KU LEUVEN

MASTER OF ARTIFICIAL INTELLIGENCE

The Neutralizer

Knowledge and the Web

Daudier Chloé

DeHaven Robbert

Luqman Elias

Tran Tuan-Anh

Click and type the subtext

January 2019

Contents

1 Introduction

It can be difficult to keep up with the news today. The abundance of news sources available means that we are rarely short of coverage of events. But how much of the news can we trust? How does the reader discern what is accurate and what is biased? How do they access a diverse enough range of views to know that they are well-informed on a particular subject? In short, is there a way of ‘neutralizing’ the news out there?

Our ‘Neutralizer’ aims to address the problem of biased news reporting by exposing the reader to a spectrum of views on a new subject, and informing them of the references for each article. Given a news subject and a set of articles in the subject, the Neutralizer presents the reader with a summary of each article. It highlight the statements from the articles that are topically similar, so that the reader is exposed to a variety of views on that topic. In addition to general summaries, the ‘Neutralizer’ also outputs fact check-worthy summaries, based on ClaimBuster’s filtering. [1] This output is especially useful for fact checkers and journalists. Finally, the Neutralizer also includes a reference mapper tool, which illustrates the relationship between the news articles and their references visually.

2 Motivation

‘Fake news’ is not a new phenomenon. Disinformation and propaganda have been used throughout history to influence people’s views. However, the reach of the internet today, and the pervasiveness of social media especially, provide an environment where news, accurate or otherwise, can be spread extremely quickly and cheaply. This obviously has huge consequences.

Various parties have introduced a number of methods to try and combat the problem of fake news. There are fact checking websites such as Politick, FactCheck.org and Snopes, which aim to fact check widespread claims (mostly politics-related). [2] [3] [4] Social media platforms such as Facebook have introduced improvements to their ranking algorithm and added tools that users can use to flag offending content. [5] Some governments have even enacted laws that punish parties and platforms complicit in spreading fake news. [6]

Our attempt at trying to help address the problem of fake news takes a slightly different approach. We aim to educate the reader by providing them with a spectrum of views on a particular news subject. We highlight the parts where there may be common or clashing perspectives on a specific topic. We also show the reader how the articles are linked to their references and possibly each other, graphically. We then leave it to the reader to make their own conclusions (or to continue their research on the news subject) based on the information provided. In short, the Neutralize provides tools to help the user inform themselves on a particular news subject, but does not attempt to decide what is accurate and what is not for them.

3 Architecture

The Neutralizer system consists of two main pipelines. The first pipeline generates the visual reference map and consists of the reference mapper tool. The second pipeline generates the article summaries, and is made up of multiple NLP components combined serially. The architecture is summarized in the sequence diagram below.

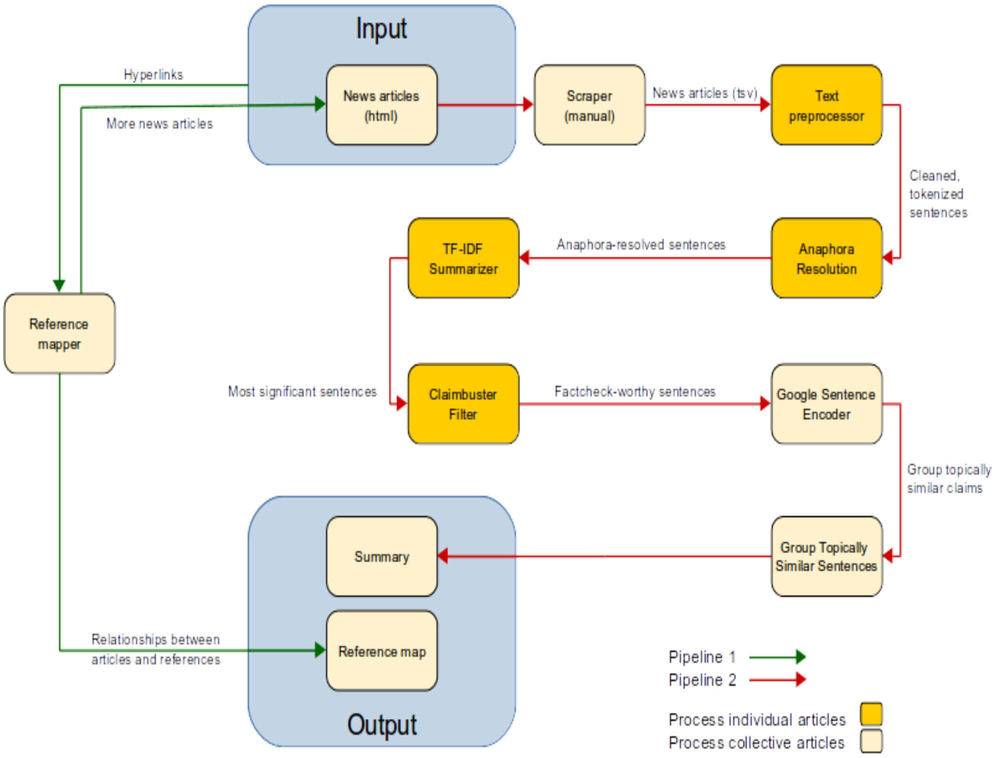

Input
Hyperlinks
More news articles
News articles (html)
Scraper (manual)
News articles (tsv)
Text preprocessor
Cleaned, tokenized sentences
Anaphora Resolution
Anaphora-resolved sentences
TF-IDF Summarizer
Reference mapper
Most significant sentences
Claimbuster Filter
Factcheck-worthy sentences
Google Sentence Encoder
Group topically similar claims
Group Topically Similar Sentences
Summary
Relationships between articles and references
Reference map
Output
Pipeline 1
Pipeline 2
Process individual articles
Process collective articles

Figure 1. Figure 1 High-level architecture of the Neutralizer

In pipeline 1, the reference mapper reads in the set of news articles (in HTML) in the given news subject and extracts all hyperlinks contained in the article. It then plots a graph of the hyperlinks. An example of the output of pipeline 1 looks like below.

