$  from the preceding decoder layer are used for this form of attention.

**2.3.2 Pre-training of a BERT model**

Pre-training is an important stage in the construction of BERT (Bidirectional Encoder Representations from Transformers) models, which are a powerful family of language models that have made substantial contributions to the field of natural language processing. Pre-training entails training the model on a large dataset of unlabelled text, allowing it to acquire basic language representations and word relationships. These learnt representations serve as the foundation for fine-tuning the model in subsequent tasks such as question answering, sentiment analysis, and text summarization [7].

As mentioned above, pre-training consists of 2 steps MLM and NSP:

**Masked Language Modelling (MLM)**

Masked Language Modelling (MLM) is an important pre-training technique for BERT models that captures the subtle interactions between words and their contextual dependencies in natural language. In MLM, a subset of words in a sentence are randomly replaced with specific masks or placeholders, and the model is tasked with predicting the original, masked words based on the context.

This masking process serves two primary purposes:

First, focus on understanding meaning and relationships: By eliminating words at random and challenging the model to predict them, MLM forces the model to focus on comprehending the semantic meaning and links between words rather than simply memorizing their order. This helps the model to use contextual cues like synonyms, antonyms, and grammatical relationships to accurately guess the missing words.

Second, MLM increases the model’s robustness to noise and out-of-vocabulary words. By encountering masked words, the model learns to deal with missing information and adapt to new words that may not appear in the training data.

This method of predicting masked words efficiently educates the BERT model to understand the underlying structure and meaning of language, allowing it to perform a variety of NLP tasks. BERT can better read and generate natural language by recognizing word relationships and sentence structure.

**Next Sentence Prediction (NSP)**

Next Sentence Prediction (NSP) is another essential pre-training technique for BERT models, which focuses on understanding the flow and coherence of language. In NSP, pairs of sentences are presented to the model, and it is asked to predict whether the second sentence logically follows the first.

This task challenges the BERT model to not only identify individual word meanings but also to grasp the overall context and relationships between sentences. By determining whether the second sentence logically follows the first, the model learns to distinguish between connected and unrelated sentences, improving its understanding of the structure and flow of language.

NSP also enhances the model’s ability to capture long-range dependencies, which are crucial for understanding complex sentences and paragraphs. By recognizing the coherence between sentences, BERT can better grasp the overall meaning and intent of a text passage.

### **The Combined Effectiveness of MLM and NSP**

The combination of Masked Language Modelling (MLM) and Next Sentence Prediction (NSP) creates an effective synergy for pre-training BERT models. MLM focuses on recording word relationships, whereas NSP emphasizes language flow and coherence.

Together, these strategies allow BERT to excel at a variety of NLP tasks, including as question answering, sentiment analysis, and text summarization. By efficiently grasping the intricacies of language, BERT can deliver more accurate and nuanced predictions, making it a helpful tool for natural language processing.

According to Devlin et al. (2019), BERT’s pre-training strategies, particularly MLM and NSP, have dramatically enhanced NLP model performance across a variety of tasks. Clark et al. (2020) show that various MLM strategies can be improved to improve BERT’s pre-training process, whereas Song et al. (2019) indicate that MLM and NSP are successful for unsupervised sentence representation learning.

Rajpurkar et al. (2016) introduce the SQuAD dataset, which has become a standard for assessing BERT models’ performance on question-answering tasks. The combination of MLM and NSP has established a new standard for language representation learning, paving the door for future advances in natural language processing.

### **2.3.3 Fine-tuning and Transfer learning**

BERT models benefit considerably from fine-tuning, which involves training a pre-trained model on a smaller, domain-specific dataset. This method is a type of transfer learning in which the model modifies its general skills to do specific tasks well. Our model was fine-tuned to better handle the intricacies of Swedish, enhancing its performance on sentiment analysis tasks by adjusting to the Swedish language’s distinctive qualities, such as idioms and syntax.

### **2.3.4 KB-BERT**

BERT models differ not only in size, such as BERT-Base and BERT-Large [7], but also in the domain-specific training adaptations. For example, BERT has been specialized into models such as Sentence-BERT, which is trained particularly for sentence-level tasks [24], and BioBERT, which is optimized for biomedical literature [25].

The model we use is the KB-BERT model for multiclass Swedish sentiment analysis is trained using a broad and extensive dataset. This dataset has multiple text sources, each of which contributes to the model’s capacity to reliably estimate sentiment in a variety of scenarios.

One substantial chunk of the dataset consists of 13,000 Trustpilot reviews. These evaluations span a wide range of items and services, and the star ratings range from one to five. Originally, the dataset omitted evaluations with a three-star rating, which served as impartial proxies. To close the gap, neutral reviews were added to the corpus via extra scraping. This updated Trustpilot scraper is now free to use.

In addition, the collection contains 4,000 manually annotated news headlines from KBLab. Due to copyright restrictions on the underlying content, this dataset cannot be shared.

The dataset includes 5,000 texts from Språkbankens ABSAImm corpus, specifically on immigration. These writings come from both news sites and social media, providing a diverse range of opinions on this important issue.

The collection includes 40,000 machine-translated reviews from the Norwegian Review Corpus (NoReC). These texts were translated using CTranslate2 and a model from Helsinki NLP. While the translations have not been formally evaluated, trials show that they improve sentiment classification. To validate the model’s performance, it was tested on 1,000 headlines collected during the project mentioned in a related blog post. These headlines were evaluated at the sentence level, ensuring the model’s robustness in sentiment analysis[26].

## 2.4 SVM-model

Support Vector Machines (SVM) are a class of supervised learning models used for classification, regression, and outlier detection. They are particularly known for their application in classification tasks. Introduced by Vladimir Vapnik and colleagues, SVMs aim to find the optimal hyperplane that separates data points of different classes with the maximum margin [27].

### 2.4.1 Theory of SVM

SVM operates by transforming the input data into a high-dimensional feature space using a kernel function, enabling it to handle both linear and non-linear data. The primary goal of SVM is to construct a hyperplane that maximizes the margin between two classes. This margin is the distance between the hyperplane and the closest data points from each class, known as support vectors.

The mathematical representation of the hyperplane is given by:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{2.4}$$

where  $\mathbf{w}$  is the weight vector,  $\mathbf{x}$  is the input feature vector, and  $b$  is the bias term.

The optimization objective is to minimize the norm of the weight vector  $\|\mathbf{w}\|$ , subject to correctly classifying all training samples:

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.5)$$

subject to:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i \quad (2.6)$$

where  $y_i$  is the class label of the  $i$ -th training sample [27].

### 2.4.2 Kernel Trick

To deal with non-linearly separable data, SVM uses a technique known as the kernel trick. This involves using a kernel function to map the input data into a higher-dimensional space where a linear hyperplane can effectively separate the classes. Common kernel functions include:

- **Linear kernel:**  $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$
- **Polynomial kernel:**  $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$
- **Radial Basis Function (RBF) kernel:**  $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$
- **Sigmoid kernel:**  $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x}_j + c)$

### 2.4.3 Multiclass SVM

Although SVM is fundamentally a binary classifier, it can be extended to handle multiclass problems through strategies such as:

- **One-vs-One (OvO):** Constructs a binary classifier for each pair of classes, resulting in  $\frac{k(k-1)}{2}$  classifiers for  $k$  classes. During prediction, a majority voting scheme is used.
- **One-vs-All (OvA):** Constructs a binary classifier for each class against all other classes, resulting in  $k$  classifiers. The class with the highest confidence score is chosen during prediction.

### 2.4.4 Advantages and Disadvantages

SVM has several advantages:

- Effective in high-dimensional spaces and with datasets where the number of dimensions exceeds the number of samples.
- Memory efficient by using a subset of training points (support vectors) in the decision function.
- Versatile due to the variety of kernel functions that can be specified.

However, SVM also has some disadvantages:

- Poor performance with very large datasets due to its quadratic optimization problem.
- Less effective on noisy data with overlapping classes.
- Requires careful tuning of hyperparameters and selection of the appropriate kernel function, which can be computationally intensive.

### 2.4.5 Applications

SVMs are widely used in various fields, including:

- **Text and Hypertext Categorization:** Classifying documents into predefined categories.
- **Image Classification:** Categorizing images based on their content.
- **Bioinformatics:** Protein classification and cancer detection.
- **Handwriting Recognition:** Recognizing handwritten digits and characters.

### 2.4.6 Performance Evaluation

The performance of an SVM model is often evaluated using metrics such as accuracy, precision, recall, and F1-score. Additionally, confusion matrices are useful for detailed analysis of classification performance. Cross-validation techniques are often employed to ensure the model generalizes well to unseen data.

In summary, SVMs are a robust and versatile tool for classification tasks, capable of handling both linear and non-linear data through the use of kernel functions. While they have certain limitations, their strengths make them a valuable addition to the machine-learning toolkit [27].

## 2.5 Evaluation methods

### 1. Accuracy

- **Description:** Accuracy quantifies the overall ability of the model to correctly identify both positives and negatives. It is defined as the proportion of true results (both true positives and true negatives) among the total number of cases examined. It is particularly effective as a measure when the classes are nearly balanced [28].

- **Formula:**

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

Where:

\*  $TP$  = True Positives

- \*  $TN$  = True Negatives
- \*  $FP$  = False Positives
- \*  $FN$  = False Negatives

## 2. Precision

- **Description:** Precision, also referred to as the Positive Predictive Value, measures the accuracy of positive predictions. It reflects the proportion of positive identifications that were actually correct, highlighting the model's ability to not label negative samples as positive [28].

- **Formula:**

$$\text{Precision} = \frac{TP}{TP + FP}$$

## 3. Recall

- **Description:** Recall, or Sensitivity or True Positive Rate, measures the model's ability to capture all relevant instances in the dataset. It represents the proportion of actual positives that were correctly identified, thus indicating the effectiveness of the model in identifying positive results [28].

- **Formula:**

$$\text{Recall} = \frac{TP}{TP + FN}$$

## 4. F1 Score

- **Description:** The F1 Score is the harmonic mean of Precision and Recall. This score is particularly useful when the costs of false positives and false negatives are high and when the classes are imbalanced. The F1 Score provides a balance between Precision and Recall by taking their harmonic mean, capturing the trade-off between these metrics [28].

