Feature Selection for Sentiment Analysis of Swedish News Article Titles
Feature Selection for Sentiment Analysis of Swedish News Article Titles

JONAS DAHL

Master in Computer Science
Date: July 6, 2018
Supervisor: Gabriel Skantze
Examiner: Olov Engwall
Swedish title: Val av datarepresentation för sentimentsanalys av svenska nyhetsrubriker
School of Electrical Engineering and Computer Science
Abstract

The aim of this study was to elaborate the possibilities of sentiment analyzing Swedish news article titles using machine learning approaches and find how the text is best represented in such conditions. Sentiment analysis has traditionally been conducted by part-of-speech tagging and counting word polarities, which performs well for large domains and in absence of large sets of training data. For narrower domains and previously labeled data, supervised learning can be used.

The work of this thesis tested the performance of a convolutional neural network and a support vector machine on different sets of data. The data sets were constructed to represent various language features. These included for example a simple unigram bag-of-words model storing word counts, a bigram bag-of-words model to include the ordering of words and an integer vector summary of the title.

The study concluded that each of the tested feature sets gave information about the sentiment to various extents. The neural network approach with all feature sets combined performed better than the two annotators of the study. Despite the limited size of the data set, overfitting did not seem to be a problem when using the features together.
**Sammanfattning**

Målet med detta arbete var att undersöka möjligheten till sentimentanalys av svenska nyhetsrubriker med hjälp av maskininlärning och förstå hur dessa rubriker bäst representeras. Sentimentanalys har traditionellt använt ordklassmärkning och räknande av ordpolariteter, som fungerar bra för stora domäner där avsaknaden av större uppmärkt träningsdata är stor. För mindre domäner och tidigare uppmärkt data kan övervakat lärande användas.

Inom ramen för detta arbete undersöktes ett artificiellt neuronnät med faltning och en stödvektormaskin på olika datamängder. Datamängderna valdes för att representera olika språkegenskaper. Detta inkluderade bland annat en enkel ordräkningsmodell, en bigramräkningsmodell och en heltalssummering av generella egenskaper för rubriken.

I studien dras slutsatsen att varje datarepresentation räckte för att tillföra information till klassificeraren. Det artificiella neuronnätet med alla datamängder tillsammans presterade bättre än de två personer som märkte upp data till denna studie. Trots en begränsad datamängd verkade inte modellerna övertränas.
# Contents

1 Introduction 1
   1.1 Sentiment analysis 1
   1.2 Objectives and motivation 2
   1.3 Problem statement 2
   1.4 Delimitations 3
   1.5 Outline 3

2 Background 4
   2.1 Sentiment analysis 4
      2.1.1 Data annotation agreement 5
   2.2 Traditional approaches 6
   2.3 Machine learning approaches 6
      2.3.1 Representing words and phrases 6
      2.3.2 Language features 8
      2.3.3 Algorithms 8
      2.3.4 Challenges 11
   2.4 Evaluation 11
      2.4.1 Class specific measures 12
      2.4.2 General measures 13
      2.4.3 Baseline classifiers 14

3 Method 16
   3.1 Collection of data 16
   3.2 Feature selection 18
      3.2.1 Feature types 18
      3.2.2 Combinations of features 19
   3.3 Algorithms 20
      3.3.1 Convolutional neural network 20
      3.3.2 Support vector machine 21
# CONTENTS

3.4 Implementation .................................. 21
   3.4.1 Stemmer ........................................ 22
3.5 Evaluation ....................................... 22
   3.5.1 Feature relevance ............................... 23

4 Results ............................................. 24
   4.1 Confusion matrices ............................... 24
      4.1.1 Feature set 1 ................................. 24
      4.1.2 Feature set 2 ................................. 25
      4.1.3 Feature set 3 ................................. 26
      4.1.4 Feature set 4 ................................. 27
      4.1.5 Feature set 5 ................................. 28
   4.2 $F_1$-scores and accuracy ....................... 29
   4.3 Feature impact .................................. 30
      4.3.1 Feature set 1 ................................. 30
      4.3.2 Feature set 2 ................................. 30
      4.3.3 Feature set 3 ................................. 31
      4.3.4 Feature set 4 ................................. 31
      4.3.5 Feature set 5 ................................. 32

5 Discussion and conclusions ......................... 33
   5.1 Main findings .................................. 33
   5.2 A linear approach ............................... 34
   5.3 Classifier characteristics ....................... 35
   5.4 Sources of error ................................ 35
   5.5 Comparison to other studies .................... 36
   5.6 Sustainability and ethical aspects ............ 37
   5.7 Future work .................................... 38
   5.8 Conclusions .................................... 39

Bibliography ........................................ 40
Chapter 1

Introduction

This chapter briefly introduces the area of sentiment analysis and its area of use. A problem background is presented in order to put the problem statement into a context. Subsequently, limitations and an outline of the report is given.

1.1 Sentiment analysis

Since the breakthrough of computers in everyday life, data has been central when giving users the best possible experience using them. Not only can data be a product to sell, but also a way of understanding the customer to improve public relations. Sales and marketing have always focused on reaching the customer with the right information at the right time to hopefully sell a product or service, now or in the future. This concept is still the basis for modern marketing. However, the number of ways of connecting with customers is continuously increasing, which makes it more difficult to break through the noise and get the attention of the customer.

Detecting feelings of written texts is a subarea within the language technology domain which can be solved using various approaches. Sentiment analysis is another name of this area. The most basic approaches include counting polarities of words and summarizing them to get a sentiment prediction \[1\]. However, advanced machine learning models might also be used. With plenty of previously classified pieces of text, new texts can be assessed using models created from the said data.
Sentiment analysis can be used in many different applications. It has been used for detecting conversion temperatures on Twitter \[2\] and identifying positive and negative reviews \[22\], in addition to a large number of other applications. However, the methods change with the data set used.

1.2 Objectives and motivation

This study elaborates the possibilities of sentimentally analyzing news article titles using an artificial neural network and a support vector machine. The objective was to visualize the differences in result between the different feature sets and test which features increase the accuracy the most, with respect to the problem statement. The result was evaluated by comparing both accuracy and weighted measures, since the data was not evenly distributed.

A news title consists of a vast number of features. These features are for example the words, their order, punctuation, quotes, number of words and number of words in specific word classes. This thesis is an attempt to visualize if any feature contributes more to the classifier than others.

Sentiment analyzed news article titles can act as information used when presenting for example advertisements and improving the user experience when reading the article. Some advertisements may perform better for articles of certain sentiments, and increase both leads and sales. The motivation of this thesis is to enable more information to be taken into account when trimming the user experience of reading news articles.

1.3 Problem statement

The study was carried out with the following problem statement as foundation:

*Which features of Swedish news article titles have the most impact on the sentiment analysis, using a neural network approach and a support vector machine approach and how well can the sentiments of the titles be predicted?*
1.4 Delimitations

The study focuses on Swedish news titles, which may differ in both grammar and semantics from news titles in other languages. Only articles from the Swedish daily newspaper Aftonbladet was analyzed. Only the support vector machine and the neural network approach was analyzed to narrow the problem considerably. Since the goal was to identify well performing feature sets, and different algorithms behave differently on different sizes of inputs, it is hard to generalize the results to other algorithms. The work does not either directly expand to larger domains of words or texts.

