Computational Analysis of Swedish Newspapers
Using Topic Detection and Sentiment Analysis

Simon Wallbing
Abstract

Computational Analysis of Swedish Newspapers

Simon Wallbing

Newspapers might report on the same event, say a sport event or a political statement, but since they most likely differ in the presentation, are the content and under laying message of the articles actually the same? A human can read two separate articles and determine if they touch similar subjects and if they approach the subject in a positive or negative way. If this comparison would be performed over several thousand of articles a computer would very much be the preferred method. However, a computer needs to be trained to understand the topics of the articles to be able to detect the topics and make the comparison.

The two goals of this project is to find and identify topics within articles extracted from Swedish newspapers as well as preforming sentiment analysis on the most similar topic pairs. This project presents a Python 3 implementation of extracting textual data from Swedish newspapers, identify and assign topics to those articles, as well as preform sentiment analysis on articles based on their topics and day of publication. To extract the text from each article web scraping was used. The topic detection was performed with the help of non-negative factorisation matrices. To determine each article polarity and emotional state TextBlob was utilised.

Both goals were accomplished. The method used to extract textual data was successful and topics for each article was successfully identified. The topic detection and sentiment analysis proved to be mostly correct while manually inspecting the most similar article pairs between the newspapers. The result was presented with dumbbell plots for the most similar article pairs. These plots shows each pairs polarity and subjectivity score and was therefore used to manually analyse the actual similarity between these articles as well as to their sentimetic structure.

However, the results are deemed to be too unreliable to draw any significant conclusion in the sentiment difference and likeliness between the newspapers. This is because of the absence of a proper implementation of Swedish part-to-speech tags and lemmatization, which was noticed too late into the development process to be able to correct. These changes are however discussed and reflected upon in the purpose to gain insight in how the implemented solution could have been improved.
## Contents

1 Introduction
   1.1 Purpose and Goals ........................................... 4
   1.2 Methodology ................................................. 5
   1.3 Outline ..................................................... 5

2 Related Work
   2.1 Detecting Trending Topic on Chatter ......................... 7
   2.2 Sentiment Analysis for Tweets in Swedish ...................... 7

3 Data Set and Methodology
   3.1 Web Scraping ................................................ 10
   3.2 Preprocessing ............................................... 10
   3.3 Document-term matrix ....................................... 12
   3.4 Data Statistics .............................................. 12

4 Topic identification & similarity
   4.1 Non-negative Matrix Factorisation .......................... 13
   4.2 Similarity Index ............................................ 15

5 Results ................................................................ 17

6 Discussion
   6.1 Textual similarities ......................................... 19

7 Conclusion
   7.1 Future Work .................................................. 22
   7.2 Limitations .................................................. 23
   7.3 Lessons Learned ............................................ 24

References .................................................................. 26
1 Introduction

A lot of people get their news from newspapers, either physical or digital, and they might only subscribe to one of the local newspapers, especially if they are physical ones. Some newspapers might report on the same event, say a sport event or a political statement, but since they most likely differ in the presentation, are the content and underlying message of the articles actually the same?

It’s quite easy for someone to read two separate articles and determine if they touch similar subjects, such as sports results, political questions, or entertainment, and/or if they approach the subject in a positive or negative way. If you want to perform this analysis over several hundred or even thousand of articles a computer would very much be the preferred method. However, a computer needs to be trained to understand the topics of the articles to be able to detect its topic and make the comparison.

By using something called topic modeling[1] a program can identify and assign topics to these articles. We now have something a computer can work with and can therefore start comparing articles to find if they touch similar topics. This method can however not determine what under-laying opinions and emotions that the written may or may not have towards some topic. This is where a text analysis technique called sentiment analysis comes in. Sentiment analysis, also called opinion mining, is a way to analyse a text and identify if that text is objective or not and how opinionated it is.

By utilising these two concept a computer can be used to systematically compare huge amount of textual data and identify similarities.

1.1 Purpose and Goals

The purpose of this project is to use computational methods to train a system so that it can recognise topics from Swedish news-articles. Using these methods I want to know how the newspapers compare to each other. Are one paper more positive towards a certain topic? Do the papers stay objective or do they show some kind of bias? In short, the two points below defines the goals of this project.

1. Identifying topics of Swedish news-articles using topic modeling.

2. Compare articles with similar topics from two newspaper and analys if their sentiment structure differs.

How similar the topics needs to be to be included will be presented in Section 4.2
1.2 Methodology

There are several steps that need to be completed to be able to investigate the points mentioned in Section 1.1. The first is to determine and obtain the data set, that will serve as a basis for all future calculations and assumptions during this project. This will be done using web scraping, a method used to extract and download content directly from websites. However, which sites, henceforth called journals, the data set will be acquired from will need to be chosen before the web scraping method can be implemented. By choosing two journals that have outspoken different political standpoints from each other, I hope to gain the most interesting result. After two journals have been chosen I can continue with extracting the textual data from as many articles as possible.