- **Formula:**

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$



# 3

## Data

In this chapter the data that will be used and other types of data, like the lexicons, that are relevant to this study will be presented. Some difficulties encountered in the assembling and annotation of the dataset will also be discussed.

### 3.1 Lexicons

When developing the rule-based model we utilized the lexicon SenSALDO which, contains words along with their sentiment annotation. SALDO, on which the SenSALDO lexicon is based, is what was used for expanding the SenSALDO lexicon. These will be explained and described in this sub-chapter.

#### 3.1.1 SALDO

In the SALDO lexicon [3][29][30], each entry, which is a word sense [30] that is to say a word but words with different meanings like homographs are distinguished by numbers at the end like: "*däck..1*" (tyre) and "*däck..2*" (deck). There are a total of 137 128 entries. Each word is also annotated with its lemma form [29], which can be thought of as the dictionary form, where a word like "*drake..1*" (dragon), and "*drake..2*" (kite) share the same lemma form "*drake..nn.1*" most likely due to their shared inflection, same plural and definite forms et cetera.

Words with different inflections also have different lemma forms, like "*damm*" (dust) and "*damm*" (dam) have the lemma forms "*damm..nn.2*" and "*damm..nn.1*" respectively. However, the two entries for "*däck*" have different lemma forms, "*däck..nn.1*" and "*däck..nn.2*", even though they share the same inflection. The fact that the two entries for "*däck*" do not share the lemma form, even though they have the same inflection, while the two entries for "*drake*" do, is not fully explained from what we could find.

Each word entry also has the plain word, like "*drake*" "*drake..1*", and its part of speech. The parts of speech classes are: noun (nn), proper name (pm), adjective (av), pronoun (pn), counting word (nl), verb (vb), adverb (ab), preposition (pp), conjunction (kn), subjunction (sn), infinitive marker (ie), interjection (in) [29]. Multi-word entries are denoted as the class of the base word followed by

an "m" e.g. "vbm", abbreviations are denoted by the base word followed by an "a" [29].

SALDO also provides semantic descriptors, both primary and secondary, that categorize and define the word's semantic attributes [29][3][30]. For example, the Swedish word *stenyxa* translates to stone axe in English, where *sten* (stone) is the secondary descriptor indicating the material, and *yxa* (axe) is the primary descriptor indicating the type of tool. The secondary descriptor is optional while the primary is obligatory and has to be more general than the word itself [29][30]. This leads to a hierarchical structure where a top node is provided, "*PRIM*", that serves as the primary descriptor for words that cannot be further generalized [30][29].

Furthermore, the lexicon links to *children* of the word, which are semantically related words derived from the primary and secondary descriptors and an example of which can be seen in 3.1. This structure enhances the richness of linguistic analysis by showing not only the direct meanings but also associated concepts that help in understanding language usage [3].

SALDO ▾ 2 TRÄFFAR (VISAR 2)						
	BETYDELSE	LEMGRAM	ORDKLASS	PRIMÄR	SEKUNDÄR	BARN (PRIMÄRA) BARN (SEKUNDÄRA)
⚙	yxa	yxa (substantiv) ✎ ...	substantiv	hugga		bila bronsyxa bödelsyxa celt däxel fler...
⚙	yxa till	tillyxa (verb) ✎ ... yxa till (verb) ✎ ...	verb	forma yxa		grovyxad tillyxande tillyxning

Figure 3.1: The structure of the SALDO lexicon shown by the example word *yxa* (axe).

### 3.1.2 SenSALDO

The SenSALDO lexicon as shown in the picture below 3.2 contains 12 287 word entries which are the entered as the sense ID they would have in the SALDO lexicon. Each entry is labelled with its polarity, polarity label, ranging from negative to neutral to positive (-1, 0, 1) [4] [6] [5]. From our analysis of the lexicon it seems to contain: 6 001 neutral words, 4 273 negative words and 2 013 positive words.

```

# Data format:
# <SALDO sense ID>tab<polarity label>
# <polarity label> = "-1" (negative) | "0" (neutral) | "1" (positive)
#
#
absolut..1      0
absolution..1   1
abstinens..1   -1
abstinensbesvär..1   -1
absurd..1      -1
absurditet..1  -1
acceptera..1    1
accepterande..1 1
ackuratess..1   1
adelsmärke..1   1
administration..1  0
administrativ..1  0

```

Figure 3.2: Structure of the SenSALDO lexicon.

## 3.2 Assembling the dataset

The dataset was manually assembled and annotated, consisting of 2 001 sentences from different sources, mainly from news articles and blog posts but also from other datasets available on Språkbanken’s website [31]. The dataset was also complemented by some manually crafted sentences written by us. This was mainly done because we wanted to be able to test specific things. For example, different types of negations and negative sentences written without using overtly negative words. The dataset is further made up of the three classes: positive, neutral and negative. Each class has 667 entries summing up to a total of 2 001 sentences. Further analysis showed that the dataset is comprised of 28 755 words and that the average word per sentence is 13.37.

The sources of the dataset are represented in the table 3.1 where we can see that the main four types of sources used are: news articles, blog posts, Swedish FrameNet and sentences manually created by us.

Type	Source	Entries (sentences)
News	8 Sidor	611
	Dagens Arena	44
	SVT Nyheter	59
Blog posts	Bloggmix	582
	Flashback	87
Own	Own sentences	297
Miscellaneous	Swedish FrameNet	313
	ASPAC	4
	August Strindbergs brev	4

Table 3.1: The different sources of our dataset

### 3.2.1 Annotation and its difficulties

Sentiment analysis, also called opinion mining, is about finding and classifying the sentiment or opinion of a text [32]. However, defining what an opinion actually is is something that is not as straightforward as it could seem. Opinions can be seen as being of mainly two categories: beliefs like "I believe he will win the election." and judgements like "Unfortunately he won.", they can also be of both categories at the same time like "I think he is smart." [33]. Furthermore, it has been observed that while human annotators often agree on judgement statements being opinions, they often disagree when it comes to belief opinions whether or not the sentence in question is or is not opinionated and carries a sentiment [33].

Additionally, something that is viewed as being an opinion in one domain might be considered a factual statement, not an opinion, in another domain [33]. In a study by Kim and Hovy, an example given of such a sentence is "The screen is very big." which for a review of computers would carry a positive sentiment but in a newspaper might just be a factual statement [33]. Furthermore, a sentence's sentiment is also tied with perspective, for example, the sentence "The president died" can be negative, neutral or positive depending of the perspective of the writer and reader [34].

When we then manually annotated the dataset of collected sentences, we faced many of these issues like distinguishing between objectivity and subjectivity, whether or not a sentence does or does not carry an opinion. Since a large portion of the sentences were collected from news articles, the requirement to distinguish between the author's perspective and the objective content of the news stories provided us with a new perspective on the annotation process and forced us to reconsider how we approached this subject. Sentiment analysis for basic sentences with clear sentiments is easy and the sentiments are easy to recognize and annotate, but more objective-written material is difficult even for us as human annotators, especially since we have limited previous annotation experience and there were well over 200 such hard-to-classify sentences.

Furthermore, the inherent uncertainty in the sentiment of several words complicated the study. When phrases contained delicate or mixed emotions, it became difficult to categorize them as strictly positive, negative, or neutral.

## 3.3 Train, Test, and Validation Split

In this study, the dataset used for training and testing the Hybrid SVM model is derived from the data we annotated ourselves. This dataset was partitioned into training and test sets to evaluate the model's performance effectively.

For the Hybrid SVM model, we employed a train-test split approach. Specifically, the data was divided into 80% for training and 20% for testing. This split ensures that the model has sufficient data to learn from while also maintaining a separate set of data for unbiased evaluation of the model's performance.

**Training Set:** The training set comprises 80% of the annotated dataset. This portion of the data is used to train the Hybrid SVM model, allowing it to learn the patterns and relationships between the features and the sentiment labels. Thus the training set contains 1 601 entries and the distribution of classes was: 549 neutral, 526 negative, 526 positive.

**Test Set:** The test set consists of the remaining 20% of the annotated dataset. This set is reserved for evaluating the model after training, providing an assessment of its predictive capabilities. Thus the test set contains 401 entries and the distribution of classes was: 118 neutral, 141 negative, 142 positive.

### 3.3.1 Rule-Based Model Test and validation data

For the rule-based model, the same test dataset was used as for the Hybrid SVM model, however, a different split was used: 20% of the dataset was earmarked for validation and the remaining 80% for testing. This is because the rule-based model does not require any training data but some data is needed in order to evaluate the model while developing it, the validation set.

# 4

## Methods

In this chapter the methods will be explained, covering topics like bag-of-words, the lexicons, our baseline and iterative development strategy as well as a description of our dataset.

### 4.1 Saldo and linguistic-based sentiment analysis

In this subchapter the lexicons and ruled-based sentiment analysis will be discussed and described.

#### 4.1.1 Rule-based sentiment analysis and bag-of-words

Rule-based sentiment analysis is performed through predefined rules, unlike machine learning approaches that incorporate stochastic elements. A commonly used baseline method in this approach is the Bag-of-Words (BoW) model, where sentences are decomposed into individual words, each assigned a sentiment score based on a lexicon. These scores are summed, and the final sentiment value is normalized by the number of polarity words to accurately reflect the sentiment intensity of the text [15].

The BoW model simplifies text analysis by converting data into a collection of words, ignoring grammar and word order but retaining the frequency of occurrence.

#### Our BOW model

For sentiment analysis, words are scored using the SenSALDO lexicon, which categorizes words as positive, neutral, or negative. The sentiment of a text is determined by aggregating these scores, ensuring the analysis accounts for text length and avoids bias towards longer texts.

The calculation of the sentence score which was used in the beginning is outlined in the formula below, 4.1. The word scores are summed up and divided by the number of sentiment-carrying words. The sentiments of the words that are affected by negations are also inverted and this is done by using the Sparv sentence breakdown. In order to know which words are negations, and other

such types of words that we want specific cases for, a word list will have to be created.

$$S = \frac{\sum_{i=1}^N (score(w_i) \times neg(w_i))}{N} \quad (4.1)$$

- $S$  is the sentiment score of the sentence.
- $score(w_i)$  represents the sentiment score of the  $i^{th}$  word in the sentence.
- $neg(w_i)$  is a function that returns -1 if the  $i^{th}$  word is within the scope of a negation, effectively inverting its score, and 1 otherwise.
- $N$  is the total number of sentiment-carrying words in the sentence, excluding neutral or non-sentiment words for the purpose of this calculation.