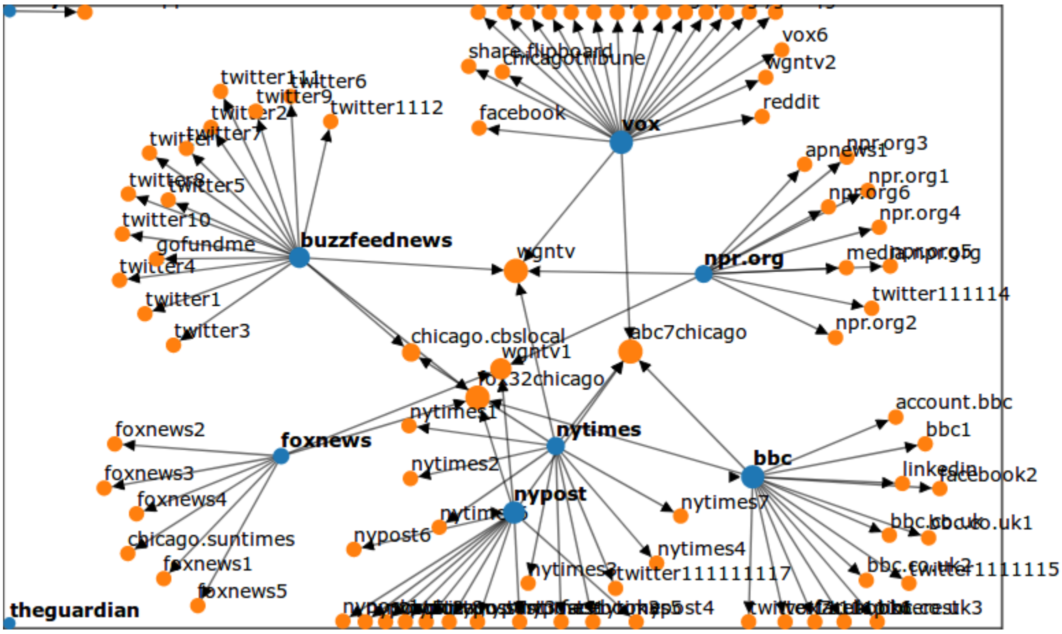


Figure 2 .Example output of the reference mapper. The blue nodes are articles in the dataset and the orange nodes are articles/webpages referenced by the blue nodes that are not in the dataset.

The basic functioning of pipeline 2 goes as follows. Once a set of news articles in a given subject has been identified, each article (in HTML) is put through the scraper, text preprocessor, anaphora resolver, TF-IDF summarizer and ClaimBuster API (the ClaimBuster API is an optional filter that can be turned on and off). The collection of summaries of the articles is then passed through the Google Sentence Encoder, which determines the sentences in the articles that are the most topically similar. A snippet of the output of pipeline 2 looks like below.

The Guardian summary	Daily Mail summary
A police officer fatally shot an armed security guard who witnesses say was trying to detain a man following a shooting at a suburban Chicago bar, authorities said.	A police officer fatally shot an armed security guard who was wearing a hat with "security" emblazoned across the front and holding a man down following a shooting inside the suburban Chicago bar where the guard worked, an attorney for the guard's family said Monday after filing a federal lawsuit.
Investigators said 26-year-old Jemel Roberson was pronounced dead shortly after being taken	Four others were shot and wounded, including a man who police believe fired a

to a hospital following the shooting early on Sunday at Manny's Blue Room in Robbins, just south of Chicago.	gun before police arrived, Cook County sheriff's spokeswoman Sophia Ansari said.
Four other people were shot and wounded during the incident, including a man who police believe fired a gun before police arrived, the Cook county sheriff's spokeswoman, Sophia Ansari, said.	Attorney Gregory Kulis filed a civil rights lawsuit seeking more than $1 million on behalf of Roberson's mother, Beatrice Roberson.
When police arrived at the scene, Roberson was holding "somebody on the ground with his knee in his back, with his gun in his back," witness Adam Harris told WGN-TV.	Kulis also echoed witness reports that Roberson was holding down another man outside the bar when the officer arrived and shot him.
Roberson was working to "enough money together for a deposit on a new apartment", said Hunter, the great uncle of Laquan McDonald, a black teenager fatally shot in 2014 by a white Chicago police officer.	Roberson was trying to "get enough money together for a deposit on a new apartment," said Hunter, the great uncle of Laquan McDonald, a black teenager fatally shot by a white Chicago police officer in a high-profile 2014 case.

Table 1 Example output of pipeline 2. Sentences highlighted in the same color are topically similar.

We will describe each component of the pipelines in more detail in section 5.

4 Data

In this section, we will describe our data sources and data preparation methods.

4.1 Data Sources

For our project it was essential to collect news articles from a range of different news outlets covering the same stories. We chose to focus on national news stories located in the USA as it has the largest amount of news outlets reporting in English. Furthermore, bias in the news seems to be a more pressing topic in the USA compared to the rest of the English speaking world. It has been the topic of several research papers and can be linked to the political bias of the news outlet. To capture the potential differences in reporting, we selected articles from all across the spectrum as described in [7], and can be split into three main groups: left, center and right leaning news outlets.

- Left: BuzzFeed News, Daily Kos, Huffington Post, Slate, The Guardian, The New Yorker, The Washington Post, Vox
- Center: ABC News, BBC News, Bloomberg, New York Times, NPR, The Hill
- Right: Breitbart, Daily Mail, Fox News, New York Post, The Washington Times

We handpicked approximately ten articles for each one of three controversial events to compare results. The events chosen where: “Accidental shooting of a black security guard by police”, “Recount of the Florida midterm election votes”, and “A shooting at a Synagogue in Pittsburgh”. These where relevant events when compiling the dataset (October 2018). The articles were scraped and stored in a .tsv file along with the subject, title, publication and URL of the articles.