To determine these topics a statistical model called topic modeling will be used. Topic modeling is a method used for automatically find textual structure in a collection of text documents. It will be utilise to discover topics for each article from the newspapers.

After the articles topics has been determined, sentiment analysis will be applied to the data set. Sentiment analysis is a text analysis method of detecting and identifying the under-laying subjectivity and emotional state, positive or negative, of a text. This will be used as a final summarisation of the articles and provides the final overview of the processed data.

1.3 Outline

The introduction at Section 1 consist of goals, purpose, and methodology of the project. Section 2 covers similar projects as well as everything that has been used during the project, tools, programming languages or libraries, and methods. Section 3 describes the data set and how it was attained. In Section 4, the actual analysis method of the textual data will be presented. Finally, Section 7 will present the result and discuss conclusion and future work will be presented in Section 7.1.
2 Related Work

All code produced during this project was written in Python 3[2] and utilises some standard libraries, sys and os, aswell as some more specialised libraries.

The web scraping part of the code uses selenium webdriver[3] and pandas[4]. Selenium WebDriver was used to automate the downloading process of textual data thru the Firefox[5] a web browser. This was done by opening an instance of a Firefox browser, finding the element within the sites HTML code where the desired textual data resides, and stores the raw text data to a local pickle file (suffix .pkl) using DataFrames from the pandas library. Pandas contains tools for reading and writing data between in-memory structures and local files, such as text files, CSV, and pickle.[4] Dataframes was used to organise the obtained data into a file format that was easy to work with, e.g. in a matrix format. By saving the different kind of data into column it is possible to easily obtain sets of data again by reading the file again.

A preprocessing algorithm, which utilised the nltk library[6], was used on the downloaded data to prepare it for further processing. The preprocessing algorithm is further explained at Section 3.2. The processed data was then converted into a Document-term Matrix, explained in Section 3.3, using CountVectorizer from the scikit-learn library.[7] Results are saves locally using the same methods as before.

To find and determine the actual topics, sklearn’s decomposition module was used. The decomposition module includes algorithms for breaking down matrices into smaller matrices.[8] The actual method used is called Non-Negative Matrice Factorization (NMF) and is explained in Section 4.1. With the topics identified each article from both journals have their topics compared with eachother using a jaccard index[6] calculation. The articles was also compared with regard of their publication date using a linear function described in Section 4.2.

TextBlob is a library used for processing textual data[9] and was used to perform sentiment analysis with its functions sentiment.polarity and sentiment.subjectivity. The polarity function gives a float number between −1.0 and 1.0, where lower numbers indicates a negative emotional state and higher numbers indicates a positive one. The subjectivity function returns a float number between 0.0 and 1.0, where 0.0 indicates objectivity and 1.0 indicates subjectivity. Both functions was run on the top 28 pairs of articles that gave the highest jaccard index and publication date similarity.

The final result is visualised as dumbbell plots using the matplotlib library, which is a comprehensive library for creating a range of different visualisations in Python.[10] By using these plots, the final data can easily be presented and allows for very easy comparisons between the articles.
2.1 Detecting Trending Topic on Chatter

'The amount of posts on social network is overwhelming: for example Twitter has more than 50 millions posts a day. It has become crucial to be able to sort them. By detecting trending topics, which are topics the most discussed on a social network, we allow the user to instantly know what is happening in the network and if he is interesting in one topic, he can get access to all the posts related to this topic. In this work we present and compare different algorithms to detect trending topics. Our approach is to compute similarities between posts and then to find clusters in the graph of similarities using clustering algorithms.'[11]

This papers have some similarities with what I want to do, mainly identifying topics from textual data. However, the difference is primarily that it focuses on the actual algorithms that detects the topics where this project chooses an appropriate algorithm to be able to continue with point two mentioned in Section 1.1. Also, this project does not utilises clusters or clustering algorithms. It instead uses different methods to compute the similarities between articles, that will be further explained in Section 4.