Later, we changed another formula to instead sum up the scores of all the words in the sentence and divide by the sum multiplied by itself, plus a value called alpha that helps regulate the aggressivity normalization.

$$S = \frac{sum}{\sqrt{(sum \times sum) + alpha}} \quad (4.2)$$

- $S$  is the sentiment score of the sentence.
- $sum$  represents the sum of the scores in the sentence, also taking into account negations. Like the sum in the first formula 4.1.
- $alpha$  a value used to regulate the normalization, a higher value leads to more normalization while a lower value leads to less.

As can be seen in Figure 2.1, the negation word "*inte*" is connected to the root word of the sentence "*gillar*", which carries a positive sentiment score. To implement a rule for negations one has to change the sentiment score of the word connected to the negation. This is more straight-forward for the specific sentence above however, for a sentence such as with the verb "*att vara*" ("*to be*") like in 1.1, the negation is connected to the root which is the verb "*är*" and the adjective is also connected to the root. In this case the word that we want to negate the sentiment of is "*glad*" which then is not directly connected with the negation but indirectly through the root. This then poses a somewhat larger difficulty than for the first sentence example.

Another such difficulty is when we have a sentence with a negative determiner like *ingen* or *inga*. For example *Jag har inga pengar.* or, as below in 4.1, *Jag har ingen vän.* The sentiment score of the sentence is negative, so we reverse the score of the word *vän*. The reason is that the sentence expresses the fact that you do not have a friend. Although the determiner is connected to the word *vän* simplifying the association, the challenge remains to differentiate within the determiner class between negating words and non-negating words, such as *många* (many).

```
<text lix="4.00" ovix="inf" nk="0.50"> [Visa XML]
```

```
<paragraph> [Visa XML]
```

```
<sentence id="8f7-8c4"> [Visa XML]
```

token	msd	lemma	lex	sense	complemgram	compwf	sentimentclass	deprel
Jag	PN. UTR. SIN. DEF. SUB	jag	jag..pn.1	jag..1				SS
har	VB. PRS. AKT	ha	ha..vb.1	ha..1 (0.667), ha..3 (0.333)			neutral	ROOT
ingen	DT. UTR. SIN. IND	ingen	ingen..pn.1	ingen..1				DT
vän	NN. UTR. SIN. IND. NOM	vän	vän..nn.1	vän..1			positive	OO
.	MAD							IP

```
</sentence>
```

```
</paragraph>
```

```
</text>
```

Figure 4.1: Breakdown of an example sentence with a negation

## 4.2 Our baseline

Our project adopts a systematic iterative methodology to progressively enhance the accuracy of sentiment analysis models. This approach is underpinned by the following structured phases:

### First Phase: Assembling a dataset

A good quality test dataset will be crucial for accurately measuring and assessing the performance of both the rule-based linguistic model as well as the BERT model. Since there, to our knowledge, does not exist an expansive, good quality dataset for Swedish sentiment analysis, annotated on a sentence level, we have to assemble one ourselves.

To do this we utilize a collection of different datasets available from Språkbankens website, such as 8Sidor, SVT Nyheter, Dagens Arena, Sentimentlex [35], among others. We chose to try to avoid using text from more informal websites like Flashback forum or similar since posts there might contain a lot of slang, misspellings and incorrect language that might not be optimal for the test dataset.

Since these datasets are not annotated for sentiment on a sentence level, we have to annotate the entries manually in order to compile a dataset. We primarily use sentences sourced from the references mentioned above, with the addition of some sentences that we will construct ourselves. Our goal is to have a minimum of 2 000 labelled sentences in the test dataset, and this could then if time is available be expanded further. The sentences are annotated with -1, 0, 1 corresponding to negative, neutral and positive.

While machine-translated data is an option for expanding our training dataset, we are aware of the potential limitations in accuracy and context sensitivity.



As a result, we aim to continue to rely on Swedish datasets from more official sources like news or literature. If machine translation is used, it will be used sparingly, and the translated data will be thoroughly reviewed to ensure it fulfils our linguistic precision and contextual relevance criteria, but likely machine translation will not be used.

### **Second Phase: Developing a Lexicon-Based Model**

We begin by constructing a sentence-level sentiment analysis model that utilizes linguistic features. This initial step is crucial for establishing a baseline in sentiment analysis using a predefined lexicon and set of linguistic rules. More precisely this will be a rule-based model, not utilizing machine learning, as we want to explore a more advanced sentence-level rule-based approach, which is largely unexplored for Swedish, and we then want to compare it to a machine learning model to assess the strengths and weaknesses.

As part of our methodology, we aim to ensure accurate handling of key linguistic features such as negations and conjunctions like *and but* etc. Specifically, we plan to enhance our sentiment analysis model by:

- Identifying conjunctions within sentences to separate them into distinct parts. This allows for independent sentiment analysis of each segment.
- Analysing these segments individually to capture nuanced expressions of sentiment, particularly in complex sentences where both positive and negative sentiments may be present.

As our development advances, we will specify which more language elements will be examined, as well as the exact strategies for implementing them.

The input of the model will be a sentence and the output will be a number between -1 and 1 that will be approximated to the nearest label -1, 0, 1 corresponding to negative, neutral and positive. The unapproximated number will also be displayed to see the strength of alignment with the label.

### **Third Phase: Lexicon Expansion and Analysis**

The next stage involves expanding the lexicon SenSALDO by adding more labelled words. The lexicon SenSALDO currently contains 12 218 word entries labelled with sentiment, while the lexicon it is based on, SALDO, contains 131 020 word entries (not labelled with sentiment). Our aim is to expand SenSALDO with a minimum of 2 000 new entries and potentially more. This expansion is aimed at refining the model’s capability for accurate sentiment interpretation. We will assess the impact of this lexicon expansion by measuring changes in performance metrics after each iteration, providing insights into how the enrichment of the lexicon affects the model’s accuracy.

### **Fourth Phase: Using the KB-BERT Model**

Subsequently, we will be using KB Labs BERT (Bidirectional Encoder Representations from Transformers) model. This model represents an advanced approach, employing deep learning to understand contextual nuances in text.

### **Fifth Phase: Using the SVM Model**

in the fifth step we will apply an SVM model in conjunction with our rule-based model to create a Hybrid Model.

**Final Phase: Comprehensive Model Comparison** In this phase, we will undertake a thorough comparison of the three models:

1. Direct Comparison: We will test both the rule-based, KB-BERT model and the hybrid model on identical datasets. This approach aims to provide an objective basis for comparing their performance.
2. Evaluating Accuracy: The accuracy of both models will be evaluated by calculating the percentage of correctly classified sentiments (positive, negative, neutral) in a test set. This will be a key metric in assessing each model's effectiveness.
3. Analysing Precision, Recall, and F1-Score: We will also compute precision, recall, and the F1-score for a more nuanced understanding of each model's strengths and weaknesses, especially in the context of potentially imbalanced datasets.
4. Confusion matrix: The models will also be analysed through a confusion matrix, which will give further information about how the model performs on different classes and how many false positives and negatives it predicts.

Our initiative aims to construct and critically assess sentiment analysis models using this methodical, iterative process. We seek to provide thorough insights into sentiment analysis by contrasting more recent approaches like BERT with more conventional lexicon-based methods, therefore improving our comprehension of textual sentiment interpretation.

### 4.2.1 Negations

The handling of negations is implemented through a method that is called for each sentence in the dataset. Since it is not indicated in Sparv which words are negations, we assembled a list manually of such common negating words. This list is given to the function, which then checks the sentences for negation words and if it finds one it checks the hierarchical dependency relations connected to the negation word and negates the connected words. The list of negations that we created contains these words: *inte*, *ej*, *icke*, *ingen*, *inget*, *inga*, *knappast*, *aldrig*, *ingenstans*, *sällan*, *varken*, *nej*, *trots*.

As illustrated in Figure 4.1, sentences are analyzed using a hierarchical tree structure. In this structure, the negation word "*ingen*" serves as a child node to the word it negates. Consequently, the function ascends from the negation word to its parent node to apply negation to the associated word, "*vän*".

Similarly, consider the example shown in Figure 1.1. Here, the target of negation is the word "*glad*". However, the negation word "*inte*" is linked to the verb "*är*", with "*glad*" positioned as a child node of "*är*". To address this, our function extends negation to all child nodes of the word directly connected

to the negation word, ensuring comprehensive negation processing within the sentence structure.

One notable challenge was navigating through the linguistic construct of double negations. Such constructs often reverse the apparent sentiment of a statement, posing a considerable interpretative challenge, particularly for rule-based models. The precision required to discern the intended sentiment underlines the complexity of processing natural language, where double negations can subtly alter the meaning.

### 4.2.2 Expansion of the lexicon

The expansion of the SenSALDO lexicon was automated. In this process, each word is analyzed to identify its primary and secondary descriptors, which are broader semantic categories than the word itself. For example, the word might be decomposed into a primary descriptor that indicates its fundamental characteristic and a secondary descriptor that provides additional context or usage. This method helps to systematically categorize words within the lexicon based on their semantic relationships.



SALDO ▾ 1 TRÄFF (VISAR 1)						
BETYDELSE	LEMGRAM	ORDKLASS	PRIMÄR	SEKUNDÄR	BARN (PRIMÄRA)	BARN (SEKUNDÄRA)
 <a href="#">adventsstjärna</a>	<a href="#">adventsstjärna (substantiv)</a>	substantiv	<a href="#">prydnad</a>	<a href="#">advent</a>		
	 3559					

Figure 4.2: The entry in SALDO of the word "adventsstjärna" (advent star)

For example, consider the word "*adventsstjärna*", which is annotated in the SALDO lexicon. The primary descriptor for "*adventsstjärna*" is "*prydnad*" (decoration), and its secondary descriptor is "*advent*". This structure is depicted in Figure 4.2. The descriptor "*prydnad*" is further linked in a hierarchical chain: "*adventsstjärna* -> *prydnad* -> *pryda* -> *vacker* -> *bra*" (adventsstjärna -> decoration -> decorate -> beautiful -> good).