4.2 Data Scraping

For pipeline 1, the reference mapper takes the HTML of the articles as input, so no scraping was needed. For pipeline 2, we scraped the raw text from the HTML of the articles manually.

4.3 Data Collection Limitations

Given the scope of this project, we decided not to prioritize the data gathering aspect of the Neutralizer and focused our efforts on other parts of the system that we thought had more value to demonstrate the Neutralizer as a proof-of-concept.

We collected ten articles for each news subject as we felt this number would be enough to provide a representative sample of views from across the political spectrum; we picked three news subjects simply because we thought this would be adequate for our immediate testing needs. There is plenty of room for further work related to data collection, as described in the section on further work.

5 Pipeline 1 (Reference Mapper) Components

In the section, we will describe the components of pipeline 1 in more detail.

5.1 References

5.1.1 Interest of a sources mapper

One of the first things we should verify when reading a newspaper article is the nature and value of its **sources**. Of course it depends on the subject of the article itself. A prince having a new crush, for instance, will not have the same pattern as a scientific paper: the sources do not have the same nature and reliability.

Furthermore, from a sample of articles (the initial input), it can be interesting find new articles to add to the input of the neutralizer. As some of them are linked together, looking for the sources can lead to new articles about the same subject. If so, it can be consistent to add new ones to the input of the neutralizer. On the other hand, if too many articles have the same source, they are probably redundant and we have to pay more attention to the manner they relay the information. Sometimes, articles can simply be a review of another article of a main newspaper.

Therefore, a good overview of those phenomenon’s can be really useful as a preparation before “neutralizing” the articles. To this end, we choose a mapping representation of the references of the articles.

5.1.2 Mapping representation

The mapping takes as input the URLs of different articles about the same subject. It returns a directed graph with the original articles in blue and the references in orange. The surface of the vertices is proportional to the number of edges that are connected (in or out).

See below for examples.

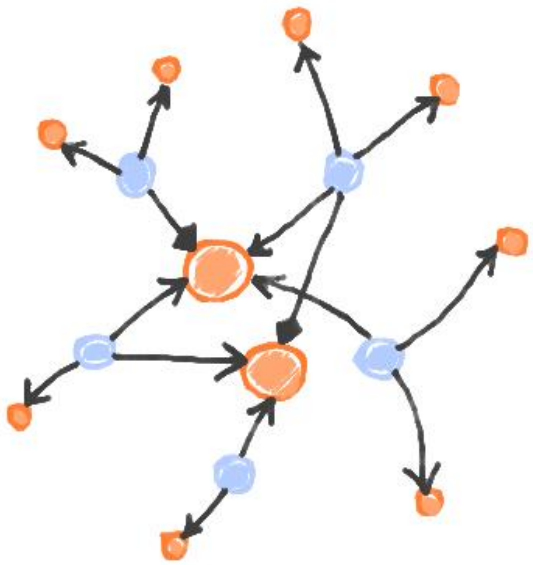
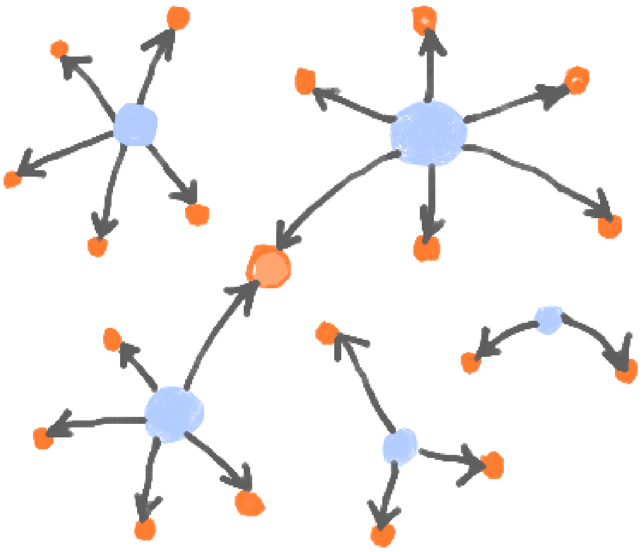
How to read source mapping?	
● corresponds to the original articles (input)	● corresponds to the sources/references
	
1- The input articles are linked to different sources. (a) <u>If the central sources are other newspaper articles</u>: if we add them to the neutralizer, it will probably lead to comparable sentences with and between the newspapers related to them. If so, *replacing the articles by the article-sources* can be considered. Beware, that implies that a mistake in the central source can be relayed in the “smaller” ones. (b) <u>If the central sources are social network posts</u>: it is probably a *human source to check*. (c) <u>If the central sources are scientific papers</u>: we assume articles referencing scientific papers are more likely to be trustworthy. Beware of bad understanding of the scientific lingo. In this case, it’s better to read the source directly.	**2- A lot of articles are isolated.** A comparison of the articles will probably lead to different points of view. There is probably no consensus about the subject OR the input articles are not well selected. It can be interesting to *add new articles to the input*.

Table 2 An explanation of how to read the source mapping

5.2 Newspaper and web libraries

A library named *newspaper* has been developed specifically to deal with web newspaper articles. [17] It is itself based on basic web libraries as *requests* and *urllib*. The next subsection will explain why we finally did not use the *newspaper* library.

5.2.1 Article extraction

The *newspaper* library provides an object `Article(url)` that can be downloaded (`article.download()`) and automatically parsed (`article.parse()`). Though, those functions are really expensive so it prolongs the execution of the algorithm, so we really have to be sure of their utility before using them. They can provide the following

- `article.authors` : not really optimal
- `article.publish_date` : depending on the newspaper, works very well or not at all
- `article.text` : needs to be improved but gives really good results that can be used for the neutralizer.
- `article.title` : perfectly works !