2.2 Sentiment Analysis for Tweets in Swedish

'Sentiment Analysis refers to the extraction of opinion and emotion from data. In its simplest form, an application estimates a sentence and labels it with a positive or negative sentiment score. One way of doing this is through a lexicon of sentiment-laden words, each annotated with its respective polarity. Tweets are a specific kind of data that has spurred interest in researchers, since they tend to carry opinions on various topics, such as political parties, stocks or commercial brands. Tools and libraries have been developed for analyzing the sentiment of tweets and other kinds of data, but mainly for the English language. This report investigates ways of efficiently analyzing the sentiment of tweets written in Swedish. A sentiment lexicon translated from English to Swedish, together with different combinations of syntax rules, is tested on a labeled set of tweets. Machine-translating a lexicon did not provide a fully satisfying result for sentiment analysis in Swedish. However, the resulting model could be used as a base for constructing a more successful tool.'[12]

Contains overlapping aspects, such as sentiment analysis of texts written in Swedish, but has a focus on finding 'efficient method for extracting opinions'[12]. This project are however more focused on the results gained from such a method rather then investigating the method itself.
3 Data Set and Methodology

Like mentioned before, this project aims to implement a solution of identifying topics of Swedish news-articles and then compare and analyse their sentimentic structure. These articles will be extracted from two journals. The articles from each journal will then be compared based on the identified topics as well as when they were published, so that current underlaying biases and ideas are captured as well. This is to make the articles as comparable as possible. The most similar articles will be paired together and be run through a sentiment analysis algorithm. The pairs will be manually checked to see if they make sense or not and a selected few will be discussed to formulate a final result.

The two journals are Aftonbladet[13] and Expressen[14]. These were chosen for a few reasons. The first is that both journals are well known by the Swedish people since they were established several decades ago, Aftonbladet in 1830 and Expressen in 1944. Both journals are viewable on the web, which is an necessity for this project, and both also have relatively easy to navigate HTML code, which simplifies the data extraction stage of this project. Presented below are some information and screenshots of both journals used during this project.

![Figure 1](image.png)  
**Figure 1:** Screenshot of the Swedish newspaper Aftonbladet.
Above shows an example of the frontpage of Aftonbladet on the 5th of August 2020. Aftonbladet is an evening paper with the political designation "independent social democratic"[15].

![Figure 2: Screenshot of the Swedish newspaper Expressen.](image)

Above shows an example of the frontpage of Expressen, also taken on the 5th of August 2020. Expressen is a Swedish evening paper with the political designation "unbound liberal"[16]. Since both journals have a similar layout, appearance, and content published around the same time they are in fact quite comparable and therefore very well suited for this project. In the same way that articles extracted within the same time-period have the purpose to find similarities, picking two journals with different political designation have the purpose to find dissimilarities. In short, the articles need to be similar enough to be able to be compared but dissimilar enough to be able to gain valuable result.
The general workflow of extracting and preprocess all articles into a usable Document-Term Matrix (DTM), explained further in Section 3.3, is illustrated below. Articles from newspaper A and B are comprised into a large document containing every article from both A and B and then processed into a Document-term Matrix.

![Workflow Illustration](image)

**Figure 3:** Workflow illustration

### 3.1 Web Scraping

Web scraping is a method to extract data from websites. Every article for each newspaper is stored locally as pickle files, as shown in Table 1. Column *snippet* is the first 80 characters of each article’s *text* and is used as an identifier to the articles. *text* contains the actual text of each article, *link* contains each article’s URL, and *date* is each article’s publication date.

<table>
<thead>
<tr>
<th>snippet</th>
<th>text</th>
<th>link</th>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snippet 1</td>
<td>Text 1</td>
<td>Link 1</td>
<td>Date 1</td>
</tr>
<tr>
<td>Snippet 2</td>
<td>Text 2</td>
<td>Link 2</td>
<td>Date 2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Snippet n</td>
<td>Text n</td>
<td>Link n</td>
<td>Date n</td>
</tr>
</tbody>
</table>

**Table 1:** Raw data pickle file.

### 3.2 Preprocessing

Text preprocessing serves a major role in topic modeling as it is used to clean the textual data into something that can be represented as topics. Without any preprocessing steps
the results would be significantly less accurate.[17] Therefore, every article text was run through a cleaning algorithm that did the following:

1. Tokenize the text into smaller parts. First separating the whole text into sentences, then converting each sentence into tokens that are separated by whitespace. Note that the tokens are not necessarily just words but might contain or be special characters and/or numbers, such as "code.", "-", or "1976".

2. Remove Swedish stop-words and words from a manually defined exclusion list. Names were however not removed.

3. Make every word lowercase.

4. Remove special characters, such as \., \+, /., [., (.), !, \?.

5. Remove numbers.

6. Exclude articles with a character count lower than 500 since it is presumed that they do not contain enough information to be of use.

7. Make a corpus over all the words. A corpus is a collection of documents in the format of Bag of Words (BoW). Bag of Words can be described as a list of words (tokens) paired with the number of occurrences of that word.[17] The sentence "Code is great, code is good." would look like this: \{"code": 2, "is": 2, "great": 1, "good": 1\}. The semantic structure of the sentence is not important.

