Intermediate Report

Covid-19 News Analysis

Sam Sweere (i6231098), Alexander Reisach (i6197692)

KEN4135: Information Retrieval and Text Mining

Faculty of Science and Engineering
Dpt. of Data Science and Knowledge Engineering
Maastricht University
Netherlands

Contents

Chapter 1

Division of work

We are continually discussing how to approach certain aspects and regularly do joint code reviews. Table 1.1 gives a rough indication of who spend most time on which parts.

Sections	Alexander Reisach	Sam Sweere
Data Loading	X	
Processing Pipeline	X	X
Topics and Entities - Default Model	X	
Topics and Entities - Co-reference Resolution	X	
Topics and Entities - Computational Improvements		X
Sentiment Analysis - Training/Benchmarking		X
Sentiment Analysis - Sentiment Tracking		X
Visual Presentation	X	X

Table 1.1: Division of work

Chapter 2

Introduction

This project is a part of the course KEN4153 Text Mining and Information Retrieval. As such, we use the methods explained in said course to mine an extensive data set so to be able to automatically answer content related questions about actors, locations, events, and trends. Given the recent global developments, we use a large publicly available text corpus on Covid-19 news coverage as the basis for our analysis.

2.1 Data Set

Our data set is a pre-aggregated selection of 500,000 news articles on the Covid-19 global pandemic[1]. It covers articles published between November 1st 2019 and April 6th 2020. The data set comprises publications from 400 prominent news sources with an Alexa [2] rank below 5,000. It contains English language publications only. Moreover, the data set has been pre-processed and enriched by entities recognised, topical category tags, a sentiment measure and summaries. Source information and date of publishing are included as well. The data set contains the following keys:

{*'author', 'body', 'categories', 'characters_count', 'entities', 'hashtags', 'id', 'keywords', 'language', 'links', 'media', 'paragraphs_count', 'published_at', 'sentences_count', 'sentiment', 'social_shares_count', 'source', 'summary', 'title', 'words_count'*}

Usage of the data set is open for researchers, scientist and any other form of non-commercial analysis.

[1] `https://aylien.com/coronavirus-news-dataset/`
[2] `https://alexa.com/topsites/category/Top/News`

Chapter 3

Methodology

3.1 Data Loading

Our data set contains 500,000 news articles which take a total of 7.8 gigabytes of space. At this size we have to take computational and memory limitations into account. Therefore we implement a function to take a sample of the data based on publication dates. You can specify from and to which date you want to extract articles and how many articles you want from each specific day. At the current stage, we are working with samples from a sub-period only to allows us to test prototypes of our text mining pipeline more quickly.

3.2 Processing Pipeline

Before doing any text analysis we first have to do some pre-processing of the data. The data set already contains separate keys for headlines, body, authors, etc. This saves us a lot of time, since we do not have to text mine these articles ourselves. However, the text still consist out of plain text which we will have to structure to be able to run our data enrichment procedures. We use the Spacy [1] framework in Python since it is relatively easy to use, fast, and very versatile. By default, Spacy comes with a rudimentary pre-processing pipeline (figure 3.1). It includes pre-trained models but also gives us the flexibility to train models of our own and incorporate them into the pipeline.

[1] `https://spacy.io/`

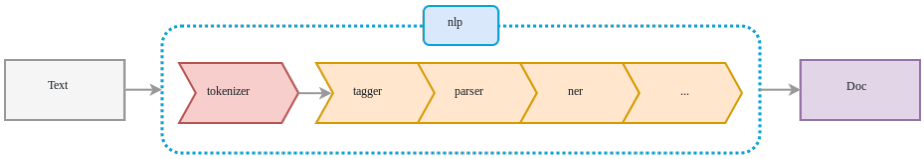

Figure 3.1: Spacy pipeline. (From: `spacy.io/usage/spacy-101`)

3.2.1 Tokenizer

Segment text into tokens. This segments the text into words, punctiation, etc. It does this by first splitting the text based on whitespaces, next on each substring it checks for two things:

- Does the substring match a tokenizer exception rule? Where for example some words do not contain whitespaces but should be split. Such as "It's" should become "It" and "'s".
- Can a prefix, suffix or infix be separated to become it's own token?