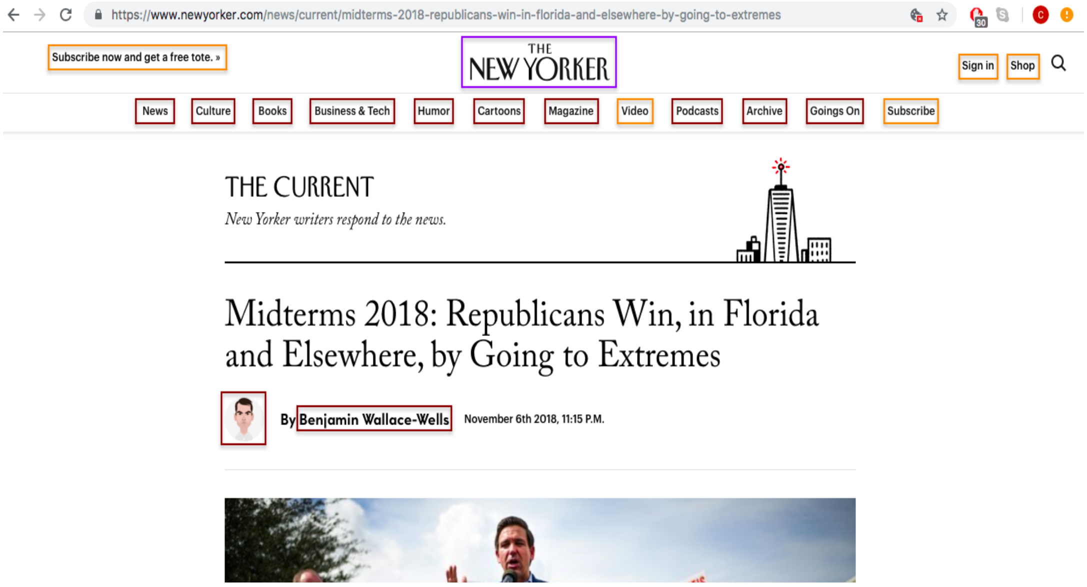
The tests of these functions were realized on the basis of our datasets: for each URL from the first subject, the results of the functions were compared to a manual check of the objects.

All those components can be really useful but they still unfortunately present a lot of mistakes and the time lost to call the functions is too big for the gain. That's why we finally extracted the articles ourselves.

5.2.2 Hyperlinks extraction

Now we have the articles, we must verify their sources.

There are different types of hyperlinks annotated in the HTML source code as such. Some of them are totally irrelevant.


https://www.newyorker.com/news/current/midterms-2018-republicans-win-in-florida-and-elsewhere-by-going-to-extremes
Subscribe now and get a free tote. »
THE NEW YORKER
Sign in
Shop
News
Culture
Books
Business & Tech
Humor
Cartoons
Magazine
Video
Podcasts
Archive
Goings On
Subscribe

THE CURRENT

New Yorker writers respond to the news.

Midterms 2018: Republicans Win, in Florida and Elsewhere, by Going to Extremes

By Benjamin Wallace-Wells November 6th 2018, 11:15 P.M.

Figure 3 Example of irrelevant hyperlinks found in a newspaper webpage.

Therefore, it is really important to sort the hyperlinks to have a clean reference map.

For this step we had to use empirical tips…

1) Retrieve all the hyperlinks of the HTML page of a given URL using xpath
   ⇒ obtain_fullinks(url)
   → ex : 125 href
2) Keep only "real hyperlinks", the links to other URLs
   ⇒ only keep the ones that start with 'http' (= contain 'http', of course it can also be https...)
   → ex : 49 href
3) Subtract the hyperlinks of the given URL with the hyperlinks of the home page of the newspaper : removes all the structural hyperlinks left ("contact us", "Sport" category etc.)
   ⇒ `urlparse(url).netloc` (from *urllib*) is used to isolate the url of the main page of the newspaper.
   → ex : 45 href

4) Take off most of the false references. For example, at this point, “`http://video.newyorker.com`” is still in the list and does not refer to another article. We suppose that real articles (sources) will contain a title into the last part of their URL and we suppose that the length of this title is at least 25 characters (empirical - we try to optimize sensitivity and not specificity, so it doesn’t matter if we still have false positive hyperlinks).
   ⇒ select the links according to their size
   → ex : 23 href
5) Take off all the “share” links. Removing the popular domains can eliminate social media sources like tweets, so we choose another method.
   ⇒ remove the links that contain exactly the name of the article itself. If so, that means that it is a hyperlink directly related to the article and more an “incoming reference” than an “outgoing reference”.

This method can be improved but is sufficient to remove most of the useless hyperlinks. For our purpose it is not necessary to be strict: the main sources will be highlighted and the others set aside. For this, a dynamic representation of the map would be a plus (with JavaScript it is possible to integrate gravity and attraction and this can allow a coherent visual distribution).

5.3 Graphic representation

That is what led us to the *D3js* library (https://d3js.org/).

The graph templates of this library takes “links” and “nodes” as input and can be configured to show a dynamic and interactive graph.

The attributes of the nodes are:

- id : unique and relatively short identifier of the source
- url
- radius : arbitrarily given, will be replaced by the calculated radius of the printed node

- group: 1 if the node is one of the given article, 2 if it is a hyperlink to another article.

The attributes of the links are:

- source : one of the given urls
- target : one of the hyperlinks of this URL
- radius

The webpage is made of HTML + CSS +JS.

6 Pipeline 2 (Article Summarization) Components

In the section, we will describe the components of pipeline 2 in more detail.