The manually defined exclusion words are as follows: via, expressen, expressens, sportexpressen, expressentv, aftonbladet, oh, år, bit, aftonbladets, se, rss, https, pages, section, rss2, 0481d, 01d40, small, tipsa, sportbladet, sportbladets, säsong, säsongen, netflix. These words were excluded because they are either uninteresting, such as the name of the newspapers, or are some kind of tag related to the site's HTML code.
3.3 Document-term matrix

The processed data is then converted into a more appropriate format called a document-term matrix as illustrated in Table 2. A document-term matrix is a matrix that utilises the bag of words of each article, summarising every article’s bag of words into one, large matrix.

<table>
<thead>
<tr>
<th>snippet</th>
<th>word 1</th>
<th>word 2</th>
<th>...</th>
<th>word n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Article snippet 1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>Article snippet 2</td>
<td>0</td>
<td>2</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Article snippet n</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2: Document-term matrix

3.4 Data Statistics

The total number of articles used was 1,736 and they were extracted from each site across seven days. In Table 3, the distribution of the number of articles gathered from each journal as well as section distribution in each journal.

<table>
<thead>
<tr>
<th></th>
<th>Expressen</th>
<th>Aftonbladet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Culture</td>
<td>314</td>
<td>281</td>
</tr>
<tr>
<td>Politics</td>
<td>217</td>
<td>284</td>
</tr>
<tr>
<td>Entertainment</td>
<td>283</td>
<td>99</td>
</tr>
<tr>
<td>Sports</td>
<td>217</td>
<td>118</td>
</tr>
<tr>
<td>Total per journal</td>
<td>954</td>
<td>782</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>1,736</td>
</tr>
</tbody>
</table>

Table 3: Distribution of the extracted articles.
4 Topic identification & similarity

Identifying topics from a text corpus is not an easy endeavour from a text mining perspective. There are however multiple ways of doing topic modeling, such as probabilistic methods[18] that will be further discussed in Section 7.1. However, for this project something called matrix factorisation, or more specifically Non-negative Matrix Factorisation (NMF), will be utilised. The Python library scikit-learn was used to apply NMF. Using this method, producing a desired result will hopefully be possible without implementation complications that may have occurred with a probabilistic method.

4.1 Non-negative Matrix Factorisation

Given an DTM, NMF will results in two separate matrices, $W$ and $H$, where $W$ is a documents, in this case articles, to topics matrix and $H$ is a topics to term matrix. the values within each matrix represent the weighted value of how relevant each topic is to each article and term respectively. The number of topics generated was set to 20 and the number of terms describing every topic was also set to 20. Values in both tables are representations and not actual values.

<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>...</th>
<th>Topic m</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.2</td>
<td>...</td>
<td>1.0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0</td>
<td>...</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>...</td>
<td>0.5</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>...</td>
<td>0.8</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0.0</td>
<td>0.1</td>
<td>...</td>
<td>0.4</td>
</tr>
<tr>
<td>Article 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Matrix W
Weights for n articles relative to m topics

<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>...</th>
<th>Topic m</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.2</td>
<td>...</td>
<td>1.0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0</td>
<td>...</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>...</td>
<td>0.5</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>...</td>
<td>0.8</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0.0</td>
<td>0.1</td>
<td>...</td>
<td>0.4</td>
</tr>
<tr>
<td>Term 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Matrix H
Weights for n terms relative to m topics

From the topic to term matrix $H$, each topic can be represented as a graph, as shown below. Using this information it is possible for a human to determine what each topic most likely concerns.
Figure 4: Topic nr.4 for Aftonbladet

Figure 5: Topic nr.18 for Expressen
The two graphs above shows the terms which constitutes a topic. Each term has a weight, meaning how much that term describes the topic. In Figure 4, Eva Illouz, who is a professor in sociology, have a high weighted value so together with other terms such as “relationer” (relations), “människor” (humans), and “samhälle” (society) a general idea of the topic can be formed. This information can therefore and was used to validate that the generated topics are reasonable for a human to understand.

The document to topic matrix W was used to find the most relevant topics for each article.

4.2 Similarity Index

In this project a similarity index is applied on two factors, the publication date of each article and each articles most relevant topics.