The tokenizer is only dependent on the language of the text corpus and will stay the same independent of what model is used for further processing.

3.2.2 Part-of-Speech Tagger

The Part-of-Speech (POS) tagger assigns its tags using the OntoNotes 5 [2] version of the Penn Treebank tag set.

3.2.3 Dependency Parser

The default dependency parser does within-sentence dependency parsing using a convolutional neural network based on [1] and inspired by [2].

3.2.4 Named Entity Recognition

Depending on the result of the previous steps, the named entity recognition assigns labels such as *Person* or *Location* to token it recognizes to be entities. Similar to the Parser, it is based on a convolutional neural network.

3.3 Topics and Entities

We approach the topics & entities by asking the question "What are the key entities concerned with Coronavirus and how do they change over time?"

[2] `https://catalog.ldc.upenn.edu/LDC2013T19`

3.3.1 Default model

We do this analysis by using the default small version of the Spacy English model to reduce computation time while the pipeline is still in development. For the final results, we will use a more extensive and computationally expensive English language model for greater accuracy.

Syntax Accuracy	
Labelled dependencies (LAS)	89.71
Unlabelled dependencies (UAS)	91.62
Part-of-speach tags (POS)	97.05

Named Entities Accuracy	
F-score	85.55
Precission	85.89
Recall	85.21

Table 3.1: Performance of the small default Spacy model.

The smaller Spacy model consist of a multi-task CNN trained on OntoNotes [3]. Similar to every Spacy pipeline it starts with tokinization, it then does POS tagging, dependency parsing and finally it detects and labels named entities. The performance of this model is shown in table 3.1.

Syntax Accuracy	
Labelled dependencies (LAS)	90.09
Unlabelled dependencies (UAS)	91.93
Part-of-speach tags (POS)	97.15

Named Entities Accuracy	
F-score	86.25
Precission	86.34
Recall	86.17

Table 3.2: Performance of the big default Spacy model.

The bigger Spacy model also consist of a multi-task CNN trained on OntoNotes [3] but with the GloVe vectors trained on Common Crawl. The performance of this model is shown in table 3.2.

3.3.2 Co-Reference Resolution

In order to extract more valuable patterns from our corpus, we will resolve within- and between-sentence references to identical entities. For example, in the sentence "*On March 26, Johnson revealed he had tested positive and that he had been dealing with symptoms since that date.*" we see the name *Johnson* is referred to again later in the sentence by *"he"*. From the context we know that both refer to politician *Boris Johnson*. We would like to count both of them as a reference to the corresponding real world entity. The default dependency parser usually does not connect such relationships correctly, see figure 3.2.

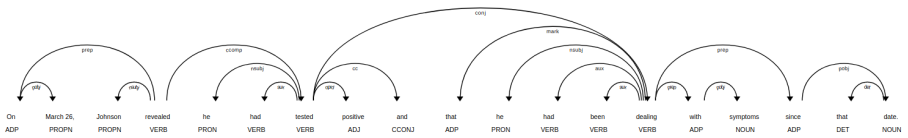

Figure 3.2: Spacy default dependency plot.

In order to resolve this issue, we first use an entity merger to ensure entities that consist of multiple words such as *Prime Minister Boris Johnson*, are treated as single entities. We extend the default model by using a neural network based co-reference resolution before estimating the most frequent entities. For this purpose we apply *NeuralCoref*[3], an open source implementation of the reinforcement-learning based approach by [4].