6.1 Anaphora Resolution

Since the news articles in our dataset will be summarized, it is possible that sentences will be taken out of context. The context in this case can be a coreference or anaphora. There is a slight difference between the two but for our purposes they can be seen as the same. A coreference occurs when two or more expressions in a text refer to the same person or thing.

An example would be: "I have a dog named Jules. He likes to go on walks."

Here "Jules" and "He" are considered anaphora. If we were to feed this text fragment into our program, and "He likes to go on walks" would be selected as a part of the summary but not the first sentence, a great deal of information would be lost. This could be only a minor inconvenience or make the summary unintelligible. To alleviate this issue, we implemented a coreference resolution framework called Neuralcoref[1]. This is an open source extension for spaCy[2] (an NLP toolkit) that is being developed by huggingface, a company specializing in creating chatbot AI. There are other options available for implementing coreference resolution, such as the Stanford CoreNLP suite (Java). Neuralcoref's extension was preferred as it offered an implementation in python, while being based on the same papers and theory as the CoreNLP suite.

The resolution is achieved by using a combination of word vectors and neural nets. First, all mentions are extracted from the text. Second, all mentions and words surrounding the mentions receive word embeddings, using word2vec. This captures the context of the mention. The third step feeds this information into 2 neural nets. The first net scores each pair of mention and a possible antecedent. The second net scores the possibility of the mention not having an antecedent. The highest score is chosen as the connection (or lack thereof).

[1] https://github.com/huggingface/neuralcoref

[2] https://spacy.io/

Using this method, mentions can be grouped together and the most informative is selected to replace the others. In our earlier example a grouping would be ["Jules", "He"], and "He" would be replaced by "Jules".

The Neuralcoref extension is not very tweakable as it only allows you to change the size of the pretrained model. Even when using the largest model the results were quite disappointing. Often mistakes were made in classifying coreferences that resulted in confusing articles. Another problem that arises concerned the incorrect handling of contractions: "she's" would not be replaced with, for example, "The suspect is" but with "The suspect", which reduced readability. However this could be prevented with some text preprocessing.

In the end, we decided not to implement anaphora resolution into our pipeline as the results without it were, for the most part, interpretable. The authors of [19] find the same to be true, they attribute the lack of improved readability to the low performance of the anaphora resolver.

6.2 Summarizer

We looked at two common extractive summarization approaches in order to find the one that best suited our needs. One is based on TF-IDF and the other on TextRank.

6.2.1 TF-IDF Summarizer

An extractive TF-IDF summarizer ranks sentences in a document according to some function of the TF-IDF scores of the terms in the sentences. The most basic ranking function is calculated by simply summing the TF-IDF score of each term in a sentence. The TF-IDF score for each term reflects how important or relevant the term is to the document (compared to a background corpus), so similarly, one can think of the sum of these scores for a sentence as representing the significance of the sentence in the context of the document. The top N ranked sentences are then chosen for the summary, based on a desired N value or a ranking threshold. For this project, we generated summaries with N values of 5 and 10. The top-5 summaries strike a good balance between being concise while still being fair representations of the articles. From a UI point of view these summaries also fit well in our output comparison tables. The top-10 summaries were generated to help us experiment with ClaimBuster filtering (we were not sure how many sentences would remain in the summaries after filtering) and to provide the user more information on news subjects that require it.

The TF-IDF summarizer used in the Neutralizer includes the following optimizations on top of the basic summation function:

- Stop word filtering
- Only use the TF-IDF scores of nouns and verbs
- Give a bonus score to sentences related to the title

Stop word filtering is a common feature of summarization systems. The TF-IDF scores for stop words are small, so their inclusion would not normally have much of an effect. However, a sentence with many stop words could potentially cause problems, so outright filtering is a prudent step to take.

The second optimization is inspired by Seki. [8] Seki only scored nouns in his approach, but we experimented with scoring verbs as well. The idea behind only scoring nouns and verbs is that these are the parts of speech that carry the most substance. Our testing showed that there was not a big difference in the sentence rankings between scoring only nouns and scoring both nouns and verbs. This could be because in the noun-only scoring scheme, verbs with subjects and objects (which are nouns), are already accounted for. Nonetheless, scoring both parts of speech seems theoretically sound, so that is what we did.

The third optimization is also inspired by Seki. We add a small score bonus to sentences with words that occur in the title (exact matching). [8] [18] the intuition is that this biases sentences that are related to the title, making for more focused summaries.

There are some commonly used optimizations that we considered implementing, but did not include in the final summarizer. [18] The most notable ones include:

- Normalizing scores by sentence length
- Stemming or lemmatizing terms
- Applying positional weights

We experimented with normalizing the sentence scores by the sentence length and logarithm of the sentence length, but both of these methods appeared to overly favor shorter sentences and penalize longer ones. We also considered stemming or lemmatizing terms and applying positional weights, but decided not to experiment with these optimizations given time constraints and the fact that we were already getting reasonable results.

We generated two versions of each summary, one with the top 5 ranked sentences and another with the top 10 ranked sentences. The sentences in both versions were ordered by the order they occurred in the document. For the summarizer's implementation, we used scikit-learn to calculate the TF-IDF term scores and NLTK to filter stop words and tag the parts of speech. We used a subset of NLTK's Reuters corpus for the background corpus used in the IDF calculations. This subset is the union of the Reuters train and test data, with the following news categories filtered out: barley, cotton, cotton-oil, earn, meal-feed, oat, rice, sorghum, soy-meal, soy-oil.