To expand the SenSALDO lexicon, we analyzed words from the SALDO lexicon, upon which SenSALDO is based but which lacks sentiment annotations. For each word not already in SenSALDO, we examined its primary and secondary descriptors. If a descriptor already exists in SenSALDO with an assigned sentiment, we apply the same sentiment annotation to the new word. For instance, since the descriptor "*prydnad*" carries a positive sentiment in SenSALDO, the word "*adventsstjärna*" is similarly labelled as positive, 1, when added to the lexicon, however, one could argue that this should in fact be a neutral word, and thus we can see that the automatic expansion could in some cases incorrectly label words.

In the process of expanding the SenSALDO lexicon, each word is initially checked for its primary and secondary descriptors in SALDO. Priority is given

to primary descriptors because they are more directly linked to the core meaning of the word. If a primary descriptor is identified in SenSALDO with an existing sentiment label, that label is applied to the new word. If the primary descriptor is not found, we then consider secondary descriptors. This recursive approach continues until a descriptor with a sentiment label is found or until the descriptor hierarchy is exhausted.

It is important to note that not all words in SALDO are associated with a primary descriptor that is also a word, as some are associated with the top node *PRIM*. In cases where the primary descriptor as a word is absent, the secondary descriptor would be considered instead. The special case where the secondary descriptor is "*inte*" is also always checked and then the resulting sentiment value is flipped.

Regarding the accuracy of applying labels through descriptors, this aspect is critical for validating the effectiveness of the lexicon expansion. We plan to systematically evaluate the accuracy of newly labelled words in a subsequent phase of this project. This evaluation will help determine the reliability of the descriptor-based sentiment assignment and ensure the utility of the expanded lexicon in practical applications.

### 4.3 The hybrid SVM model

The hybrid SVM model was created to explore a hybrid approach between a fully rule-based model and a fully machine-learning based model and it was created using State Vector Machines (SVM) while also combining it with input from the rule-based model, namely the one with both negation handling and the expanded lexicon. The motivation behind this hybrid approach is to attempt to enhance the predictive performance by integrating insights from the rule-based model's analysis with the flexibility and scalability of machine learning. The inputs provided to the SVM model were: the sentence, the individual word scores (from the rule-based model) and the rule-based model's prediction.

# 5

## Results

### 5.1 Rule-based model

The rule-based model can successfully, given a sentence annotated by Sparv, classify it as either positive (1), neutral (0) or negative (-1). The accuracy on the test dataset was found to be 0.429, and with negation handling it was 0.479. However, when using the expanded lexicon the accuracy actually dropped to 0.423 and 0.478 for without negation handling and with it respectively.

More specifically, the baseline model without negation handling nor the expanded lexicon achieves for three classes an accuracy of around 43% on the test dataset. However, looking more deeply at the results for each class reveals that it is much more proficient at classifying positive sentences with an accuracy of around 71%.

The model struggles to correctly classify negative sentences and looking at the precision versus recall values as well as the confusion matrix we can see that it can distinguish the neutral sentences from the negative ones but that it struggles to differentiate them from the positive. Additionally, we can see that, although not performing well on the negative sentences, when it actually predicts a sentence as negative it is quite likely that it is actually negative.

Test Accuracy: 0.4285714285714286  
Class 1 Accuracy: 0.7125748502994012  
Class 0 Accuracy: 0.4782608695652174  
Class -1 Accuracy: 0.09445277361319337

Precision (weighted): 0.5727757768773003  
Recall (weighted): 0.5203939583768358  
F1 Score (weighted): 0.5116691355436428

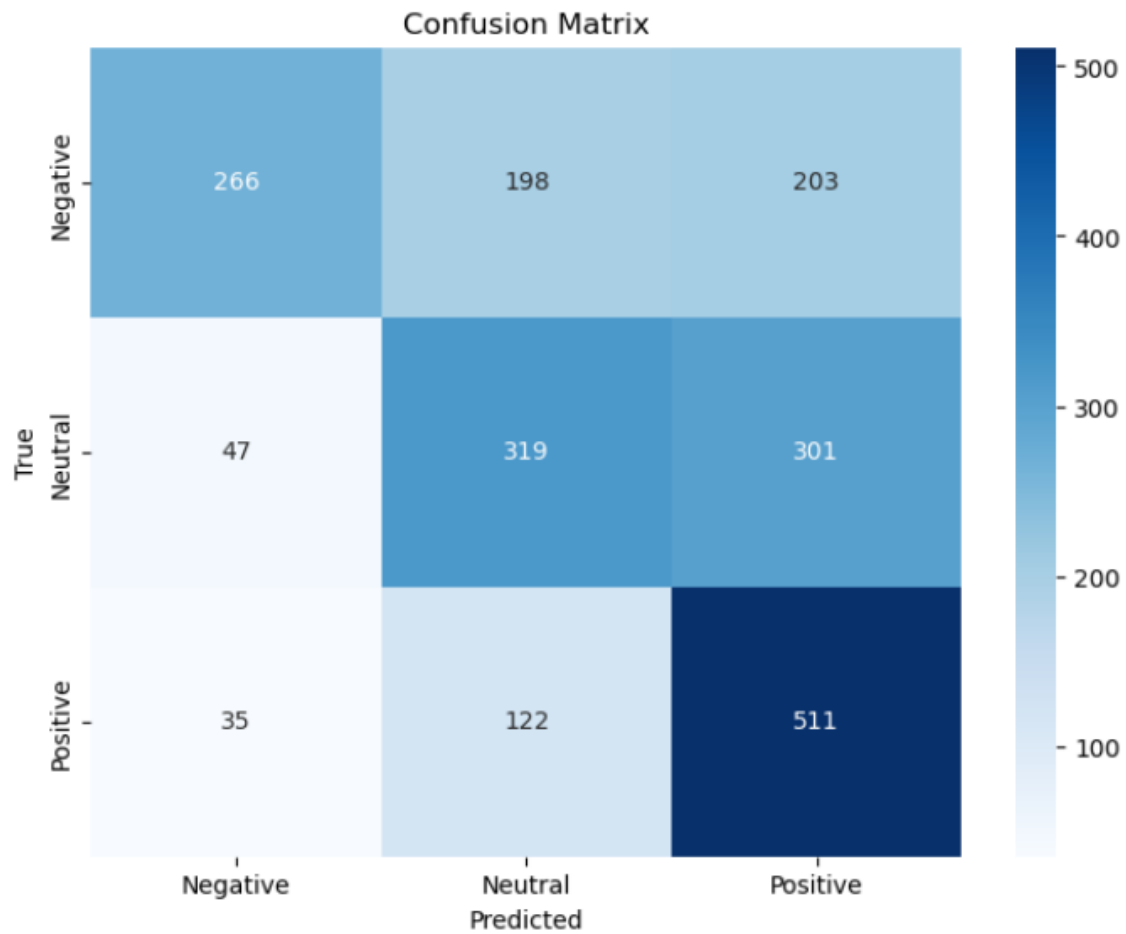


Figure 5.1: The confusion matrix of the baseline model

## 5.2 Negations

The inclusion of negations did increase the accuracy to 48% and thus providing an increase of around five percentage points. It is thus visible that the negation handling implemented had a positive effect on the model's performance. Since not all sentences contain negations the impact is thus limited by this fact and further testing can be done to assess the impact of negation handling on the accuracy of the model by only looking at the sentences containing negations.

Test Accuracy: 0.4785214785214785

Class 1 Accuracy: 0.7080838323353293

Class 0 Accuracy: 0.47976011994003

Class -1 Accuracy: 0.24737631184407793

Precision (weighted): 0.5782119166846458

Recall (weighted): 0.5395350455905895

F1 Score (weighted): 0.5338286942145618

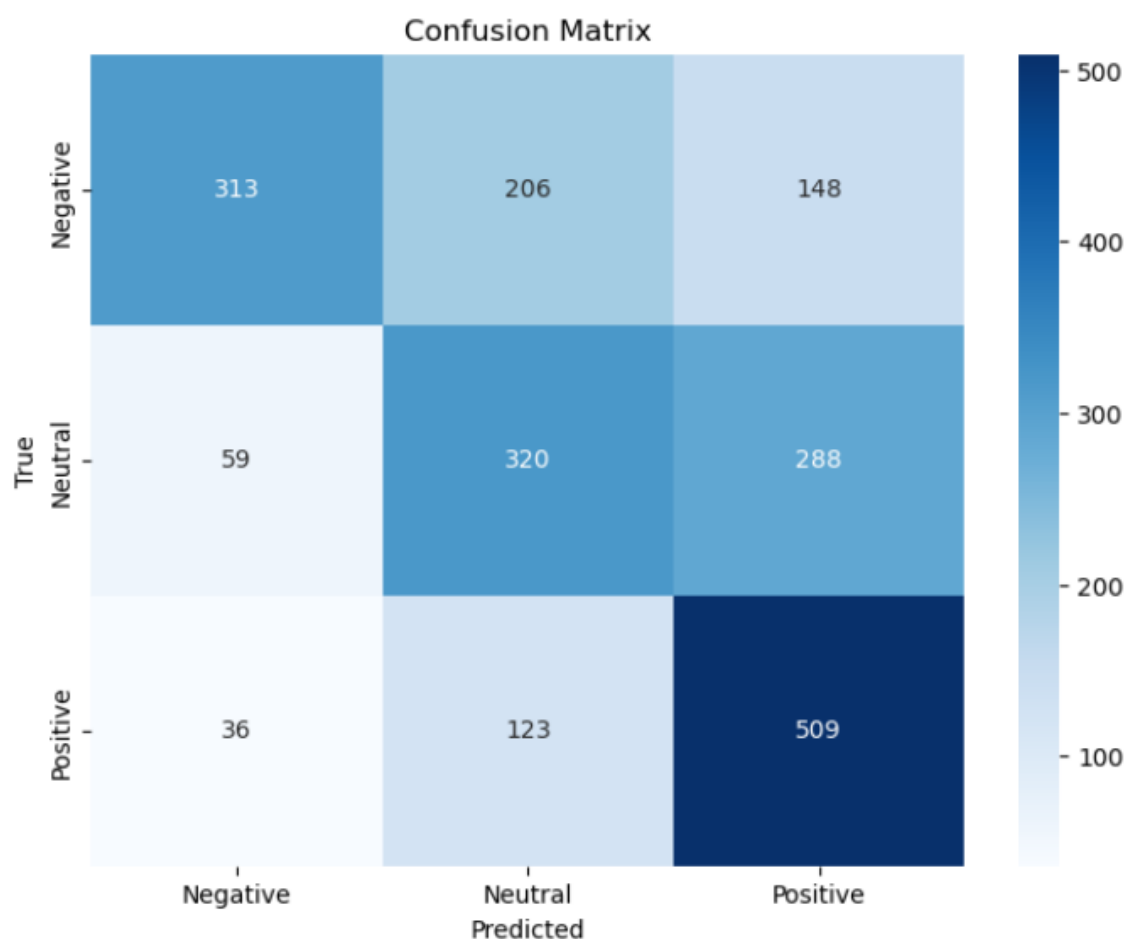


Figure 5.2: The confusion matrix of the model with negations handling added

### 5.3 The expansion of the lexicon and its contributions

The expansion of the lexicon was successful, increasing the number of entries from 12 287 to 115 422 which means that the lexicon was expanded with 103 135 entries. However, the correctness of the new entries' labelling is yet to be evaluated. This thus shows that the method was successful in expanding the lexicon, but the method of adding them automatically through using their descriptors will have to be explored further.