The content of the news article belonging to the title will not be taken into account. The title is the sole item to be analyzed, without any meta data.

Sentiments include all kinds of feelings, but is in this report limited to the following three. One news title may only have one sentiment. The correct sentiment label is decided by manually labelling them with one of the below listed labels:

- positive
- neutral
- negative

1.5 Outline

The report is divided into six chapters of which the introduction is the first one. The background chapter presents knowledge necessary to interpret the results of the experiment. Previous research within the sentiment analysis area and machine learning approaches to it are presented. Subsequently, the methodology describes how the experiments were carried out, including presentation of both algorithms and data extraction.

In the fourth chapter, the results are presented, followed by a discussion where interpretations of the results are made. The discussion is carried out with implementation, result, sustainability and ethics in mind. A summary of the conclusions that can be drawn from the study can be found at the end of the discussion chapter. Following the discussion, the bibliography can be found.
Chapter 2

Background

This chapter acts as an introduction to the technical and detailed aspects required to fully understand the mechanisms and results of the study. Different state-of-the-art approaches to sentiment analysis are described, and selected machine learning algorithms are elaborated on.

2.1 Sentiment analysis

Subjective texts can be classified into sentiments. Sentiments include all kinds of feelings, but is in this report limited to the distinct labels: positive, neutral and negative. These labels lay the foundation for the sentiment analysis area, which expands to automatically labeling texts with these labels based on knowledge and own values. Sentiment analysis has been used in social media to determine discussion spirit, for example on Twitter [2]. However, there are language differences between social media texts and news article titles. Communication in social media is often written informally and as longer texts whereas news article titles are shorter and more formal.

The classification can be done on different levels of texts, spanning from single words and short phrases to complete documents. For document level classification, unsupervised machine learning algorithms have proven useful [22]. Supervised ones have performed well when focused on only the subjective parts of the texts [19].

For sentence level classification, non-machine-learning algorithms have been constructed. By part-of-speech tagging sentences, each word can be assigned a polarity from an expanded WordNet [17] initiated
with a few core polarized words [9]. The polarities of the words are then combined in order to produce a sentence polarity.

Phrase and word level classification is often conducted using pre-compiled lists of words assigned with a polarity [1]. The polarities are then adjusted to fit the context, regarding for example negations and expletives.

### 2.1.1 Data annotation agreement

Previous research also shows the problems of sentiment labeling. Supervised approaches to sentiment analysis require pre-annotated and classified data by its nature. However, two human beings may classify the same sentence differently.

Data classification agreement can be measured using Cohen’s kappa value, $\kappa$. The definition of $\kappa$ is presented in Equation (2.1), where $p_o$ is the observed agreement rate and $p_e$ is the theoretically expected agreement if the classification is selected randomly weighted according to the distribution, calculated as in Equation (2.2). The number of data points is denoted $N$, the available classes $K$ and the number of times annotator $i$ used the label $k$ is denoted $n_{ki}$.

A $\kappa$ of 1 implies full agreement and 0 an agreement that can be considered equally good as randomization.

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (2.1)$$

$$p_e = \frac{1}{N^2} \sum_{k \in K} n_{k1}n_{k2} \quad (2.2)$$

Regarding human classification, 76.19% of adjectives and 62.35% of verbs where classified into the exact same classes by two different human beings where the classes positive, negative and neutral were used by Hovy et al. [9]. Accuracy grew to 88.96% and 85.06% correspondingly when neutral and positive was seen as the same class. The study analyzed 100 sentences with a $\kappa$ of 0.91 between two human annotators. A similar study, but with expressions instead of single words, resulted in an agreement between two persons by 82% [24], and a $\kappa$ of 0.72.
2.2 Traditional approaches

Numerous attempts have been made to computationally solve the task of classifying text by sentiments. The traditional approaches not using machine learning algorithms, instead use different kinds of pre-compiled word polarity lists to grade words, phrases, sentences and longer texts.

Agarwal et al. [1] used data from The Dictionary of Affect in Language [23] and WordNet [17] to sentiment analyze phrases. Core words were selected for each sentiment, which were expanded by the WordNet graph to give polarities to neighboring words. As the distance from the initial words increased, the polarity was adjusted towards neutrality.

The traditional approach is useful when the amount of available labeled data is small, since no training is done on previous knowledge. The absence of labeled data however aggravates validation and evaluation.

2.3 Machine learning approaches

Sentiment analyzing can be done using machine learning algorithms as well. Depending on the objective and the pre-requisites, both unsupervised and supervised algorithms can be used. If no labeled data is present, the unsupervised algorithms can cluster the data for later evaluation.

2.3.1 Representing words and phrases

Data representation is central when training machine learning models to fit the data they represent [25]. For language models, this is especially essential since the domain of the input is infinite. This section presents different representations of text that has been proposed in previous research.

Bag-of-words

A bag-of-words is a vector representation of a text which holds a count of every occurring word, in the unigram case. An $n$-gram bag-of-words counts every occurrence of $n$ following words. Following from
that, in the unigram case, this representation only stores a count and
the order of the words is forgotten. Two examples of bag-of-words
representations are shown in Table 2.1.

In the \( n \)-gram case, the representation vectors will be even more
sparse than in the unigram case. Pak et al. [18] used bag or words
as representation of Twitter messages handling negation of words by
treating them separately from non-negated opposites. A variation of
the bag-of-words is to store only the binary occurrence of a word, and
not a count. This performed well in a study by Pang et al. [20].

\textbf{Table 2.1:} bag-of-words vectors for two phrases: (1) "the quick brown fox"
and (2) "the slow yellow fox eats the carrot".

<table>
<thead>
<tr>
<th></th>
<th>the</th>
<th>quick</th>
<th>brown</th>
<th>fox</th>
<th>slow</th>
<th>yellow</th>
<th>eats</th>
<th>carrot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phrase 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Phrase 2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\textbf{Word vectors}

A word vector is a representation of a word as a high-dimensional
vector. The values of the vector are learnt, often with unsupervised
learning, by processing large corpora. Chen et al. [16] constructed the
tool Word2Vec which uses neural networks to train vectors of words.
The final vectors can then be used for semantic analysis since the re-
lations between representations of words are based on context. An
example is that the vector representation of "Paris" minus the vec-
tor representation of "France", plus the vector representation of "Italy"
would result in a vector which closest neighbor is the vector represent-
ing "Rome".

Word vectors are built from the context near the words in the cor-
pora used for training. Since they only represent singular words, a
combination of the word vectors in a sentence must be derived in or-
der to classify full phrases. This can be done by for example using
paragraph vectors [12].

Daly et al. [14] finds that word vector representations might per-
form a little better than other representations for sentiment analysis on
documents, if used together with a bag-of-words.
Paragraph vectors

Le et al. [12] proposed in 2014 a representation of word sequences called paragraph vectors. They represent paragraphs and shorter documents as fixed-length vectors, by adding word vectors together. The error rate of the paragraph vector in a sentiment analysis case of 100,000 movie reviews is observed to be 7.42% in comparison to the bag-of-words which is approximated to 12.20%.