6.2.2 TextRank Summarizer

TextRank is a text summarization algorithm inspired by Google's well-known PageRank algorithm. [9] The PageRank algorithm "works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites." [15] The TextRank algorithm is fundamentally similar to PageRank, with a few adjustments:

- Sentences are used in place of web pages
- The similarity score between two sentences is equivalent to the transition probability between two web pages
- The similarity scores are stored in a square matrix, similar to the matrix M used for PageRank

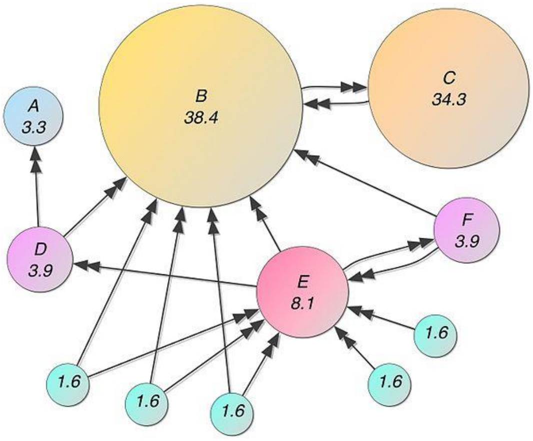


Figure 4 An example of a PageRank graph

Note that the new graph built for TextRank is undirected, but PageRank could be always applied on it. According to the authors, “Although traditionally applied on directed graphs, a

Recursive graph-based ranking algorithm can be also applied to undirected graphs, in which case the out degree of a vertex is equal to the in-degree of the vertex”.

The steps below describe the flow of the TextRank algorithm:

- Preprocess text (e.g. remove stop words, perform sentence tokenization etc.)
- Vectorize sentences by Glove.
- Create a similarity matrix by computing similarity scores between all sentences by cosine similarity. Our implement is a bit different with the original strategy in the article. According to the author, the similarity is measured by the overlap of two sentences which can be determined simply as the number of common tokens between the lexical representations of the two sentences.
- Create a graph with sentences as vertices
- Connect all vertices to each other with an edge, using the similarity score between sentences as the weight of the edge between them
- Run the PageRank algorithm on the graph
- Rank the vertices (sentences) by their PageRank scores

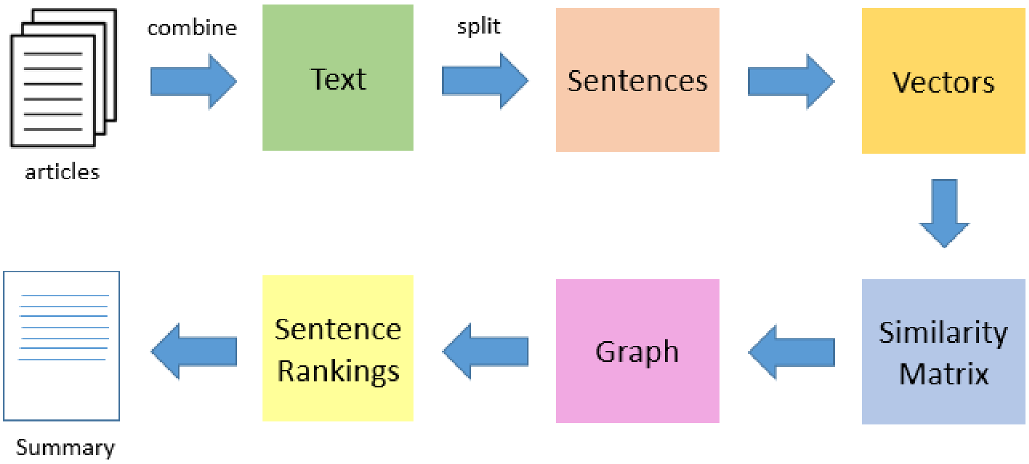


Figure 5 TextRank flow

For our implementation of TextRank, we used the pytexrank package with its default settings. [10]

6.2.3 Summarizer Comparison

In our testing, we found that TF-IDF summarizer consistently outperformed the TextRank summarizer. This evaluation was done by manually reviewing the top-5 summaries generated by both methods for a majority of the articles in our dataset. In many cases, the top two or three ranked sentences were similar between both methods, but TextRank's ranking below this was unreliable. The TF-IDF summaries were in general well-focused, which was likely partly down to the score boost given to sentences related to the title. Below are example summaries generated by the TF-IDF and TextRank summarizers on the same news article. [11]

Sentence	TF-IDF summary	TextRank summary
1	An armed security guard at a bar in suburban Chicago was killed by police as he detained a suspected gunman, according to officials and witnesses.	An armed security guard at a bar in suburban Chicago was killed by police as he detained a suspected gunman, according to officials and witnesses.
2	After gunfire erupted around 04:00 local time on Sunday, Jemel Roberson, 26, chased down an attacker and knelt on his back until police arrived.	Moments after police came on the scene, an officer opened fire on Roberson, who was black, killing him.
3	Moments after police came on the scene, an officer opened fire on Roberson, who was black, killing him.	Friends say Roberson was a musician who had dreams of joining the police.
4	Sophia Ansari, a spokeswoman for the Cook County Sheriff's office, said police were called to the scene after a fight broke out in the bar and four people were shot.	"Just waiting on the police to get there."
5	"How in the world does the security guard get shot by police?"	Illinois State Police's Public Integrity Taskforce have been asked to investigate the shooting, Ms Ansari told the BBC.

Table 3 A comparison of summaries generated by the TF-IDF and TextRank summarizers

6.3 ClaimBuster Filter

After the article has gone through the summarizer, it is useful for us to remove sentences that are not factual claims that made it into the summary. This can be done using the claim spotter module of ClaimBuster [16]. ClaimBuster is a tool developed specially to aid human fact checkers with their work. There is a free API endpoint[3] available. It can receive a text and score the sentences individually. The score ranges from 0 to 1, the higher the score, the more likely the sentence is to have check-worthy factual claims. Lower scores likely contain non-factual, subjective and opinionated elements. There are several ways in which this score can be used to remove lower scoring sentences. For instance, an absolute or relative number of the top sentences can be retained. However, for our purposes, we decided to hold all articles to the same standard and use a hard cutoff for being cut from the article. Through trial and error we determined 0.3 to be an acceptable score. A disadvantage is that not all summaries will have the same amount of sentences. Below are some examples of sentences that were redacted from the summary.