Furthermore, it is interesting that the model when using the expanded lexicon actually performs slightly worse, although it seems to have helped with classifying the negative sentences correctly, especially when also handling negations. This could be for a number of reasons. For example, it could be that some of the newly added words are possibly wrongly labelled during the automatic expansion and thus negatively impacting the performance.

Test Accuracy: 0.4225774225774226

Class 1 Accuracy: 0.7065868263473054  
 Class 0 Accuracy: 0.4302848575712144  
 Class -1 Accuracy: 0.13043478260869568

Precision (weighted): 0.5602453576821964  
 Recall (weighted): 0.5162525231433145  
 F1 Score (weighted): 0.5066240967766541

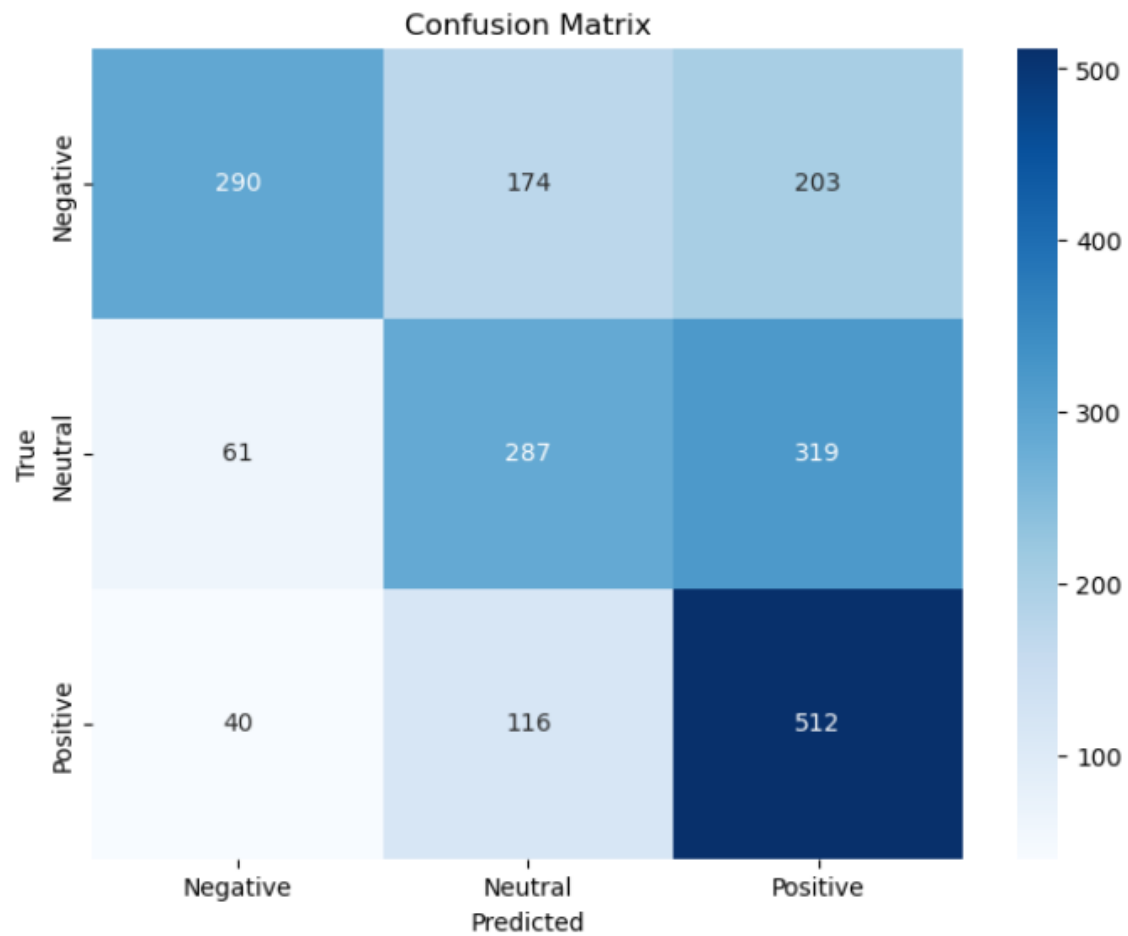


Figure 5.3: The confusion matrix of the model using the expanded lexicon

Test Accuracy: 0.478021978021978  
 Precision (weighted): 0.570767156571001  
 Recall (weighted): 0.540961926637433  
 F1 Score (weighted): 0.5340637404718841

Class 1 Accuracy: 0.7035928143712575  
 Class 0 Accuracy: 0.4347826086956522  
 Class -1 Accuracy: 0.29535232383808097



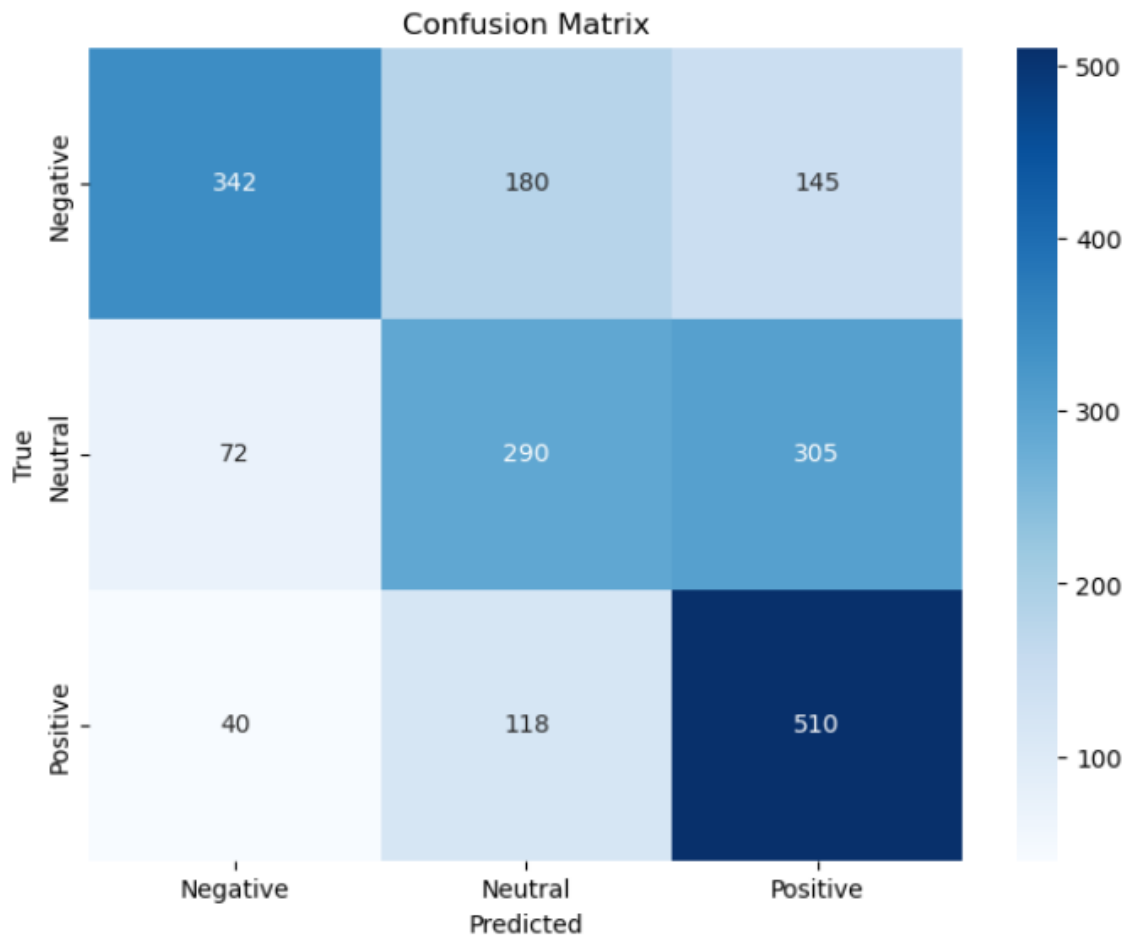


Figure 5.4: The confusion matrix of model using the expanded lexicon and negation handling

## 5.4 The KB-BERT model

The KB-BERT model was run on the same dataset and showed an accuracy of 0.796, which is similar to the reported accuracy of around 80% [26]. To assess the effectiveness of the model, a confusion matrix was employed, categorizing predictions into negative, neutral, and positive sentiments.

This confusion matrix revealed that the model is proficient in identifying neutral sentiments, with a significant majority of them correctly classified. However, there was a noticeable challenge in distinguishing negative sentiments, with a tendency to classify them as neutral. Positive sentiments were also more likely to be classified as neutral rather than positive, suggesting a conservative bias in the model's predictions.

The model's weighted precision was approximately 0.749, indicating that its predictions are correct about 74.9% of the time when considering the class imbalance. The weighted recall, at approximately 0.575, was lower, pointing to an area for improvement in detecting all instances of sentiment, particularly

for positive cases. The disparity between precision and recall is captured in the weighted F1 score of approximately 0.554, which suggests that while the model's predictions are relatively precise, they lack in consistency and completeness across all classes.