2.3.2 Language features

Typical approaches to sentiment analysis involves lists of words with their a priori polarity. When scoping out to sentence level analysis these a priori polarities are combined by different rules. Hoffman et al. [24] presents examples of this. For example negations must be taken into account, since they change the overall polarity of the phrase. Negations can be short distance or long distance, changing the following word or words further away.

Hoffman et al. [24] also studied the impact of word features and modifiers. Examples of such modifiers are word classes of surrounding words and magnitude of the word itself. Sentence features included connections to nearby sentences and part-of-speech tags of the contained words. The inclusion of the additional features improved the classification by 4.2%.

2.3.3 Algorithms

There are numerous machine learning algorithms. With background from previous research within the language technology area, a few are selected and presented in this section.

Naive Bayes classifier

Bayes' theorem with strong and naive independency assumptions between features acts as the base for the Naive Bayes classifier family [21]. By examining the general probability and combine it with the class-specific probability a final value is calculated which is used for ranking the possible turnouts.

However, the Naive Bayes classifier has been outperformed by for example the support vector machine in sentiment analysis contexts [5],
which refrains from using it further since there are other better performing alternatives.

**Support vector machine**

The support vector machine is a binary classifier that approximates a hyperplane which maximizes the distance to both classes in an often high-dimensional space [7]. Bhayani et al. [5] found that their support vector machine outperformed both the Maximum Entropy and the Naive Bayes classifier when sentiment analyzing Twitter data.

Furthermore, a support vector machine is a binary classifier and can only differ between two classes. Since this thesis considers a three-class problem, at least two support vector machines have to be used. The traditional way of taking care of this extension is building one classifier for each class [11]. The classifier with the highest probability of its positive label is selected as the label for the data point.

Figure 2.1 shows how two support vectors divide the data points of the training set. The training, which consists of finding the vector, can be performed using various approaches.

*Figure 2.1: A conceptual image of the support vector machine. The distance between the support vectors is maximized. Two of the data points are considered to be noise.*

**Artificial neural network**

Deep learning has proven to be useful in a wide area of situations, from image classification to text information retrieval [13]. With increasing computational power, multiple artificial neurons can be connected to create nets. The nets, called artificial neural networks, are inspired by nature and the human brain.
Artificial neural networks consist of neurons connected together. The outputs of one neuron acts as one of possibly many inputs to other neurons. Every neuron has a function, generally a weighted sum of the input, producing output for given input. These weights are trimmed during the training stage. The neurons are often divided into layers, where the outputs of neurons in one layer only connects to inputs of neurons in the next layer. No intralayer connections are allowed in the layer case. One layer is used for input modelling and one for output modelling. In between those, an optional number of hidden layers, but at least one, is present [4]. Figure 2.2 shows this graphically.

![Figure 2.2](image)

**Figure 2.2:** A conceptual image of an artificial neural network with two hidden layers with four neurons in each. The left and yellow layer is the input layer, the red layers in the middle are the hidden layers and the blue is the output layer. The number of neurons in each layer is variable.

The network is trained by feeding training data into the input neurons and comparing the output with the actual classification. The errors are back-propagated and the weights are changed to better fit the desired output. This training is repeated a fixed number of times [13].

There are multiple variations of the neural network specialized on different areas. For example Kalchbrenner et al. [8] presents a convolutional network designed to improve analysis of sentences. A convolutional network interprets sequences that are filtered by an \( n \)-dimensional weight vector for each \( n \)-gram in the sequence. The resulting sequence is then used for learning. At least one layer has to be convolutional for the network to be called convolutional [8]. The convolution reduces
the dimension of the input and takes the order of the input into account.

2.3.4 Challenges

When applying machine learning algorithms a lot of challenges must be solved. The most important and relevant ones for this thesis are elaborated in this section.

Overfitting and the Curse of Dimensionality

When training models the possibility of overfitting must be taken into account. Overfitting occurs when the model is too well fit to the training data but generalizes badly to unseen data [6]. In the neural network case, the number of parameters to tune is big. The feature arrays are often sparse. As an example, the bag-of-words case can be observed. The count of unique words in all corpora is often many multiples of the count in a sentence, which leads to sparse vectors. Neural networks also have meta parameters, like initial weights and learning rate, that can be seen as more dimensions. In the neural network case, the number of neurons can actually act as a dimension limiting factor depending on the input features.

  Regarding the support vector machines, a too high polynomial of the kernel function may make the classifier overfit.

  Overfitting can be observed by validating the model with unseen data. If the model is overfit, new data will probably not be classified correctly. This will show in accuracy, precision and recall measures [6].

Uniqueness

When training language models, the exact sequence of words in a sentence is unlikely to exist even partly in the training data. The models must therefore adapt well to similar, but not identical data [16]. The use of too specific features may make the model unusable.

2.4 Evaluation

There are multiple evaluation measures in machine learning and prediction contexts. Both whole models and single classes can be evaluated using different measures. This section presents both kinds.
2.4.1 Class specific measures

There are plenty of properties that identify a good prediction model. When examining a specific class, four different turn-outs are possible. True positives (TP) are the data points correctly classified in the examined class and true negatives (TN) are the ones correctly classified in the non-examined classes. False positives (FP) are the examples wrongly classified as members of the examined class and false negatives (FN) are ones wrongly classified as members of other classes.

Accuracy

The accuracy of the class is defined in Equation (2.3). It measures the number of correctly classified samples in relation to the total amount. However, the accuracy has drawbacks when comparing classes and models. If the classes differ in size, and the current class is smaller than the others, the accuracy will seem lower since the numbers of true positives and true negatives are smaller.

\[
\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}
\]

Precision

The precision of the model on a class is defined in Equation (2.4). Since the precision measures how many of the positives that are correctly classified, a high precision can be reached by classifying everything as something other than the examined class. That would however make the recall low.

\[
\text{precision} = \frac{TP}{TP + FP} \tag{2.4}
\]

Recall

The recall of the model on a class is defined as in Equation (2.5). It corresponds to the actual members of the examined class that was classified correctly. A high recall can be achieved by always guessing on the examined class, but that would decrease the precision. Other names of the recall is sensitivity or true positive rate.
\[ \text{recall} = \frac{TP}{TP + FN} \quad (2.5) \]

**F\textsubscript{1}-score**

A combination of high recall and high precision is sought in order to guarantee the accuracy. The \( F\textsubscript{1} \)-score of the model on a class defined in Equation 2.6 combines both recall and precision into a value useful for comparison.

\[ F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (2.6) \]

Since the \( F_1 \)-score takes both precision and recall into consideration, tactical guesses on the same class will be punished, as well as bias against certain classes.

### 2.4.2 General measures

General measures does not require a specific class to be examined, but examines the whole model itself. This is useful since this study compares models and not classes.

**Micro averaged \( F\textsubscript{1} \)-score**

The micro averaged \( F\textsubscript{1} \)-score is defined as an average of the \( F\textsubscript{1} \)-scores of the classes of the model. The average is calculated with respect to each data point, with compensation for different amount of documents in each class.