Every articles top topic and publication date, which is obtained from the raw data matrix, is run through a small algorithm that generates two matrices; \( D \) and \( T \). Note that \( D \) and \( T \) are not to be confused with matrices W and H since they are not used for the actual topic detections, but are instead used solely to find each articles most similar partner. So \( D \) shows how similar each article from one journal are to every other article from the other journal is in terms of publication date and \( T \) shows the same for topics. \( D \) contains the index value between every articles publication date. This is calculated using the linear equation 1, which results in a value of 1 if the two articles are published on the same date, and a value of 0 if the difference in publication date is more then 7 days. \( a \) and \( b \) is the publication date of each article. The difference between dates are measured in whole days.

\[
date \ similarity = 1 - \frac{|a - b|}{7} \quad (1)
\]

The values of matrix \( T \) is calculated using cosine similarity, which is a measurement of similarity between two non-zero vectors.[19]

\[
topic \ similarity = \cos \theta = \frac{A \cdot B}{\|A\|\|B\|} \quad (2)
\]

Vectors \( A \) and \( B \) are generated from matrix W, as described in Section 4.1, by individually sorting the rows to get each articles top four topics. Each article are then assigned a vector \( A, B \) (one for each journal) containing those values, giving a vector such as \( A = \{1, 0.8, 0.8, 0.7\} \). Using these vectors, the cosine similarity can be calculated for
between every article for each journal.

A third matrix $F$ is then created by adding $D$ and $T$, resulting in a final score matrix. The score threshold for an article pair to be included in the sentiment analysis is set as the article pairs with the highest score. Every article pair will then have the same score or just below if there are too few pairs with that score.

Table 6 shows the final scores of the top 29 article pairs. $Pairs$ contains the articles index from the raw data, as shown in Table 1, in the following format: [Expressen Aftonbladet]. $Topic Score$ is the cosine similarity of the article pair and $Date Score$ is the publication date similarity. $Final Score$ is the summarised score of $Topic$- and $Date Score$.

Since all pairs had a high final score, no pair will be excluded from the consideration for the sentiment analysis. The pairs in Table 6 have been manually picked and inspected to verify that they touch similar topics. The manual inspection covered the top 40 and the bottom 10 pairs. The bottom 10 was included to try to get a broader understanding of the result. Of these 50 pairs, 29 pairs was evaluated as being correctly identified by the methods used to be about the same topic. This results in a correctness percentage of 58%, which for this project is seen as sufficiently high.

<table>
<thead>
<tr>
<th>Pair</th>
<th>Final Score</th>
<th>Topic Score</th>
<th>Date Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>698 546</td>
<td>1.886521505</td>
<td>0.886521505</td>
<td>1</td>
</tr>
<tr>
<td>698 537</td>
<td>1.888153327</td>
<td>0.888153327</td>
<td>1</td>
</tr>
<tr>
<td>698 270</td>
<td>1.889638882</td>
<td>0.889638882</td>
<td>1</td>
</tr>
<tr>
<td>702 529</td>
<td>0.914419064</td>
<td>0.914419064</td>
<td>1</td>
</tr>
<tr>
<td>927 765</td>
<td>0.926306769</td>
<td>0.926306769</td>
<td>1</td>
</tr>
<tr>
<td>198 225</td>
<td>0.999942757</td>
<td>0.999942757</td>
<td>1</td>
</tr>
<tr>
<td>218 8</td>
<td>0.999942566</td>
<td>0.999942566</td>
<td>1</td>
</tr>
<tr>
<td>241 446</td>
<td>0.99997196</td>
<td>0.999971958</td>
<td>1</td>
</tr>
<tr>
<td>241 483</td>
<td>0.999973309</td>
<td>0.999973309</td>
<td>1</td>
</tr>
<tr>
<td>374 122</td>
<td>0.999936618</td>
<td>0.999936618</td>
<td>1</td>
</tr>
<tr>
<td>374 414</td>
<td>0.999929403</td>
<td>0.999929403</td>
<td>1</td>
</tr>
<tr>
<td>396 417</td>
<td>0.99992826</td>
<td>0.999928259</td>
<td>1</td>
</tr>
<tr>
<td>397 116</td>
<td>0.999968941</td>
<td>0.999968941</td>
<td>1</td>
</tr>
<tr>
<td>418 120</td>
<td>0.999983466</td>
<td>0.999983466</td>
<td>1</td>
</tr>
<tr>
<td>418 605</td>
<td>0.999916784</td>
<td>0.999916784</td>
<td>1</td>
</tr>
<tr>
<td>419 604</td>
<td>0.999936764</td>
<td>0.999936764</td>
<td>1</td>
</tr>
<tr>
<td>442 112</td>
<td>0.999941483</td>
<td>0.999941483</td>
<td>1</td>
</tr>
<tr>
<td>443 119</td>
<td>0.999980139</td>
<td>0.999980139</td>
<td>1</td>
</tr>
<tr>
<td>443 606</td>
<td>0.999938216</td>
<td>0.999938216</td>
<td>1</td>
</tr>
<tr>
<td>445 80</td>
<td>0.999898232</td>
<td>0.999898232</td>
<td>1</td>
</tr>
<tr>
<td>556 340</td>
<td>0.999913986</td>
<td>0.999913986</td>
<td>1</td>
</tr>
<tr>
<td>664 500</td>
<td>0.999913313</td>
<td>0.999913313</td>
<td>1</td>
</tr>
<tr>
<td>722 206</td>
<td>0.999927472</td>
<td>0.999927472</td>
<td>1</td>
</tr>
<tr>
<td>831 137</td>
<td>0.999925288</td>
<td>0.999925287</td>
<td>1</td>
</tr>
<tr>
<td>889 418</td>
<td>0.999974182</td>
<td>0.999974182</td>
<td>1</td>
</tr>
<tr>
<td>890 188</td>
<td>0.999948333</td>
<td>0.999948332</td>
<td>1</td>
</tr>
<tr>
<td>909 204</td>
<td>0.999982713</td>
<td>0.999982713</td>
<td>1</td>
</tr>
<tr>
<td>909 499</td>
<td>0.999899218</td>
<td>0.999899218</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6: Top 29 article pairs with the highest final score.
5 Results