The confusion matrix and accompanying metrics underscore the potential for enhancement in the models classification ability. The model exhibits robust performance in some areas but also highlights the need for improved recognition of all sentiment instances, especially in reducing the number of false negatives for the positive class.

Test accuracy: 0.796

Accuracy for 1: 0.5209580838323353

Accuracy for 0: 0.9415292353823088

Accuracy for -1: 0.2623688155922039

Precision (weighted): 0.7488440271805842

Recall (weighted): 0.5749250749250749

F1 Score (weighted): 0.5543659567931186

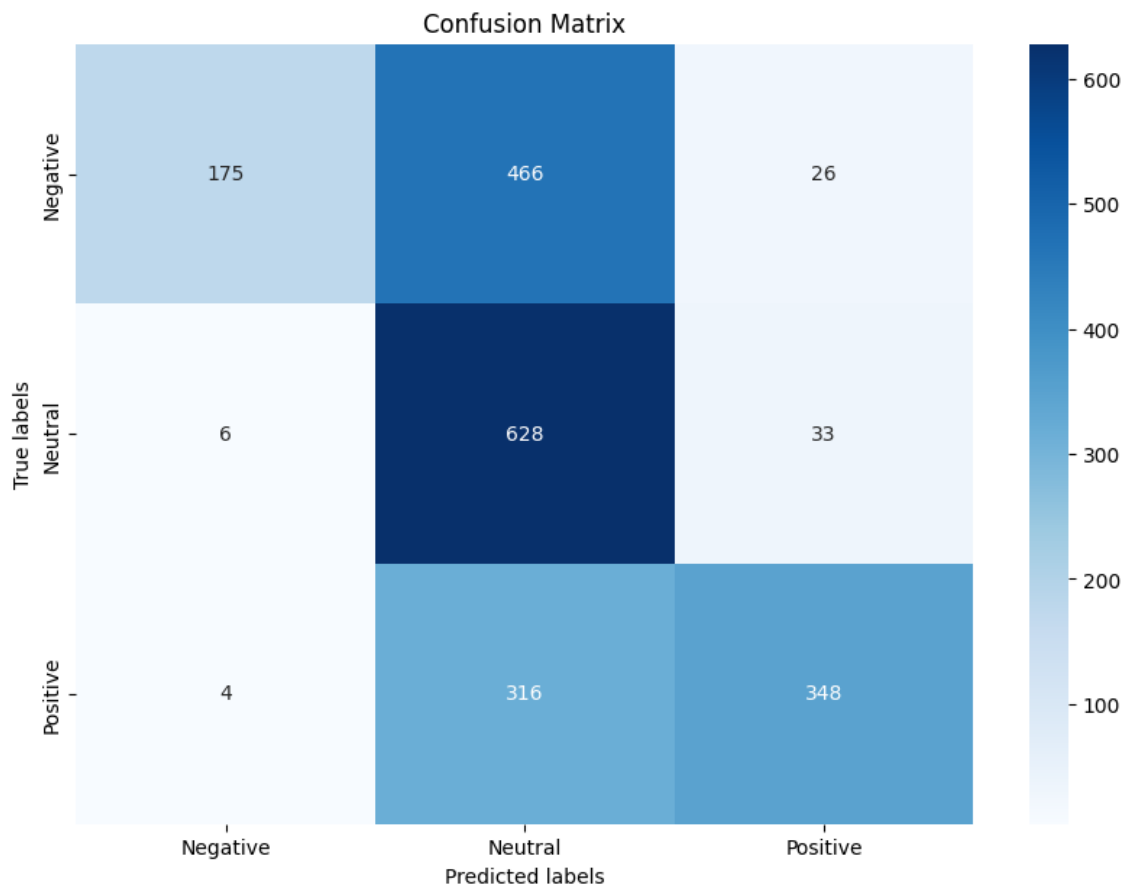


Figure 5.5: The confusion matrix of KB-BERT model

## 5.5 SVM-Hybrid Model

The precision, recall, and F1-score of the SVM-Hybrid model are all around 0.68, indicating balanced performance. This means that the model performs consistently across metrics and there are no major differences in precision and recall. For class 0, the model has intermediate precision and recall for neutral attitudes, meaning that, while it correctly detects neutral cases somewhat more than half of the time, it still makes a considerable number of mistakes. This implies that neutral sentiments should be distinguished more clearly from positive and negative ones.

The model does well in class 1 for positive attitudes, with high precision and recall. This demonstrates a relatively strong ability to appropriately recognize and categorize positive events. The strong recall indicates that most actual positive events are appropriately classified. Finally, for class -1, the model performs best for negative attitudes, with reasonably good precision and recall rates. This shows that the model excels at correctly recognizing negative cases and has fewer misclassifications in this category.

Test accuracy: 0.68

Accuracy for 1: 0.719

Accuracy for 0: 0.523

Accuracy for -1: 0.808

Precision (weighted): 0.685865

Recall (weighted): 0.683292

F1 Score (weighted): 0.684330

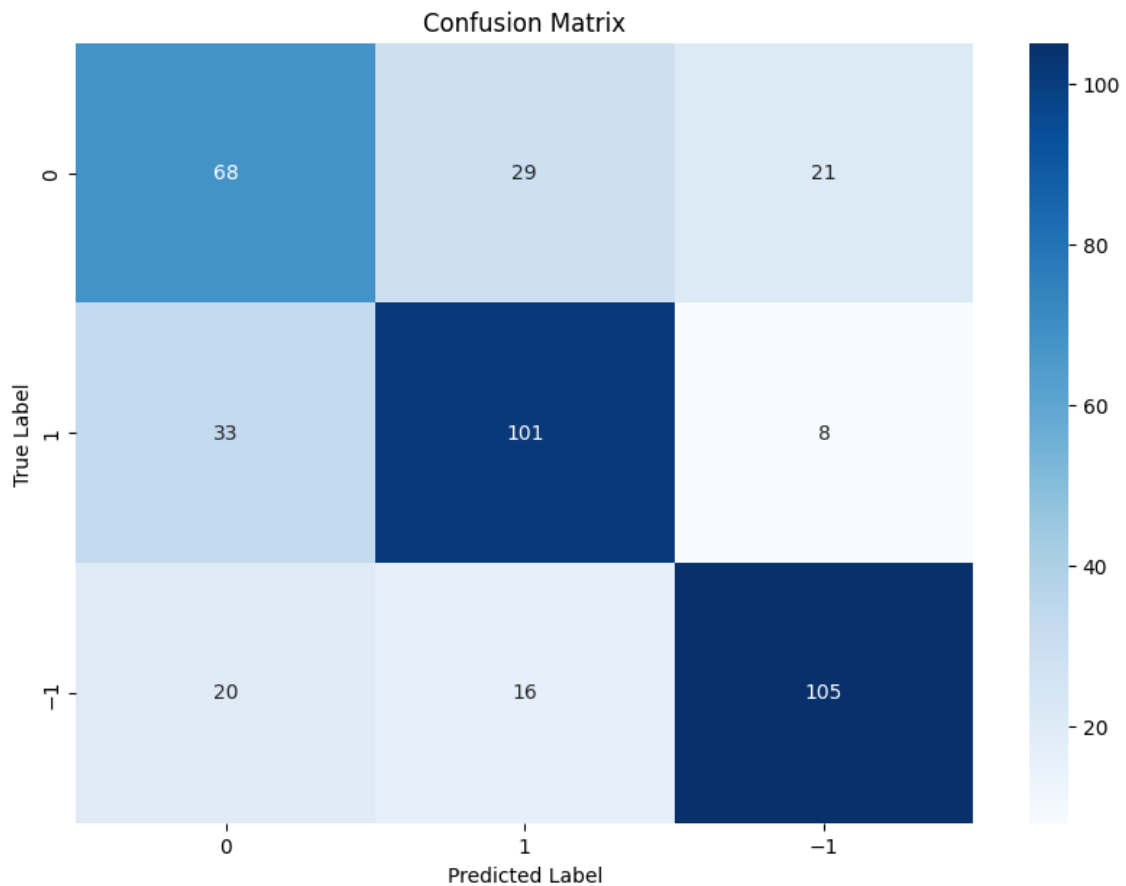


Figure 5.6: The confusion matrix of SVM-Hybrid model

## 5.6 Overview of Results

This section provides an overview of the performance metrics for the different models used in our study. The results are divided into two tables: one presenting the class-specific accuracy metrics and the other showing the weighted precision, recall, and F1 score metrics. These metrics are essential for understanding the effectiveness of each model in handling positive, neutral, and negative sentiments, as well as their overall performance in a weighted context.

### 5.6.1 Class-Specific Performance Metrics

The first table, 5.1 focuses on the accuracy of each model in classifying positive (1), neutral (0), and negative (-1) sentiments. These metrics provide insight into how well each model can distinguish between different sentiment classes.

<b>Model</b>	<b>Accuracy for 1</b>	<b>Accuracy for 0</b>	<b>Accuracy for -1</b>	<b>Test- Accuracy</b>
Lexical Model	0.713	0.478	0.094	0.429
Negations	0.708	0.480	0.247	0.479
Expansion of Lexicon	0.707	0.430	0.130	0.478
Negations and expansion	0.704	0.435	0.295	0.478
KB-BERT Model	0.521	0.942	0.262	0.796
SVM-Hybrid Model	0.719	0.523	0.808	0.683

Table 5.1: Class-specific performance metrics for different models

### 5.6.2 Weighted Performance Metrics

The second table, 5.2 presents the weighted precision, recall, and F1-score for each model. These metrics provide a more comprehensive view of each model’s performance by taking into account the balance between precision and recall, as well as the overall effectiveness in predicting the correct sentiments.

<b>Model</b>	<b>Precision (weighted)</b>	<b>Recall (weighted)</b>	<b>F1-Score (weighted)</b>
Lexical Model	0.572	0.520	0.511
Negations	0.578	0.539	0.533
Expansion of Lexicon	0.560	0.516	0.506
Negations and expansion	0.571	0.541	0.534
KB-BERT Model	0.748	0.574	0.554
SVM-Hybrid Model	0.685	0.683	0.684

Table 5.2: Weighted performance metrics for different models

# 6

## Discussion

In this chapter the results from our work and previous studies will be discussed.