**Macro averaged \( F\textsubscript{1} \)-score**

The macro averaged \( F\textsubscript{1} \)-score is another average of the \( F\textsubscript{1} \)-scores of the classes of the model. However, unlike the micro average, the macro average has equal weights for the classes of the model. This equals the arithmetic average of the \( F\textsubscript{1} \)-score of the classes.

If the distribution of data points is equal for each class, the macro and micro averages of the \( F\textsubscript{1} \) scores are equal.
Confusion matrix

The confusion matrix presents each classified sample showing both the actual class and the predicted class. It visualizes the classifier’s predictions. A model with 100% accuracy will have a diagonal matrix as confusion matrix with the diagonal’s values summing up to the amount of samples in the validation set.

Table 2.2: Example of a confusion matrix.

<table>
<thead>
<tr>
<th>Actual value</th>
<th>Predicted class</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>74.1%</td>
<td>21.1%</td>
<td>4.8%</td>
<td></td>
</tr>
<tr>
<td>Class 2</td>
<td>15.1%</td>
<td>72.3%</td>
<td>12.6%</td>
<td></td>
</tr>
<tr>
<td>Class 3</td>
<td>9.7%</td>
<td>25.1%</td>
<td>65.2%</td>
<td></td>
</tr>
</tbody>
</table>

2.4.3 Baseline classifiers

To be able to compare the generated classifiers, baseline models are defined and presented in this section. The baselines implies limits for certain quality measures.

Randomization baseline

Randomly selecting a class when prediction results in an average accuracy of \( \frac{1}{n} \), for \( n \) possible classes. This is often considered the lowest possible baselines since it is easy to implement without any logic, and never performs better than the majority class baseline in the average case.

Majority class baseline

The majority class baseline predictor guesses the same class every time. The prediction is educated enough to select the most represented class in the population, which leads to an accuracy better than or equal to randomization. Since the same class is selected each time, the accuracy always equals the percent of items in the largest class of the total amount of items.
Human baseline

Studies show that humans do not agree on all sentiment classifications. 76.19% of adjectives and 62.35% of verbs were classified into the exact same classes by two different human beings where the classes positive, negative and neutral were used [9]. An accuracy of around 70% is therefore acceptable for single word classification. A human baseline can also be calculated by validating the data set with multiple human annotators. The value of $\kappa$ then indicates how well a human being can predict the classes without any previous knowledge. For the definition of $\kappa$, see section 2.1.1.
Chapter 3

Method

This chapter introduces the methodology of the project and focuses on data processing and analysis of the processed data. The selected actions are motivated. Subsequently, the methods for evaluating the results are presented and substantiated.

3.1 Collection of data

Machine learning algorithms rely on both data quality and quantity. In order to generalize well, the data must be both representative for the whole input spectrum and as correct as possible. Therefore, new data was created by manual classification.

News titles from the Swedish daily newspaper Aftonbladet was gathered from 8,500 articles published from 2018-02-04 to 2018-03-14. All article titles were downloaded from an internal Aftonbladet Content API without any filtering or further selection. The titles were then classified as positive, neutral or negative by the sole author of this report. They were judged without any other context than the assessor’s own knowledge and mind. In case of titles involving more than one category of sentiment and more than one subject, the most polar expression preceded. Examples of titles classified into each class can be found in Table 3.1.

The classification of sentiments was decided from the consequences for the main, or first occurring, subject. For example, the title "Team A won against Team B" is positive despite the negative implications for Team B. The title "Team B lost against Team A", which conveys the same piece of news, is negative, since the focus is on the loss of Team B rather
than the win of Team A. If the main subject could not be determined, the sentiment was set to neutral.

Table 3.1: Examples of title kinds classified into each class.

<table>
<thead>
<tr>
<th>Title category</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective statements about negative things</td>
<td>Negative</td>
</tr>
<tr>
<td>Objective statements about positive things</td>
<td>Positive</td>
</tr>
<tr>
<td>Objective statements about neutral things</td>
<td>Neutral</td>
</tr>
<tr>
<td>Guides and instructions</td>
<td>Neutral</td>
</tr>
<tr>
<td>Titles not understandable without context</td>
<td>Neutral</td>
</tr>
<tr>
<td>Quotes with criticism</td>
<td>Negative</td>
</tr>
</tbody>
</table>

A quantitative summary of the resulting data sets can be found in Table 3.2. There were in total 8,500 news titles, that were not distributed evenly over the classes. There were more negative articles than positive, and more positive than neutral ones.

Table 3.2: Number of titles classified in each class by annotator 1.

<table>
<thead>
<tr>
<th>Class</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive titles</td>
<td>2,578</td>
</tr>
<tr>
<td>Neutral titles</td>
<td>2,099</td>
</tr>
<tr>
<td>Negative titles</td>
<td>3,823</td>
</tr>
<tr>
<td>Total</td>
<td>8,500</td>
</tr>
</tbody>
</table>

To set a standard for the human baseline, 478 of the 8,500 classified titles were also classified by another annotator. The titles annotated twice were randomly selected without any stratification. The annotators did not discuss the titles and did only have the criteria stated earlier in this section.

The $\kappa$ value was calculated to be 0.669, using 373 of 478 agreeing annotations and 160.8 agreements expected by chance. The agreement rate between the annotators was 78.0%. Table 3.3 shows the confusion matrix of the 478 cross-annotated titles.
Table 3.3: Titles classified into each class by the two different annotators.

<table>
<thead>
<tr>
<th></th>
<th>Annotator 2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Positive</td>
<td>Neutral</td>
<td>Negative</td>
</tr>
<tr>
<td>Annotator 1</td>
<td>124</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>Neutral</td>
<td>24</td>
<td>97</td>
<td>37</td>
</tr>
<tr>
<td>Negative</td>
<td>2</td>
<td>23</td>
<td>152</td>
</tr>
</tbody>
</table>

3.2 Feature selection

In this section the choice of input features for the machine learning algorithms and their motivation of use is presented.

3.2.1 Feature types

A list of possible features and their variations was produced by analyzing available literature. The following features were selected for testing, since they proved useful in previous research presented in chapter 2.

The different types of features used were unigram bag-of-words, bigram bag-of-words, special characters counts, and other textual properties. The motivation of using these is presented in this section.

Unigram bag-of-words

The bag-of-words feature is a word counter since it is implemented in the unigram case. Each feature in the input vector represents the count of a certain word in the input sentence. Stemming and lower casing was used to decrease the number of unique words, and join different inflections of the same primitive together. However, the information loss of removing the end of words must be taken into consideration. Section 3.4.1 defines the implementation of the stemmer.

The amount of different words in the news title corpus was, after stemming, 8,113.

Bigram bag-of-words

To be able to model the word order a bigram bag-of-words was created from the data set. Stemming and lowercasing was used in the bigram
case as well to decrease the sparseness of the arrays and unify flexions
of words.

The same sparseness of the data made n-grams higher than bi-
grams difficult to use. They were tested, but yielded results similar
to a random distribution. The uniqueness of the n-grams higher than
bigrams from the validation set gave no information to the algorithm.

**Special characters**

The presence of special characters might give information about for
example quotes, exclamations and other characteristics. These were
counted and inserted into a feature array, like a bag-of-words but for
special characters.