Presented below is the result of the sentiment analysis visualised as dumbbell plots, one for the articles polarity and one for the subjectivity.

Look at the article pairs from Section 6.1, (418, 120) and (397, 116). Article 396 has a polarity score of around 0, which seems kind of reasonable by looking at its snippet. It mostly talks about history and what happened but I would argue that it leans towards the positive side, since the freon restrictions is a good thing for the ozone layer. However, like discussed before, it can be difficult to pinpoint an articles topic, and by extension its polarity, just from the snippet. Article 116 looks to be slightly more positive then its partner. The same logic already mentioned before can also be applied here however.

Article pair 418, 120 have more diverse result. Article 120 looks to be quite a lot more positive then 418. This actually make a lot of sense when looking at the snippet. 418 talks about a proposal that would greatly benefits some countries during the COVID-19 pandemic and help them get back at their feet, while 120 have negative choice of words and a message that feels hostile.
The subjectivity scores of both pair 418, 120 and 397, 116 fluctuate more than the polarity scores. Even though article 397 has a polarity score of around 0 its subjectivity score are relatively high. Article 116 are more factual in its presentation while being just above 397 in polarity score. Article 120 has a subjectivity score of around 0.2 while staying quite neutral, unlike its partner 418 with its polarity score of around 0.2. 418 has a higher score in subjectivity.
6 Discussion

As seen in Table 6 the majority of the pairs topic score are rather similar, since they differ within a span of around 0.11. This might very well be an result of the chosen number of generated topics, as mentioned in Section 4. Since there are 20 topics consisting of a relatively low term count, the calculated values from equation 2 does not have a very large set of possible values. Also, the number of top topics for each article are four, as described below equation 2. This might further contribute to the final scores being so similar.

The date score are all 1, not only among the manually evaluated pairs but also among all pairs. Equation 1 is simply an linear equation with the possible values of 0, 1, 2, 3, 4, 5, 6, and 7. Just by the small set of possible values the final score is significantly affected, especially since the articles where only sampled within a time period of one week. This results in a very high probability that the top scoring article pairs was published on the same day, since the articles are not spread out within a larges time-period. Alternatively, the date could have been represented with 24 hours instead of just whole days. This would most likely result in a more varied presentation of the date similarity since articles published on the same day could take on more values rather then just 1.

Very few article pairs had the same top topics as their partner, none within the manually inspected ones. However, there were a pattern that the majority of topics assigned to Expressen were similar, or even the same, as a lot of other articles from Expressen. This might very well be an result from how the pairing is done. The pairing algorithm iterates over every article from Expressen and tries to find the top scoring article from Aftonbladet to pair up with. In other words, the algorithm uses articles from Expressen as a basis and operates from that.

6.1 Textual similarities

Below are a few example article pairs and about the first 80 character of each article. The first two are article pair 418, 120 with a final score of 1.999983466, and the other two are pair 397, 116 with a score of 1.999968941.