### 6.1 Comparisons of the different approaches

In order to improve sentence-level sentiment analysis for the Swedish language, this study compares a lexicon-based model, the KB-BERT model, to the more modern SVM-Hybrid model. Our findings reveal substantial strengths and potential for improvement across all approaches.

The lexicon-based model, which uses classic rule-based methodologies and a vast lexicon, performs admirably in capturing positive emotion. However, its performance indicates a limited accuracy of 42.9%, which is moderately improved to 47.9% by integrating negation handling. The enlarged lexicon does not significantly improve accuracy, indicating potential faults in automatic sentiment label assignments throughout the lexicon’s augmentation process.

The KB-BERT model, which combines deep learning and contextual awareness, outperforms with 74.8% weighted precision, 57.5% recall, and 55.4% F1 score. Notably, the model does well at discriminating neutral sentiments. Despite this, the model’s conservative bias toward neutral classifications and difficulty identifying negative emotions indicate areas for development.

The SVM-Hybrid model, which combines rule-based and machine learning approaches, performs consistently across all criteria, with an overall accuracy of 68.33%. The precision, recall, and F1-score for the SVM-Hybrid model are roughly 0.686, 0.683, and 0.684, respectively. When examining the accuracy for each class, the model excels in identifying negative sentiments (80.8%), followed by positive sentiments (71.9%) and neutral attitudes (52.3%). This suggests that the hybrid technique outperforms the basic rule-based model and provides a viable alternative to the deep learning model.

## 6.2 Reasons for the model's performance

### 6.2.1 Difficulties with classifying negative sentences

From the results, we can see that our rule-based model is struggling with identifying negative sentences. After thorough investigation, we have encountered some of the issues that might be causing the issues with our model's poor classification. When using the sentence "*Jag glömde mitt lösenord och kunde inte logga in.*" ("*I forgot my password and could not log in*") the reason for the bad result is that every single word is either neutral or does not have a sentiment classification. So for sentences like this, one would need to use context to discern the actual sentiment of the sentence.

In regard to this, a BERT would perform better because it is a trained model and can to some degree figure out contextual nuances. Another reason for the poor result can be in how *sensaldo* is annotated, in the sentence "*Jag är allergisk mot nötter och måste vara försiktig med vad jag äter.*" ("*I am allergic to nuts and must be careful with what I eat.*") "*försiktig*" is annotated as positive in SenSALDO because of the meaning that can be being attentive to details. Though it can also have another meaning such as being careful to not get hurt. Given this problem of only having one entry, that being positive, it affects the classification of the sentence. Even if we did have more entries, it would be unclear if Sparv would pick the right entry given the situation.

### 6.2.2 Drop in accuracy when using the expanded lexicon

When the expanded lexicon is used, as mentioned, a drop in accuracy is noted. This means that some of the sentences that were previously correctly classified become incorrectly classified. Looking at and comparing the confusion matrices for the model with the expanded lexicon and the one without, but both with negation handling, we can see that the expanded lexicon slightly helps the model identify negative sentences but makes it worse at discerning neutral one, while positive sentences were largely unaffected.

When then looking at the results more closely, they seem to corroborate our analysis of the confusion matrices. Below are three examples of sentences that were correctly identified before expanding the lexicon but then became wrongly classified.

Firstly, the sentence "*Jag har planerat att städa mitt rum denna eftermiddag.*" is annotated as a neutral sentence but when using the expanded lexicon it gets classified as a positive sentence. This is because the word "*städa*" ("*to clean*") was not present in SenSALDO originally and when it was added to the expanded lexicon by automatically looking at the descriptors, it was added with the sentiment label 1 (positive) since its descriptor is "*ordning*" ("*order*") which is labelled as positive.

The main problem here is then when the new word was added through the descriptors, it didn't get the right label and thus negatively impacting the classification of the sentence as a whole. The same is the case with some other sentences like "Det är vanligt att ha ett paraply med sig på vintern." where the word "paraply" ("umbrella") is added as positive because of its descriptor "skydd" ("protection") which is annotated as positive.

Similarly, some neutral sentences become negative like "Jag såg en katt sitta på ett staket när jag gick hem." where "staket" ("fence") is added as negative because of its descriptor "hinder" ("obstacle") which is negative.

### 6.2.3 The hybrid approach

In this study, the SVM-Hybrid model was constructed using SVM in conjunction with a rule-based model that incorporated both negation handling and an expanded lexicon. The rule-based model, with its linguistic rules, provided a structured framework for sentiment analysis, while the SVM introduced a machine learning component capable of handling patterns in the data.

The performance metrics for the SVM-Hybrid model demonstrated a balanced and consistent output across various sentiment classes. The overall test accuracy of the model was 68.3%, indicating a robust performance in classifying sentiments. When broken down by class, the model exhibited an accuracy of 71.9% for positive sentiments, which is similar to what we got on our rule-based model, 52.3% for neutral sentiments, and 80.8% for negative sentiments. This suggests that the hybrid model is particularly effective at identifying negative sentiments, outperforming both the purely rule-based and the KB-Bert model in this category.

The SVM-Hybrid model achieved balanced precision, recall, and F1-scores, all hovering around 0.68. This indicates that the model maintains a consistent level of performance across different metrics, without major disparities between precision and recall. One of the most significant strengths of the hybrid approach is its high accuracy in identifying negative sentiments. With an accuracy of 80.8% for classifying negative sentiments, the model significantly reduces the number of wrongly classified negative sentences, a common issue encountered in the other models. The model's accuracy for neutral sentiments was 52.3%. While this is lower than the performance for positive and negative sentiments, it indicates a need for further refinement in distinguishing neutral sentiments from other classes. For positive sentiments, the model achieved a recall rate that indicates most positive events are correctly identified. This is crucial for applications where recognizing positive feedback accurately is important.

Compared to the lexicon-based and KB-BERT models, the SVM-Hybrid model provides a middle ground. It outperforms the lexicon-based model, which showed limited accuracy improvements even with negation handling and lexicon expansion. Although the KB-BERT model showed higher weighted preci-



sion and excelled in neutral sentiment classification, it struggled with negative sentiments, an area where the SVM-Hybrid model excelled.

One notable issue regarding this approach was the lack of training data. Given the time-frame of the study, we could only use the limited data available for training the model. This constraint likely affected the model's overall performance and generalization capabilities. Adequate and diverse training data are critical for training robust machine learning models, particularly in sentiment analysis, where linguistic nuances and contextual variations play a significant role. The limited dataset may have restricted the model's ability to learn and generalize from diverse linguistic patterns. Additionally, if the model is run multiple iterations of the same data, it could lead to potential overfitting on the available data and underperformance on unseen data. Expanding the training dataset with more annotated sentences from varied sources could significantly enhance the model's capability to accurately classify sentiments across different contexts and domains [9].

The integration of rule-based components with machine learning in the SVM-Hybrid model has proven to be a viable approach for enhancing sentiment analysis. The rule-based model provides clear, interpretable rules for sentiment classification, which are enhanced by the machine learning model's ability to learn from complex patterns in the data. This combination allows the hybrid model to maintain high accuracy in detecting negative sentiments and balanced performance across other sentiment classes.

#### **6.2.4 Other reasons and overview**

The model has problems distinguishing neutral and positive sentences. There are many possible reasons for this. Looking at the sentence "Den nya kaffebaren i grannskapet serverar utsökt espresso." this is a positive sentence however Sparv wrongly identifies the word "utsökt" ("delicious") as being from the verb "utsöka" which is annotated as negative. Thus here the problem is not in the lexicon itself, nor the rules of the rule-based model, rather it is a problem in Sparv itself that it sometimes incorrectly analyses sentences.

### **6.3 This thesis in context of previous studies**

This section contextualizes our study within the existing body of research on sentiment analysis for the Swedish language, incorporating both the rule-based model, the expanded lexicon approach, and the SVM-hybrid model. Previous studies have explored various methodologies, including lexicon-based approaches, simple machine learning models like SVM, and more advanced deep learning techniques such as BERT.

### 6.3.1 Lexicon-Based Models and Their Expansion

In previous work, lexicon-based models like SenSALDO have been utilized for sentiment analysis in Swedish, labeling words as negative, neutral, or positive. Studies such as those by Rouces et al. highlighted the potential and limitations of these models. Our study initially replicated the approach of using a predefined lexicon for sentiment analysis, achieving an accuracy of 42.9%, which improved to 47.9% with negation handling. However, expanding the lexicon did not significantly increase accuracy, most likely due to the automatic process not annotating all the new words correctly.

The inclusion of linguistic features for our rule-based model did improve its accuracy in the same range as was found by the study looking at the Korean language [18], however towards the lower end of the improvement range they mentioned, 2-28 percentage points. This then corroborates the inclusion of linguistic features in sentiment analysis as being something beneficial and causing an increase in accuracy. However, more features need to be studied as well as in combination with other approaches to fully assess their contribution.

### 6.3.2 Simple Machine Learning Approaches

An earlier study by Gustafsson [15] studied machine-learning-based algorithms including SVM, random forests, and multinomial logistic regression on Swedish tweets, finding SVM to be the most effective with an accuracy score of roughly 0.7. Our study builds on this method by combining SVM with a rule-based model to produce a hybrid system. This integration sought to combine the structured language rules of the rule-based approach with the pattern recognition capabilities of SVM, resulting in a more robust model. The SVM-Hybrid model attained an overall accuracy of 68.3%, exhibiting superior performance over the standalone rule-based model and agreeing with the accuracy seen in basic SVM models from earlier studies.

### 6.3.3 Advanced Deep Learning Models

The introduction of BERT models dramatically improved sentiment analysis capabilities. The KB-BERT model, developed by KBLab and made available through Hugging Face, achieved high accuracy rates of 0.8 for multiclass sentiment analysis and 0.88 for binary classification. However, these models necessitate significant computational resources and large datasets, which may be a constraint for less commonly spoken languages such as Swedish. Our study adds to this discussion by comparing the performance of the SVM-Hybrid model to KB-BERT, demonstrating that, while BERT models are more accurate, hybrid models can obtain competitive results with fewer resources.