**General string characteristics**

Since string formatting may differ between the various sentiments, it
is taken into consideration by adding it as a feature vector. Such char-
acteristics are for example letter case and length of the title. A word
is defined as a sequence of characters delimited with space or begin-
ning or end of string. The general string characteristics feature array
consisted of the following items.

- Number of words in title
- Number of characters in title
- Uppercase quota
- Lowercase quota
- Non-alphabetic character quota

**3.2.2 Combinations of features**

The features were combined in feature sets to be able to combine and
compare them. The different feature sets in Table 3.4 were used in
the different experiments. For each feature set both algorithms were
tested.

Feature set 5 was represented as a concatenation of all other feature
vectors.
Table 3.4: The tested feature sets.

<table>
<thead>
<tr>
<th>Set</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Unigram bag-of-words</td>
</tr>
<tr>
<td>2</td>
<td>Bigram bag-of-words</td>
</tr>
<tr>
<td>3</td>
<td>Special characters</td>
</tr>
<tr>
<td>4</td>
<td>General string characteristics</td>
</tr>
<tr>
<td>5</td>
<td>Unigram bag-of-words +</td>
</tr>
<tr>
<td></td>
<td>Bigram bag-of-words +</td>
</tr>
<tr>
<td></td>
<td>General string characteristics +</td>
</tr>
<tr>
<td></td>
<td>Special characters</td>
</tr>
</tbody>
</table>

3.3 Algorithms

A neural network approach and a support vector machine were selected to perform the classification tasks. Their motivations and practical descriptions are presented in this section.

3.3.1 Convolutional neural network

A convolutional neural network was selected as algorithm for the comparison since it has proven well-performing in previous research [10]. Since the study focuses on text representation of different input size, the algorithm must support large dimensions of input data. The convolution reduces the number of dimensions slightly. To ensure stability and correctness of the implementation of the algorithm Microsoft Azure Machine Learning Studio [1] was used to perform the actual learning.

The used neural network consisted of one convolutional layer and one fully connected hidden layer. The output layer included three neurons which each represented one sentiment. The output was decided from the neuron in the output layer with the highest value.

Some of the parameters given to the neural network was tuned during testing. These parameters are algorithm specific and not tied to the feature array. To achieve optimal conditions different parameters of the network was tested and the best scoring model selected. The

[1] https://studio.azureml.net/
scoring was done using the validation data set. The number of hidden nodes in the fully connected layer was varied between 100, 200 and 500, and the number of learning iterations was 20, 50 or 100. For the convolutional layer, which was convolutional from the input, the stride length was varied between 1, 2, 3 and 10. The filter size was varied between 2, 10, 50, 100 and 200.

3.3.2 Support vector machine

A support vector machine approach was selected, since it has proven useful for language classification tasks. Bhayani et al. [5] achieved good results classifying sentiments with it, outperforming both a Maximum Entropy and a Naive Bayes classifier.

The support vector machine also used parameter tuning in Microsoft Azure Machine Learning Studio as implementation. The numbers of iterations was varied between 1, 10 and 100 and the lambda value, the L1 regularization weight, was varied between $10^{-5}$, $10^{-3}$ and $10^{-1}$. The best scoring model was selected for presentation.

3.4 Implementation

The input data sets were constructed by processing the raw string using the Python programming language. The processed data was exported as Comma Separated Values files (CSV), one for the training set and one for the validation set, and uploaded to the Microsoft Azure Machine Learning Studio [15]. The training and test sets were split with stratification to maintain the same proportion of classes. The split was made with 75% of the samples for the training set and 25% for the validation set.

After splitting the data, the features were extracted from the training data set. The training data was then supplied to a model hyperparameter tuner that tested different configurations of the multiclass convolutional neural network. The model producing the best $F_1$-score value was selected.

The output model was scored and evaluated using the validation data converted to a feature array, independent from the training data feature generation.

An overview of the complete experiment process is available in Figure 3.1. The process was repeated for each feature set.
3.4.1 Stemmer

A simple stemmer for the Swedish language was implemented in Python according to the algorithm presented by Boulton [3]. The stemmer was used in all feature pre-processing mechanics including word counts and word order features. However, stemming was not applied before the special characters and general string characteristics feature sets had been produced.

3.5 Evaluation

For each feature set, both classifiers were trained. The different models were evaluated by classifying the validation data set and compare accuracy, recall and precision. The $F_1$-score was calculated for each algorithm. The confusion matrices were also created.

Since the $F_1$-score value includes both recall and precision, it is a useful measure for comparison. It does take the inconsistency in sizes of both training and validation classes into account.

As comparative values, the baselines discussed in section 2.4.3 can be used. The accuracy of the human baseline classifier is calculated from the agreement rate between Annotator 1 and Annotator 2. The

---

**Figure 3.1:** The experiment process for the convolutional neural network approach. The support vector machine was trained in the same way.
majority class baseline is expected to have the same accuracy as the ratio of the largest class.

3.5.1 Feature relevance

To test which language features that had the highest impact on the classification, each feature set was run one extra time for each feature, excluding that particular feature from the training. The macro-averaged $F_1$-score was compared to the macro-averaged $F_1$-score of the whole feature set, and therefore the most impacting features could be selected. The features with highest such difference were interpreted as most impacting.

The feature relevance test was performed using the convolutional neural network.
Chapter 4

Results

The results chapter presents the results obtained from the work presented in the methodology chapter. The sentiment analysis quality is presented in different measures for the different feature sets.

4.1 Confusion matrices

This section presents the exact distribution of verdicts and their real label. The confusion matrices of each feature set and algorithm are presented. A confusion matrix of a perfect predictor is a diagonal matrix with the value 100% across the diagonal.

4.1.1 Feature set 1

The confusion matrix for feature set 1, using the neural network approach with only a unigram bag-of-words, can be found in Table 4.1. The confusion matrix for the same feature set using the support vector machine is presented in Table 4.2.

Table 4.1: The confusion matrix for feature set 1 using a convolutional neural network.

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative</td>
<td>76.4%</td>
<td>12.8%</td>
<td>10.8%</td>
</tr>
<tr>
<td>Neutral</td>
<td>38.5%</td>
<td>44.7%</td>
<td>16.8%</td>
</tr>
<tr>
<td>Positive</td>
<td>17.4%</td>
<td>25.4%</td>
<td>57.2%</td>
</tr>
</tbody>
</table>
Note how the error rate for the positive samples is higher than for the negative ones, due to the distribution of the training data. Table 4.2 shows the confusion matrix using the support vector machine, which performs a little better since less samples are classified wrongly, for all classes.

**Table 4.2:** The confusion matrix for feature set 1 using a support vector machine.

<table>
<thead>
<tr>
<th>Actual class</th>
<th>Predicted class</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative</td>
<td>79.1%</td>
<td>11.6%</td>
<td>9.3%</td>
<td></td>
</tr>
<tr>
<td>Neutral</td>
<td>38.7%</td>
<td>45.1%</td>
<td>16.2%</td>
<td></td>
</tr>
<tr>
<td>Positive</td>
<td>16.8%</td>
<td>25.5%</td>
<td>57.7%</td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.2 Feature set 2

The confusion matrix for feature set 2, with only a bigram bag-of-words, can be found in Table 4.3. The confusion matrix for the same feature set using the support vector machine is presented in Table 4.4.