Neural Co-reference Resolution

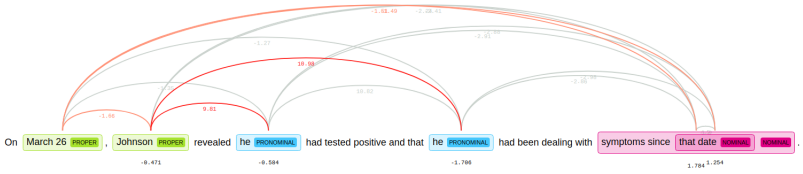

Figure 3.3: Co-reference resolution using NeuralCoref

We resolve the clusters detected by *NeuralCoref* using the cluster name for all variants of direct or indirect mentions of the same instance, and run the entity recognition module on the resolved text corpus.

At the current stage of the project, we are using document-specific clusters. It turns out that the entity recognition is often able to find the common parts even when identical clusters are assigned different names across documents, but we are still actively working on a more general solution. Moreover, our plan is to test if an extension of the current pipeline by an entity linking model that roots our entities in known real-world entities from a knowledge base will further improve our results.

3.4 Trends and Events

For all of these topics we will analyse how they develop over time to gain insights into trends and the events that may be shaping them. In terms of events, we expect to find for example the detection of the first cases outside China, inaction of curfews and state of emergency declarations by major countries.

[3] `https://github.com/huggingface/neuralcoref`

3.5 Sentiment Analysis

So to detect the cange of sentiment over time, we track the sentiment of the articles for each publishment date. We expect that it will start more neutral, become more negative and might become take a more positive towards the end. This could be correlated to specific trends and events to see what possibly changed the sentiment.

Sentiment analysis is a difficult process. For example in [5] the sentence "*This film does n't care about cleverness wit or any other kind of intelligent humor.*" is given as example. Most of the words in this sentence are positive, however the overall sentiment is negative since in the beginning of the sentence the word "*n't*" is present.

The Spacy NLP framework does not contain any sentiment analysis models. Therefore we train our own. We first train our model on the Large Movie Review Dataset [6]. This dataset contains 25000 movie reviews for training, and 25000 for testing. The movie reviews are or quite positive (rate ≥ 7) or negative (≤ 4) such that there is less ambiguity, the dataset is balanced and contains the same amount of both positive and negative reviews.

The dataset of movie reviews is quite different from our dataset of news articles. Therefore we are working on including the Stanford Sentiment Treebank dataset [5] in the training process. This dataset consist out of 215,000 phrases with fine-grained sentiment labels. We are also working to include a neutral sentiment to our model, since we expect that a part of the news articles are neutrally written.

The model we are fine tuning for sentiment analysis is the state of the art XLNET [7] model. This model has been found to score an accuracy of 96.21% on the Large Movie Review Dataset and 96.8% on the Stanford Sentiment Treebank dataset [5].

Chapter 4

Results

4.1 Topics and Entities

4.2 Trends and Events

4.3 Sentiment Analysis

Sentiment	Accuracy
F-score	92.3
Precission	93.6
Recall	91.0

Table 4.1: Sentiment Accuracy of the XLnet based model on the Large Movie Review Dataset.

The XLnet model is quite large and takes a lot of time to train. The best results we have obtained so-far are shown in table 4.1.

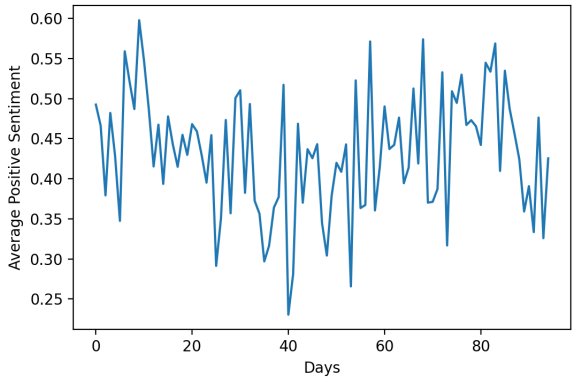

Figure 4.1: Average sentiment from 01-01-2020 until 06-04-2020. Average taken from 10 samples per day.