# 7

## Conclusion

It is evident that the rule-based model can correctly classify sentences by sentiment and is especially good at classifying positive sentences but struggles with the classification of negative sentences as well as distinguishing neutral sentences from positive ones. Negation handling had a significant beneficial effect on the accuracy but the expanded lexicon, due to the automatic process not labelling all new entries correctly, made the accuracy go down very slightly. However, the combination of negation handling and the expanded lexicon significantly boosted the accuracy of the rule-based model on the negative sentences that the base model had a hard time classifying.

The combination of the rule-based model into a hybrid SVM model significantly increased the performance. However, both the rule-based and hybrid models performed worse than the KB-BERT model, although the hybrid model was still somewhat close, ten percentage points away. Nevertheless, it is interesting to see that the rule-based model and the hybrid model both were significantly better at classifying positive sentences than the KB-BERT model. The hybrid model was also significantly better than all other models, by around 50 percentage points, at classifying negative sentences. However, the KB-BERT model was the best by far at classifying neutral sentences with an accuracy of 0.942.

Furthermore, an issue presented is the difficulty of correctly annotating sentences by sentiment, even for humans and especially when also looking at neutral sentences. The dataset that was manually assembled, annotated and used for testing might thus contain a lot of difficult-to-annotate sentences where the actual sentiment is a bit unclear, as well as incorrectly labelled entries due to human error and lack of annotation experience. However, the dataset and the code are made public on GitHub [36] for other people to be able to further develop on this thesis.

However, looking at the performance of the KB-BERT model we can see that it still performed as expected on the test dataset and thus this can not be the only problem present, but it could simultaneously be the case that a BERT model has an easier time handling classification of sentences with unclear sentiment compared to a rule-based model.

## 7.1 Future work

Future work opportunities to expand this project include things like: evaluating more closely the correctness of the lexicon expansion method, as well as the handling of negations. Adding new rules and linguistic features is also something that could be done like boosters (jätte-, mycket, as-, himla, super-, etc.), especially in different approaches in order to get a better picture of their contribution. It would also be interesting if studies were made that would compare the inclusion of linguistic features in different languages to see how it might differ.

Furthermore, although we explored a hybrid model with SVM, exploring more types of hybrid models combining the syntactical breakdown of Sparv and the rules of the rule-based model with some machine-learning methods would be of interest to see how well these methods combine. Since the rule-based model also performed better on the negative sentences compared to KB-BERT and the KB-BERT model performed best on the neutral sentences, it would be interesting if they could be combined in a way that makes a hybrid model that is better overall.

However, since there was no sentiment dataset for Swedish with three classes: positive, negative and neutral, one would have to acquire more data. Although, we recently came across a dataset containing reviews for Swedish which was large enough for training and testing of a machine learning model, but it however only contains two classes: positive and negative. Thus, it would then only be possible to explore binary classification for the hybrid model, unless a three-class dataset is created for Swedish, but this could then be compared to the other models if one makes them run binary classification.

# Bibliography

- [1] L. Borin, M. Forsberg, M. Hammarstedt, D. Rosén, R. Schäfer, and A. Schumacher, “Sparv: Språkbankens corpus annotation pipeline infrastructure,” in *The Sixth Swedish Language Technology Conference (SLTC)*, Umeå University, 2016.
- [2] M. Hammarstedt, A. Schumacher, L. Borin, and M. Forsberg, *Sparv 5 user manual*. Göteborgs universitet, 2022.
- [3] L. Borin, M. Forsberg, and L. Lönngren, “Saldo: A touch of yin to word-nets yang,” *Language resources and evaluation*, vol. 47, pp. 1191–1211, 2013.
- [4] J. Rouces, N. Tahmasebi, L. Borin, and S. R. Eide, “Sensaldo: Creating a sentiment lexicon for Swedish,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), 2018, pp. 4192–4198.
- [5] J. Rouces, L. Borin, N. Tahmasebi, and S. R. Eide, “Defining a gold standard for a Swedish sentiment lexicon: Towards higher-yield text mining in the digital humanities,” in *DHN*, 2018, pp. 219–227.
- [6] J. Rouces, N. Tahmasebi, L. Borin, and S. R. Eide, “Generating a gold standard for a Swedish sentiment lexicon,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), 2018, pp. 2689–2694.
- [7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [8] V. Bonta, N. Kumaresh, and N. Janardhan, “A comprehensive study on lexicon based approaches for sentiment analysis,” *Asian Journal of Computer Science and Technology*, vol. 8, no. S2, pp. 1–6, 2019.
- [9] M. Malmsten, L. Börjeson, and C. Haffenden, *Playing with words at the national library of Sweden – making a Swedish BERT*, 2020. arXiv: 2007.01658 [cs.CL].
- [10] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature Machine Intelligence*, vol. 1, pp. 206–215, May 2019. DOI: 10.1038/s42256-019-0048-x.

- [11] S. Baccianella, A. Esuli, F. Sebastiani, *et al.*, “Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining,” in *LREC*, vol. 10, 2010, pp. 2200–2204.
- [12] G. A. Miller, “Wordnet: A lexical database for english,” *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995, ISSN: 0001-0782. DOI: 10.1145/219717.219748. [Online]. Available: <https://doi.org/10.1145/219717.219748>.
- [13] [Online]. Available: <https://wordnet.princeton.edu/>.
- [14] A. Agarwal, V. Sharma, G. Sikka, and R. Dhir, “Opinion mining of news headlines using sentiwordnet,” in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, 2016, pp. 1–5. DOI: 10.1109/CDAN.2016.7570949.
- [15] M. Gustafsson, *Sentiment analysis for tweets in Swedish : Using a sentiment lexicon with syntactic rules*, 2020.
- [16] N. Palm, *Sentiment classification of Swedish twitter data*, 2019.
- [17] A. Bello, S.-C. Ng, and M.-F. Leung, “A BERT framework to sentiment analysis of tweets,” *Sensors*, vol. 23, no. 1, p. 506, 2023.
- [18] H. Jang and H. Shin, “Effective use of linguistic features for sentiment analysis of korean,” in *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, Waseda University, 2011, pp. 173–182.
- [19] G. Patil, V. Galande, V. Kekani, and K. Dange, “Sentiment analysis using support vector machine,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 1, pp. 2607–2612, 2014.
- [20] A. Kotelnikova, D. Paschenko, K. Bochenina, and E. Kotelnikov, “Lexicon-based methods vs. bert for text sentiment analysis,” in *International Conference on Analysis of Images, Social Networks and Texts*, Springer, 2021, pp. 71–83.
- [21] H. Li, “Deep learning for natural language processing: Advantages and challenges,” *National Science Review*, vol. 5, no. 1, pp. 24–26, 2018.
- [22] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [23] V. Kosar, *Transformers self-attention mechanism simplified*, Mar. 2022. [Online]. Available: <https://vaclavkosar.com/ml/transformers-self-attention-mechanism-simplified>.
- [24] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. [Online]. Available: <https://aclanthology.org/D19-1410>.
- [25] J. Lee, W. Yoon, S. Kim, *et al.*, “Biobert: A pre-trained biomedical language representation model for biomedical text mining,” *Bioinform-*

- matics*, vol. 36, pp. 1234–1240, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:59291975>.
- [26] H. Hägglöf, *The kblab blog: A robust, multi-label sentiment classifier for swedish*, 2023. [Online]. Available: <https://kb-labb.github.io/posts/2023-06-16-a-robust-multi-label-sentiment-classifier-for-swedish/>.
- [27] S. Amarappa and S. Sathyanarayana, “Data classification using support vector machine (svm), a simplified approach,” *Int. J. Electron. Comput. Sci. Eng*, vol. 3, pp. 435–445, 2014.
- [28] R. Yacouby and D. Axman, “Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models,” in *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, S. Eger, Y. Gao, M. Peyrard, W. Zhao, and E. Hovy, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 79–91. DOI: 10.18653/v1/2020.eval4nlp-1.9. [Online]. Available: <https://aclanthology.org/2020.eval4nlp-1.9>.
- [29] L. Borin, M. Forsberg, and L. Lönnengren, “Saldo 1.0 (svenskt association-slexikon version 2),” *Språkbanken, Göteborg universitet*, 2008.
- [30] D. Dannélls, L. Borin, and K. Friberg Heppin, *The Swedish FrameNet++ Harmonization, integration, method development and practical language technology applications*. Amsterdam, Philadelphia: John Benjamins Publishing Company, 2021, ISBN: 9789027209900.
- [31] *Språkbanken*, Accessed: 2024-06-10. [Online]. Available: <https://spraakbanken.gu.se/>.
- [32] *Sentiment analysis and opinion mining*, May 2021. [Online]. Available: <https://www.gavagai.io/text-analytics/sentiment-analysis-opinion-mining/>.
- [33] S.-M. Kim and E. Hovy, “Identifying and analyzing judgment opinions,” in *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, R. C. Moore, J. Bilmes, J. Chu-Carroll, and M. Sanderson, Eds., New York City, USA: Association for Computational Linguistics, Jun. 2006, pp. 200–207. [Online]. Available: <https://aclanthology.org/N06-1026>.
- [34] M. A. C. G. v. d. V. Wouter van Atteveldt and M. Boukes, “The validity of sentiment analysis: Comparing manual annotation, crowd-coding, dictionary approaches, and machine learning algorithms,” *Communication Methods and Measures*, vol. 15, no. 2, pp. 121–140, 2021. DOI: 10.1080/19312458.2020.1869198. eprint: <https://doi.org/10.1080/19312458.2020.1869198>. [Online]. Available: <https://doi.org/10.1080/19312458.2020.1869198>.
- [35] B. Nusko, N. Tahmasebi, and O. Mogren, “Building a sentiment lexicon for Swedish,” in *Proceedings of the From Digitization to Knowledge workshop at DH*, 2016, pp. 32–37.
- [36] E. N. Ricardo Mansour, *Evaluating-lexicon-based-models-versus-bert-for-sentence-level-sentiment-analysis-in-swedish*, [://github.com/richi1998/Evaluating-](https://github.com/richi1998/Evaluating-)

Lexicon-Based-Models-versus-BERT-for-Sentence-Level-Sentiment-Analysis-in-Swedish/tree/main, 2024.