> *Here is what we know so far about the attack, the suspect and the victims.*
>
> *As news of the recount broke, Trump weighed in on Twitter.*
>
> *Jonathan Greenblatt, chief executive officer of the Anti-Defamation League, said the group believes Saturday's attack was the deadliest on the Jewish community in US history.*
>
> *Reports said Cecil Rosenthal liked to greet people at the door of the synagogue before services.*

These sentences all contain unique words that are more common in the article (and thus receive a high TF-IDF score). But they also contain non-factual elements that do not add much to the summary, as can be seen in the first two sentences. The last two sentences contain words as "like" and "believed" which lead the claim spotter to believe these are subjective or opinionated sentences, and can also be removed.

[3] https://idir-server2.uta.edu:443/factchecker/score_text/

6.4 Identifying Topically Similar Sentences

Originally, our goal was to identify sentences in different summaries that agree with or contradict each other (i.e. stance detection). However, this task proved to be more difficult than we anticipated. The methods we investigated for this goal are semantic similarity methods, but they did not turn out to be sufficient for our stance detection needs, and so were adapted for the simpler goal of identifying topically similar sentences. The methods we looked at were Smooth Inverse Frequency and Google's Universal Sentence Encoder.

6.4.1 Smooth Inverse Frequency (SIF)

Smooth Inverse Frequency is a semantic similarity method that is based on word embeddings. The word embeddings of all words in a sentence are obtained, and a weighted average of these embeddings is taken. A distance function is used to calculate the distance between the resulting embeddings.

The weighted averages help remove potential bias from words that are semantically irrelevant. The word embeddings are weighted by a/(a + p(w)), where a is a parameter that is normally set to 0.001 and p(w) is the estimated frequency of the word calculated from a reference corpus. Additionally, SIF "removes variation related to frequency and syntax that is less relevant semantically" by "computing the principal component of the resulting embeddings for a set of sentences" and then" subtracting from these sentence embeddings their projections on their first principal component". [13][14]

We implemented SIF using the Scikit-learn and Gensim toolkits. We use Word2Vec for the word embeddings and cosine similarity as the distance function.

6.4.2 Google Universal Sentence Encoder (USE)

Google's Universal Sentence Encoder is a pre-trained model that encodes text into high dimensional vectors and can be used for NLP tasks such as semantic similarity and text classification. The model is trained with a deep averaging network (DAN) and is optimized for "greater-than-word length text, such as sentences, phrases, or short paragraphs". [12] Its advantages over less complex semantic similarity methods such as SIF include the fact that it is a supervised model, it takes word order into account and it models embeddings beyond the level of individual words.

The table below shows the results that we obtained on a sample test set. As the fourth example illustrates, sentences that are topically close but disagree on stance obtain high similarity scores, ruling out the use of the model as a stance detection tool in the Neutralizer.

Sentence 1	**Sentence 2**	**Semantic similarity score**
Let’s go home	Let’s go home	1.00
Should we get going	Should we go	0.81
I like strawberries	I do like strawberries	0.97
I do like strawberries	**I do not like strawberries**	**0.95**
I like strawberries	Let’s have strawberries	0.83

Table 4 The highlighted example shows why Google’s USE cannot be used for stance detection

One implementation detail to note is that the USE model is quite big (1GB). It takes some time for the model to load the first time it is instantiated, but caching the model speeds up loading on subsequent use.

6.4.3 SIF and Google USE Comparison

We tested both methods and found Google’s USE to return better results than SIF. The USE’s recall was a lot better for sentences that were similar semantically, but dissimilar in vocabulary and construction (precision wasn’t negatively affected). For sentences that were similar semantically and also in vocabulary and construction, both methods returned decent results. Likewise, for sentences that were dissimilar semantically, results were comparable between the methods. The table below illustrates primary pattern that we observed.

Sentence 1	**Sentence 2**	**SIF similarity score**	**USE similarity score**
Roberson played music at the Purposed Church in Chicago for the past several years, according to Pastor LeAundre Hill, who tweeted that Roberson "had just played for my grandma's funeral Friday and now he's gone."	Roberson was working to "enough money together for a deposit on a new apartment", said Hunter, the great uncle of Laquan McDonald, a black teenager fatally shot in 2014 by a white Chicago police officer.	0.26	0.43
Four other people were shot and wounded during the incident, including a man who police believe fired a gun before police arrived, the Cook county sheriff's spokeswoman, Sophia Ansari, said.	Four others were shot and wounded, including a man who police believe fired a gun before police arrived, Cook County sheriff's spokeswoman Sophia Ansari said.	0.89	0.98
A police officer fatally shot an armed security guard who witnesses say was trying to detain a man following a shooting at a suburban Chicago bar, authorities said.	An armed security guard at a bar in suburban Chicago was killed by police as he detained a suspected gunman, according to officials and witnesses.	**0.58**	0.94

Table 5 In the top row, both methods score the semantically dissimilar sentences lowly, as expected. In the second row, the sentences are semantically similar and also superficially similar. Again, scoring was accurate for both methods. In the third row, SIF

Our evaluation was done manually and on a relatively small test set. That said, the USE is theoretically the more powerful method, so the results from our (limited) testing were not surprising. Based on the results observed, we were quite confident in the encoder’s performance and decided to employ it in the final Neutralizer pipeline.