**Table 4.3:** The confusion matrix for feature set 2 using a convolutional neural network.

<table>
<thead>
<tr>
<th>Actual class</th>
<th>Predicted class</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative</td>
<td>78.6%</td>
<td>7.4%</td>
<td>14.0%</td>
<td></td>
</tr>
<tr>
<td>Neutral</td>
<td>53.7%</td>
<td>23.0%</td>
<td>23.3%</td>
<td></td>
</tr>
<tr>
<td>Positive</td>
<td>37.3%</td>
<td>11.5%</td>
<td>51.2%</td>
<td></td>
</tr>
</tbody>
</table>

Especially notable is the low scores of the neutral class, which is similar for both algorithms. The total amount of neutrally classified samples overall is low. Most are classified as negative, which probably is impacted by the skew distribution of the data set.
Table 4.4: The confusion matrix for feature set 2 using a support vector machine.

<table>
<thead>
<tr>
<th>Actual class</th>
<th>Predicted class</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative</td>
<td>75.1%</td>
<td>9.5%</td>
<td>15.4%</td>
<td></td>
</tr>
<tr>
<td>Neutral</td>
<td>55.3%</td>
<td>20.2%</td>
<td>24.5%</td>
<td></td>
</tr>
<tr>
<td>Positive</td>
<td>38.4%</td>
<td>13.5%</td>
<td>48.1%</td>
<td></td>
</tr>
</tbody>
</table>

4.1.3 Feature set 3

The confusion matrix for feature set 3, with the special characters count array, can be found in Table 4.5. The confusion matrix for the same feature set using the support vector machine is presented in Table 4.6. Since the feature set only contains information about special characters, a worse accuracy than for the two previous feature sets might be expected. However, the matrices show that there is information to gain.

Table 4.5: The confusion matrix for feature set 3 using a convolutional neural network.

<table>
<thead>
<tr>
<th>Actual class</th>
<th>Predicted class</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative</td>
<td>41.1%</td>
<td>30.1%</td>
<td>28.8%</td>
<td></td>
</tr>
<tr>
<td>Neutral</td>
<td>37.7%</td>
<td>29.1%</td>
<td>33.2%</td>
<td></td>
</tr>
<tr>
<td>Positive</td>
<td>32.4%</td>
<td>16.6%</td>
<td>51.0%</td>
<td></td>
</tr>
</tbody>
</table>

Although feature set 3 only contains information about the special characters present in the title, information can be gained and a prediction better than the majority class baseline can be made. Also in this set neutral classifications are underrepresented, which might be a consequence of the data set distribution, but also the difficulty of distinguishing between neutral and polar language.
Table 4.6: The confusion matrix for feature set 3 using a support vector machine.

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative</td>
<td>42.3%</td>
<td>29.1%</td>
<td>28.6%</td>
</tr>
<tr>
<td>Neutral</td>
<td>35.6%</td>
<td>33.1%</td>
<td>31.3%</td>
</tr>
<tr>
<td>Positive</td>
<td>33.9%</td>
<td>18.8%</td>
<td>47.3%</td>
</tr>
</tbody>
</table>

4.1.4 Feature set 4

The confusion matrix for feature set 4, with the general string characteristics, can be found in Table 4.7. The confusion matrix for the same feature set using the support vector machine is presented in Table 4.8.

Table 4.7: The confusion matrix for feature set 4 using a convolutional neural network.

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative</td>
<td>48.3%</td>
<td>29.0%</td>
<td>22.7%</td>
</tr>
<tr>
<td>Neutral</td>
<td>35.8%</td>
<td>33.1%</td>
<td>31.1%</td>
</tr>
<tr>
<td>Positive</td>
<td>30.0%</td>
<td>27.9%</td>
<td>42.1%</td>
</tr>
</tbody>
</table>

The differences between the support vector machine and the neural network are small. The spread between the classes are also larger, which indicates that less information is given about the sentiment of the sentences. However, the negative correctly classified samples are subtly prominent.

Table 4.8: The confusion matrix for feature set 4 using a support vector machine.

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative</td>
<td>46.3%</td>
<td>30.3%</td>
<td>23.4%</td>
</tr>
<tr>
<td>Neutral</td>
<td>38.7%</td>
<td>30.1%</td>
<td>31.2%</td>
</tr>
<tr>
<td>Positive</td>
<td>31.8%</td>
<td>28.1%</td>
<td>40.1%</td>
</tr>
</tbody>
</table>
4.1.5 Feature set 5

Feature set 5 joins the other feature sets together and should contain all the information from them. Both algorithms behave differently when advantaging from the expanded data.

The confusion matrix for feature set 5, with the general string characteristics, can be found in Table 4.9. The confusion matrix for the same feature set using the support vector machine is presented in Table 4.10.

Table 4.9: The confusion matrix for feature set 5 using a convolutional neural network.

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative</td>
<td>78.6%</td>
<td>7.4%</td>
<td>14.0%</td>
</tr>
<tr>
<td>Neutral</td>
<td>19.3%</td>
<td>63.7%</td>
<td>17.0%</td>
</tr>
<tr>
<td>Positive</td>
<td>11.5%</td>
<td>37.3%</td>
<td>51.2%</td>
</tr>
</tbody>
</table>

The neural network approach performs better than the support vector machine, especially when predicting neutral samples. The difference in neutral sample classification is large, where the support vector machine slips back to negative verdicts. However, the accuracy of all classes and both algorithms are at least 50%, which is not observed for any other feature set.

Table 4.10: The confusion matrix for feature set 5 using a support vector machine.

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative</td>
<td>76.1%</td>
<td>7.2%</td>
<td>16.7%</td>
</tr>
<tr>
<td>Neutral</td>
<td>27.7%</td>
<td>52.6%</td>
<td>19.7%</td>
</tr>
<tr>
<td>Positive</td>
<td>30.6%</td>
<td>16.3%</td>
<td>53.1%</td>
</tr>
</tbody>
</table>
4.2 $F_1$-scores and accuracy

The averaged $F_1$-scores and the accuracy of the models are plotted in Figure 4.1. The figure also includes the accuracy of the three baseline classifiers. The randomized guess baseline has an accuracy of 0.35 due to the unevenly distributed data set. The accuracy of the majority class baseline was calculated to 0.45. The human baseline, with an accuracy of 0.78, equals the agreement rate of Annotator 1 and Annotator 2. The $F_1$-scores were not calculated or plotted for the baseline models.

**Figure 4.1:** $F_1$-scores for the different feature sets. The $F_1$-scores for the three baselines were not calculated. CNN represents the convolutional neural network model and SVM the support vector machine.

The neutral network outperforms the support vector machine for feature set 5, which is the largest data set including all the other. The accuracy difference is near 10 percentage points. It is also worth noting that all algorithms and data sets perform better than the majority class baseline.
The human baseline is strong. Only feature set 5, run with the neural network, produces a better accuracy.