"Coronautbrottet har inte inneburit någon lysande reklam för EU-samarbetet. Italiens böner om hjälp förblev länge ohörda, Tyskland och Frankrike har beslagtagit medicinsk utrustning ämnad för export och trots ett bup-ras av historiska mått har EU-ledarna misslyckats med att hitta gemensamma lösninagar för att mildra den ekonomiska krisen. Men nu skymtar en utväg. Tysklands Angela Merkel och Frankrikes Emmanuel Macron har föreslagit att EU ska låna upp pengar i syfte att ge bidrag till de värst drabbade medlem-
sländerna. Utspelet var uppenbarligen synkroniserat med EU-kommissionen, som på onsdagen föreslog att unionen ska låna till en stödfond om hela 750 miljarder euro – varav två tredjedelar ska delas ut som bidrag. Kanske beror Merkels n-sväng på att hon ser att coronautbrottet är en existentiell kris för EU. I Italien har den uteblivna hjälpen inneburit att stödet för EU har sjunkit kraftigt under pandemin. Att i det här läget bara erbjuda lån med villkor om härda strukturreformer – l...

The above article snippet touches how different countries in EU-union have a tough time with the COVID-19 outbreak. It talks about a new proposal that EU should provide financial help to some countries that have it particularly rough during the pandemic so their economy does not completely crashes. So topics assigned to this is economics, EU, and COVID-19.


This snippets talks about how society have changed during the COVID-19 pandemic. It reflects about how people reacts and thinks regarding other citizens and how some simple things as taking a beer at the pub is meet with hostility, since it does not comply with the new norm of social distancing. Some topics for this one would be COVID-19, society norms, and social hostility.

I would argue that the above two articles are about the same topic and with a topic similarity score of 0.999983466, so does the topic detection method.

Gore, knappt tre år senare blev han USA:s vicepresident. Sverige stod då bara för en bräkdel av freonutsläppen i världen. Men det svenska förbudet bidrog inom några år till internationella avtal om total utfasning. Sverige hade helt enkelt visat hur en realistisk avvecklingsplan kunde se ut. När politiker säger att Sverige ska ”gå före” i någon fråga framstår det lätt som fluffigt och idealistiskt. Men i den nyutkomna boken Miljöframgångar - från freonförbud till klimatlag skriver S-märkta Mats Engström, med ett förflutet på miljödepartementet, om ett antal tillfällen då Sverige genom kloka reformer faktiskt har banat väg för..." - Expressen, article 397

The above snippet is a reflection on how Sweden prohibited freon in 1989 and how, a few years later, the rest of the world followed. It continues to talk about other accomplishment that pushed the world forward towards climate-smart decisions. Topics would be foreign policy, environment, and Sweden.


Article snippet for 116 discusses a book by Stefan Jonsson. This book analyses the world "fronside" and "backside, i.e how the world can be perceived - world peace - vs how it actually is - the bombing of Pakistan. These are a few topics that could be assigned to these articles; literature and social analysis.

Because of the different topics categories these articles might not actually be about the same thing. However, these assigned topics are just from the snippets. It is very likely that article 116 diverges into a discussion around foreign policy later on. With a topic similarity score of 0.999968941 that might very well be the case. It is however not something that will be discussed here.
7 Conclusion

This project has presented a Python3 implementation of extracting textual data from Swedish newspapers, identify and assign topics to those articles, as well as preforming sentiment analysis on the most similar article pairs based on their topics and day of publication. To extract the text from each article web scraping was used, using the WebDriver from the selenium library. The topic detection was preformed with the help of functions found within the scikit-learn library. To gain each article polarity and emotional state TextBlob was utilised and was then visualised with dumbbell plots.

Both goals, that was defined in Section 1.1, was achieved. I was able to extract textual data from 1 736 Swedish news-articles, 954 from Expressen and 782 from Aftonbladet, using Web scraping and identify topics using topic modeling, more specifically Non-negative Matrix Factorisation, for each article. A sentiment analysis between similar articles from both journals was done. The result of this was two plots, one for subjectivity and one for polarity, illustrates the differences between each article pair.

However, even if both goals was achieved the result was not reliable enough for me to make a concrete conclusion. This is because the preprocessing was missing a few parts, mainly lemmatisation and part-of-speech tags. Section 7.1 covers some desired improvements that could very well make the results more reliable. Without these parts I feel I can not draw any justifiable conclusions from the results.

7.1 Future Work

One of the most important part of topic modeling, and by extension this project, is preprocessing. The results would most likely be very much improved if proper Swedish word lemmatisation would have been implemented. Lemmatisation is a linguistic process of bringing words down to their basic form. Take the word "tables" and "computational". With lemmatisation "tables" becomes simply "table" and "computational" becomes "compute". This is very helpful since it removes "duplicates" from the textual data, e.g. prevents two words with the same meaning from appearing more then once.

During the preprocessing stage there is also a great idea to utilise a Part-of-Speech (PoS) tagger. PoS is a way of categorising every word with their corresponding grammatical properties, noun, adjective, subjective etc. During the sentiment analysis these tags should improve the result drastically.