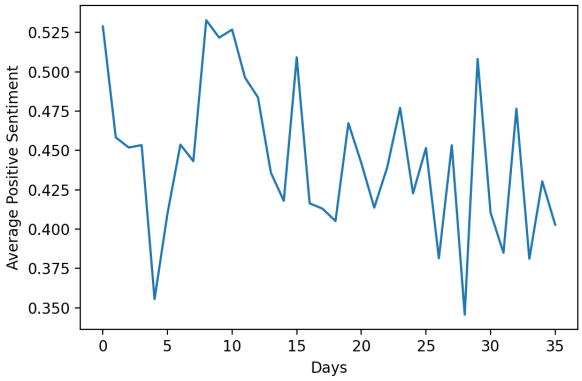

Figure 4.2: Average sentiment from 01-03-2020 until 06-04-2020. Average taken from 50 samples per day.

Some preliminary results can be seen in figure 4.1 and figure 4.2. We can see that the average sentiment is more negative then positive. Other trends are hard to determine from these results. We will run this model on more samples per day, to do this we will have to make the code compatible to run on the Aachen cluster. We expect that the results become more stable if we take the average sentiment of more samples per day.

4.4 Visual Presentation of Results

In order to make our trend analysis of the change of topics over time more accessible, we present them in form of a bar chart race. In this format, the prevalence and importance of topics is compared by their appearance in a bar chart which is updated at each time step. This results in a video capturing the rise and decline in importance of the most prominent Corona Virus news topics over time. At this stage of the project, we have a rough prototype of the concept working and are working actively on developing it further towards a bar chart race.

For an animated version of some preliminary results in the style of the screenshot in 4.3, see the attached *.gif* file. While many of the entities look like people that could likely be featured in news coverage, some of the entities seem rather out of place. We are still working on improving entity recognition and entity linking for this purpose and will need to increase the sample size massively for more meaningful results.

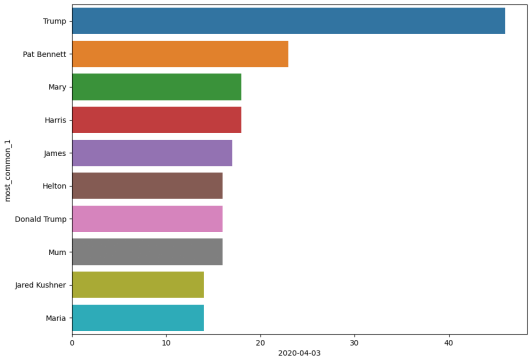

Figure 4.3: Top ten most prominent entities on a sample of 100 articles from 2020-04-03

If the scope of the project allows it, we would also like to present other aspects of our results in a way that is visually appealing and intuitive for a wider audience. This could include a time-animated world map of locations most associated with the Covid-19 health crisis, or a topic stream visualization of global trends and events. Furthermore, it could be interesting to combine our results with data on the spread of the epidemic measured in reported number of infections.

Chapter 5

Conclusion

5.1 Conclusion

5.2 Discussion

Covid-19 is often classified as a person. We hypothesize that this is the case because it is frequently used as the subject in sentences with an active verb such as *threatens*, or *prevents*.

Bibliography

[1] Y. Goldberg and J. Nivre, "A dynamic oracle for arc-eager dependency parsing," in *Proceedings of COLING 2012*, pp. 959–976, 2012.

[2] E. Kiperwasser and Y. Goldberg, "Simple and accurate dependency parsing using bidirectional lstm feature representations," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 313–327, 2016.

[3] e. a. Weischedel, Ralph, "Ontonotes release 5.0," *Philadelphia: Linguistic Data Consortium, 2013*, 2013.

[4] K. Clark and C. D. Manning, "Deep reinforcement learning for mention-ranking coreference models," *arXiv preprint arXiv:1609.08667*, 2016.

[5] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

[6] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (Portland, Oregon, USA), pp. 142–150, Association for Computational Linguistics, June 2011.

[7] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in neural information processing systems*, pp. 5754–5764, 2019.