In the comparison between the unigram and bigram model, feature set 1 and feature set 2, the unigram approach yields a better accuracy. However, the neural network can benefit from the larger training vectors and perform better than the support vector machine in the bigram case.

4.3 Feature impact

The most significant features of the different feature sets are presented in this section.

4.3.1 Feature set 1

In the unigram feature set, the following features were found to impact the result the most. The values within brackets corresponds to the loss of macro-averaged $F_1$-score when leaving the selected feature out.

1. Count of död (0.012)
2. Count of vann (0.011)
3. Count of man (0.0091)
4. Count of i (0.0087)
5. Count of hyll (0.0085)
6. Count of misstänk (0.0081)
7. Count of klar (0.0080)
8. Count of skad (0.0076)
9. Count of här (0.0073)
10. Count of succ (0.0071)
11. Count of rädd (0.0067)
12. Count of bäst (0.0064)
13. Count of sjukhus (0.0064)
14. Count of int (0.0063)
15. Count of vinn (0.0053)
16. Count of seg (0.0052)

The list of impacting features shows that strong expressions impact the labeling to a large extent. However, less polar words such as the Swedish translations of "man", "in" and "here" are present. In common, they are all often occurring in news titles.

4.3.2 Feature set 2

In the bigram feature set, the features in the following list were found to impact the result the most. The values within brackets corresponds
to the loss of macro-averaged $F_1$-score when leaving the selected feature out.

1. Count of succ för (0.00095) 9. Count of här är (0.00064)
2. Count of död man (0.00089) 10. Count of så gör (0.00060)
3. Count of döm för (0.00089) 11. Count of guid så (0.00059)
4. Count of int grip (0.00089) 12. Count of en son (0.00058)
5. Count of brinn på (0.00082) 13. Count of på sjukhus (0.00054)
6. Count of rädd poäng (0.00071) 14. Count of bäst när (0.00053)
7. Count of i hyll (0.00071) 15. Count of man död (0.00052)
8. Count of klar för (0.00071) 16. Count of till sjukhus (0.00045)

The context of the features are easier to interpret since they consist of two following words, from the nature of bigrams. Swedish translations of "success for", "dead man" and "convicted of" top the list, while only one example of a polarity twisting phrase occurs: "not arrested".

### 4.3.3 Feature set 3

In the special characters feature set, the following features were found to impact the result the most. The values within brackets corresponds to the loss of macro-averaged $F_1$-score when leaving the selected feature out.

1. Count of ! (0.15) 3. Count of ? (0.11)
2. Count of " (0.12) 4. Count of — (0.05)

Exclamations tend to increase the information gain the most, along with quotes and question signs. Moreover, dashes only yields a small improvement to the classification.

### 4.3.4 Feature set 4

In the general string characteristics feature set, the following features were found to impact the result the most. The values within brackets corresponds to the loss of macro-averaged $F_1$-score when leaving the selected feature out.
The uppercase and lowercase ratios equally affect the accuracy, which is logical since they represent the same metric, except special non-alphanumeric characters.

1. Uppercase ratio (0.12)
2. Lowercase ratio (0.12)
3. Non-alphanumeric character ratio (0.040)
4. Number of words in title (0.033)

### 4.3.5 Feature set 5

In the combined, fifth, feature set, the following features were found to impact the result the most. The values within brackets corresponds to the loss of macro-averaged \( F_1 \)-score when leaving the selected feature out.

The list equals the result of feature set 1, but the score differences are smaller.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count of död</td>
<td>0.011</td>
</tr>
<tr>
<td>Count of vann</td>
<td>0.011</td>
</tr>
<tr>
<td>Count of man</td>
<td>0.0086</td>
</tr>
<tr>
<td>Count of i</td>
<td>0.0083</td>
</tr>
<tr>
<td>Count of hyll</td>
<td>0.0081</td>
</tr>
<tr>
<td>Count of misstänk</td>
<td>0.0077</td>
</tr>
<tr>
<td>Count of klar</td>
<td>0.0076</td>
</tr>
<tr>
<td>Count of skad</td>
<td>0.0072</td>
</tr>
<tr>
<td>Count of hår</td>
<td>0.0069</td>
</tr>
<tr>
<td>Count of succ</td>
<td>0.0066</td>
</tr>
<tr>
<td>Count of rädd</td>
<td>0.0063</td>
</tr>
<tr>
<td>Count of bäst</td>
<td>0.0060</td>
</tr>
<tr>
<td>Count of sjukhus</td>
<td>0.0060</td>
</tr>
<tr>
<td>Count of int</td>
<td>0.0059</td>
</tr>
<tr>
<td>Count of vinn</td>
<td>0.0049</td>
</tr>
<tr>
<td>Count of seg</td>
<td>0.0046</td>
</tr>
</tbody>
</table>
Chapter 5

Discussion and conclusions

The results are elaborated and discussed generally in the discussion chapter. Possible sources of errors are presented together with ideas of future work. Ethics and sustainability in connection to the study is discussed.

5.1 Main findings

The problem of understanding meaning of text in a structured way has been researched a lot in recent years. The task of understanding the general temperature of the either spoken or written message, is a more overarching task that may be achieved by not considering all the information of the precise message.

This study has however shown that more information generally improves the results of the classifier when using both a neural network and a support vector machine approach. Despite the large amount of dimensions, overfitting does not seem to occur to a large extent on the created models, considering the accuracy of the validation sets. It would be beneficial to expand the data set in order to further confirm the validity of the models.

As stated in section 1.2, the objective of the thesis was to identify features of news titles that contributed to the sentiment. Using a supervised learning procedure with neural networks and support vector machines, this was investigated. All feature sets gave rise to an information gain helping classification of sentiments. Furthermore, some feature sets increased prediction precision more than others. Generally, more dimensions of the input data, and hence more information
in the input data, produced a more accurate output.

The overall performance of all the classifiers was well above the majority class baseline classifier. This implies that there are information gain in all of the different features, to different extents. Optimally, when used together, they contribute in equal measures to the end result. There are however signs of more important and less important features.

Both bag-of-words approaches perform well. It is however clear that the gain of counting bigrams with the aim of keeping the order of words in mind is arguable. Since the dimensions of the training data increases, the data set needs to be larger to be presentable for the whole data range. When working with small data sets, concise data representation is important for generalization. The bigram representation actually contains the same information as the unigram format, but restructured and extended. The algorithms does not seem to be able to make use of this information, generating a lower accuracy than for the unigram bag-of-words approach in both cases.

5.2 A linear approach

As noticed, the accuracy and $F_1$-scores of the models are quite low. This three class classification problem could also be represented as a linear scale where negative is $-1$ and positive is $1$. Neutral would then intuitively be represented as $0$. A linear approach would classify the titles as of how much they are polarized rather than bringing together nearby titles into the same class. Perhaps this would increase the perceived accuracy. Not all statements are as polar as negative, positive or neutral, but somewhere in between.

If allowing neutral as a valid class of both negative and positive in this study, the result is quite good for all feature sets. For feature set 5, using the neural network, around 13.1% of originally negative or positive classes are classified as the opposite polarity, which gives an imaginary accuracy of around 86.9% in those cases.