This project utilised TextBlob because of its easy to use "out of the box'. However, since TextBlob simply translates each word much of the semantic meaning of every sentence may change. This could be avoided, and may therefore result in a different outcome, if a
Swedish dictionary was used. Most of the research done around PoS are in English, and the ones found that are in Swedish were harder than anticipated to implement during this project.

As mentioned in Section 4, NMF was utilised to identify the topics. However, the probabilistic method *Latent Dirichlet allocation* (LDA) was considered. LDA is rather more complex than NMF since it relies on multiple different parameters and probability distributions. Shown below is an illustration of the LDA model.

![Illustration of the LDA model](image)

**Figure 8:** Illustration of the LDA model

$M$ and $N$ are, in this case, the number of documents and the numbers of words in each document respectively. $z$ describes the topic of a specific word in a specific document. $w$ is the specific word and is also the only *observable variable*, meaning that it can be observed and measured in a statistical sense. All other variables are what’s called in statistics *latent variables*, which mean that they are not directly observed but are instead reliant and dependent on observed variables. The parameters $\alpha$, $\beta$, and $\theta$ are all of a more probabilistic nature and will therefore not be explained in detail here since they are more complex and out-of-scope for this project. In short however, $\alpha$ is the per-document topic distribution, $\beta$ is the per-topic term distribution, and $\theta$ is the topic distribution for a specific document. So LDA basically describes each document as a mixture of several topics that together describes a corpus or document.

NMF was however ultimately used instead of LDA, mainly because of it was easier to integrate with the already implemented system as well as time constraints. It was however noted that LDA should drastically improve the result if it would have been implemented correctly since it is a much more reliable method of identifying topics and is well used method for topic detection[1].

### 7.2 Limitations

This project is but a first analysis and contains many areas of improvements, as described in Section 7.1. Some limitations of the project would be that it only uses two different
journals. By using more journals, say three or four or even five, one would be able to gain a lot more data from many different sources. However, this would drastically increase the work needed to present that data in a meaningful way, i.e. would not be able to use the methods used in this projects or at least not keep it unchanged.

Another limitation was the number of gathered articles. 1736 articles are in fact not that many in this context. The timespan which those articles were gathered was also quite small, just seven days. By extracting a larger number of articles over a longer period of time, say around 10000 articles gathered during the span of a month, one would gain a drastically more diverse data set to analyse.

To further improve the result one could also implement more than one method of extracting and identifying topics. This project uses just one since it was not within the scope to compare different methods. It is however a limitation.

The calculations of the similarity index discussed in Section 4.2 have some limitations. The date similarity count with whole days. If it would use a 24 hour system instead it could return more varied result. The actual topic calculations could also be improved if the vectors $A$ and $B$ would contain more values than just four. These limitations may not limit the actual final result in its credibility but it does limit the precision of the result.

During the sentiment analysis stage, just a fraction of the total article pairs were picked for the analysis. This basically boils down into a limitation in time since these article pair had to be manually inspected of their credibility. However, with some of the improvement mentioned in Section 7.1 one could first verify the credibility of a small number of pairs and the assume that the same would be true for a larger number of pairs. The number of pairs were also based on how much would be able to presented in a reasonable manner. With a larger number of pairs some other type of graph would have been preferable to use.

7.3 Lessons Learned

I feel that I’ve learned a lot during the course of the project. Since the project stood on three main pillars, text extraction, topic detection, and sentiment analysis it gave me multiple ways to cover different aspect within topic modeling.

The method used to extract the articles from the journals was a rather fun dive into HTML code and structure, something that I did not know would have been the case. While looking into what newspapers I could base my project on I found that many cites have a similar structure, which was not really that surprising but I think it was nevertheless an interesting notation.
I was rather conflicted in the fact that I was not able to implement proper Swedish dictionary support within a reasonable timeframe. I knew it would effect my final result quite substantially but had to prioritise and move on with what I had. If I were to redo or further develop this project I would spend more time and write my code so that I could more easily implement some Swedish dictionary solution.

A large portion of the project was of course the actual topic detection and to be able to gain valuable information from it. I looked into several methods and way of thinking to accomplish this while doing my research before starting. However, in the end I found it way easier to just start working on one and then see what happens. This way I was able to move forward and also learned the strengths and weaknesses of each method as well as what suited my project best. I ended up with using NMF mostly because it had way fewer parameters to keep track of then LDA but also because of its roots in linear algebra, which instantly made it more comprehensible for me since its rooted in something familiar. Because of this, NMF enabled me be to progress in a reasonable pace.

All in all, I’ve learned a lot about the uses and implications around topic modeling but also the importance in planing ahead.
References