Predicting the sentiments using linear regression would be interesting as a comparison to this study. However, since that was out of scope of the main study this was only done experimentally.

A linear regression model was set up in Microsoft Azure Machine Learning Studio for experimental purposes. Brief testing shows that a
mean absolute error of 0.69 is easily achieved for the same data set as used for feature set 1 in this study, using stochastic gradient descent. The achieved mean absolute error might seem high, since the range of the classification is 2 units wide. However, classifications in between the discrete classes will cause errors to arise.

5.3 Classifier characteristics

As observed from the results presented in Figure 4.1, the unigram bag-of-words data set produces the best scores for the feature sets that are not combined. This applies to both the support vector machine and neural network approaches. Compared to feature set 2, where bigrams were used instead, the difference is significant. A reason for this is most probably bad generalization in the bigram case. The training data set is thus too small for the desired data representation.

It is clear from section 4.3 that words connected to strong emotional events contributes to the classification a lot. The Swedish words for death, win, suspect and injure are all represented in the top features for the combined feature set 5. When other characteristics than unigram counts are included, they do not make it to the top of the list.

Considering the bigram case, it is not strange that no feature is better than any unigram count. The bigram feature vector is much larger, and only part of the information is removed, since the words may occur on other places within the vector.

Regarding the general string characteristics and special character occurrences, they do not contribute a lot to the classification.

5.4 Sources of error

One of the advantages with machine learning approaches in contrast to traditional detailed algorithm functions, is that the exact execution steps do not need to be taken into consideration. The data provided to the algorithm forms the model, which allows some input to be incorrectly classified. If the data is not correctly labeled, the model will be affected. Not only will the validation scoring fail if it contains wrongly labeled data, but the model itself may be biased from erroneous training data.
Moreover, the data set in this study is limited in size. Convolutional neural networks require large amounts of data. The experiments would probably have benefited from a larger data set, as more training data would have given better opportunities to form the model. However, if the labeling of more data had involved more than one human labeler, the quality of the data might have decreased.

The data was gathered from a Swedish news paper and classified by a single person. It is of great importance that the training data is consistently labeled, with minimized noise, since it otherwise might be impossible to generalize the data. How the data actually is labeled and the requirements for labeling, is of less importance than a consistent labeling.

5.5 Comparison to other studies

The results of this work shows that Swedish news article titles in many ways behaves similar to English sentences. For example, analyzing Twitter data using a unigram and a tree kernel model [2], resulted in an accuracy around 75%. That equals the average accuracy of feature sets 1 and 5 in this study. Using a neural network or a support vector machine can therefore be seen as a possible solution to future problems.

However, the addition of features in the case of unigram and tree kernel models [2] did not cause the same improvement as adding features in this work. This is especially the case for the neural network approach. Reasons for this might be that the convolutionality of the network helps reducing dimensionality to handle larger feature vectors.

Hovy et al. [9] presented a more traditional solution to the sentiment analysis problem by combining different sentence features in an algorithmic way, without the help of machine learning. The two elaborated model algorithms from this study both outperform that traditional approach, if evaluated with strict agreement. However, Hovy et al. takes the lenient agreement into consideration, which includes the neutral class in the positive class. That would be a simple extension to this work, which could yield interesting results and more comparative data.
5.6 Sustainability and ethical aspects

In any application involving data collection there is always an ethical aspect to take into consideration. The sentiment analysis itself is not as prone to this discussion as the result it produces. Since the result can be used to improve user experiences, it can also be used to influence users in a concealed way, which can be classified as an integrity violation.

Since the supervised approach presented in this thesis requires manually classified data, the results of it depend on the opinions of the annotator performing the training data labeling. Albeit the output itself does not directly affect the user, the result of the usage of it might do. Subconscious opinion learning is a risk that cannot be overlooked.

If connected to other services or databases, sentiment analysis of the content consumed by single users can be used to determine for example political orientation. Used wrongly or by dishonest actors this can be used against basic democratic rights. It is not the sentiment analysis itself that could be exploited, but the result of it. The risk of being used in these kinds of situations is probably low, since there are other methods of gaining the same information that may produce better results, at least for individual cases. In the light of recent accusations of governmental involvement in foreign elections, however, this must be considered.

More use cases of sentiment analysis of news article titles are yet to be discovered. Advertisement and offer optimizing using the polarity of the content is one way. When users are faced with relevant information faster, more time can be spent on more rewarding activities, leading to an increased social sustainability. Contrary, opinion mining can be connected with risks if used by for example authorities for wrong reasons.

In the case of advertisement optimizing, the financial perspective of sustainability and applications can be observed. Generating turnover from showing more relevant advertisements benefits both users and content providers. In the long run, this can act as a source of income to provide services to users without direct charges.

The conversation climate on the Internet is often described as aggressive and sometimes rude. With the protection from screens and anonymity people tend to say things they would not say in person. Language technology in general and sentiment analysis in particular can help identifying such language to act on. Language can be ana-
lyzed prior to publishing in order to ensure correctness and intention. The application is of course not limited to these cases, but can be inspired from them.

Considering environmental sustainability, there are many possible use cases within the machine learning area. However, the sentiment analysis area is not directly connected to it. Indirectly, more information about all kinds of items can increase the opportunities of a sustainable future. Using electricity to power the algorithms is the largest cost, which must be weighted against the advantages and use cases.

5.7 Future work

The bag-of-words approach to the sentiment analysis worked well for this kind of data. However, the domain of news titles is very limited. The number of different words is small and the sentences themselves are short. This study does not try to generalize the results to other data sets. Larger domains would probably be hard to train since the training data would need to be many times bigger. There is no way to extend the length of each title. The only way to increase the data set is to gather and annotate a higher quantity.

Many of the sources of error could have been decreased in amplitude by a larger training set. It would therefore be fruitful to perform a similar study but with a larger data set.

A natural expansion of the project would be to investigate how trigrams and higher n-grams could affect the classification. However, that will probably not be rewarding without a considerably larger data set. Improving the stemming algorithm to use pre-compiled word lists might also produce more accurate word stems. A better stemmer could lead to more relevant results since clearly different words otherwise could be unified in the stemming process.

A comparison with another domain in contrast to the news title one could also be valuable. The Swedish language is used differently in other contexts, and news titles are seldom multi-polar or has a turned polarity. Longer sentences or texts may confuse the classifier which must better understand the context of words. The labels would probably also need to be extended, since a single sentiment may not be enough if texts change somewhere between the beginning and the end. As seen in previous studies, a fourth *both* label can be used.
Sentiment analysis of the Swedish language has not yet been investigated widely. It would be interesting to compare results from Swedish analyzers with predictors for other languages to identify connections or possible problems arising.

5.8 Conclusions

For Swedish news article titles, a bag-of-words in a unigram format performs well for training a machine learning model for a sentiment classification task. No more information is added in the bigram case, which however does not decrease the precision when combined with the unigram bag-of-words.

A simple neural network model does perform well for this classification task. However, a support vector machine does yield similar results unless the dimensions of the input grows. Even with small efforts and feature vectors, including general string characteristics, the classification gains information.
Bibliography