6.5 Grouping Topically Similar Sentences

A	1				
B	0.84	1			
C	0.53	0.63	1		
D	0.28	0.37	0.60	1	
E	0.23	0.35	0.70	0.69	1
	A	B	C	D	E

Table 6 Similarity matrix

From the similarity matrix of all top-ranking sentences from different articles in the previous work, we build a distance matrix to show the distance between sentences:

A	0				
B	0.16	0			
C	0.47	0.37	0		
D	0.72	0.57	0.40	0	
E	0.77	0.65	0.30	0.31	0
	A	B	C	D	E

Table 7 Distance matrix

This distance matrix is symmetric and diagonal elements are zero representing the distance from a sentence to itself. Based on this matrix, we have some options to cluster the sentences. The first one is K-medoids, this is a robust algorithm. However, we need to declare k - the number of group - before clustering. Thus, this method is not suitable for our problem.

The second method we considered is Hierarchical Clustering which seeks to build a hierarchy of clusters. Then, by using a threshold, we could get groups of sentences which are similar to each other. Thus, this approach is suitable and easy to modify the result by changing the value of the threshold.

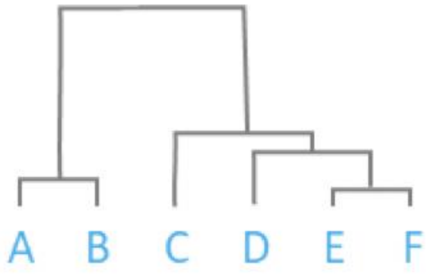


Figure 6 Cluster Dendrogram

We settled on a sentence similarity threshold of 0.9 through trial and error. We found this value to give the most accurate results across the three news subjects tested.

6.6 Visualization

To visualize our output we opted for a simple HTML page that displays the summaries of all processed data sources. The clusters are indicated by using a combination on CSS and JavaScript to highlight the individual clusters over all summaries in unique colors per cluster.

7 Results

Our result files have been output to the following location in the project package: Neutralizer/Website/cluster*/*.html

7.1 Quality of Summaries

Summary evaluation is a fairly subjective exercise. From our manual evaluation, we found that the TF-IDF summarizer summarized the news articles in our dataset reasonably well. A lot of the important information was extracted, and the summaries were generally quite readable. We compared a handful of the summaries against those generated by http://textsummarization.net/, and our summaries compared well.

One bug that we found is sentence fragments making it into the summaries. These were mostly down to sentence tokenization issues with quotes and words containing periods. We relied on NLTK's sentence tokenizer along with some basic text preprocessing to handle sentence tokenization. A more sophisticated tokenization approach may likely solve this issue.

7.2 ClaimBuster Filtering Results

The table below shows the average number of sentences filtered out by ClaimBuster per summary. The number is small for both top-5 and top-10 summaries, across different news subjects. This suggests that the significant majority of the sentences extracted by the summarizer are fact check-worthy.

News subject	**Average number of sentences filtered out by ClaimBuster per summary**	
	Top-5 summaries	**Top-10 summaries**
Accidental shooting	0.33	1.44
Florida midterm elections	0.00	1.11
Synagogue shooting	0.25	1.28

Table 8 Number of sentences filtered out by ClaimBuster

7.3 Similarity Score Thresholds

We experimented using different Google USE similarity score thresholds for grouping topically similar sentences. Through trial and error, we found a threshold of 0.9 to be the most accurate for the dataset that we collected. When we lowered the threshold to increase the number of clusters and cluster sizes, we introduced false positives, as demonstrated below.

The table below shows two top-5 summaries in the same news subject. The sentences highlighted in the same color are topically similar. We see that with a threshold of 0.9, recall and precision are good.

The Guardian	**Daily Mail**
A police officer fatally shot an armed security guard who witnesses say was trying to detain a man following a shooting at a suburban Chicago bar, authorities said.	A police officer fatally shot an armed security guard who was wearing a hat with "security" emblazoned across the front and holding a man down following a shooting inside the suburban Chicago bar where the guard worked, an attorney for the guard's family said Monday after filing a federal lawsuit.

Investigators said 26-year-old Jemel Roberson was pronounced dead shortly after being taken to a hospital following the shooting early on Sunday at Manny's Blue Room in Robbins, just south of Chicago.	Four others were shot and wounded, including a man who police believe fired a gun before police arrived, Cook County sheriff's spokeswoman Sophia Ansari said.
Four other people were shot and wounded during the incident, including a man who police believe fired a gun before police arrived, the Cook county sheriff's spokeswoman, Sophia Ansari, said.	Attorney Gregory Kulis filed a civil rights lawsuit seeking more than $1 million on behalf of Roberson's mother, Beatrice Roberson.
When police arrived at the scene, Roberson was holding "somebody on the ground with his knee in his back, with his gun in his back," witness Adam Harris told WGN-TV.	Kulis also echoed witness reports that Roberson was holding down another man outside the bar when the officer arrived and shot him.
Roberson was working to "enough money together for a deposit on a new apartment", said Hunter, the great uncle of Laquan McDonald, a black teenager fatally shot in 2014 by a white Chicago police officer.	Roberson was trying to "get enough money together for a deposit on a new apartment," said Hunter, the great uncle of Laquan McDonald, a black teenager fatally shot by a white Chicago police officer in a high-profile 2014 case.

Table 9 Top-5 summaries (pre-ClaimBuster filtering) from The Guardian and the Daily Mail on 'Accidental Shooting'. Similarity threshold of 0.9 is used.

The same summaries are presented below, this time with a similarity threshold of 0.85. In each summary we see multiple sentences which are topically similar, something not seen in the previous example. This would suggest that the summaries contains repeated information, which is not the case. The lower threshold has affected precision negatively.

The Guardian	Daily Mail
A police officer fatally shot an armed security guard who witnesses say was trying to detain a man following a shooting at a suburban Chicago bar, authorities said.	A police officer fatally shot an armed security guard who was wearing a hat with "security" emblazoned across the front and holding a man down following a shooting inside the suburban Chicago bar where the guard worked, an attorney for the guard's family said Monday after filing a federal lawsuit.
Investigators said 26-year-old Jemel Roberson was pronounced dead shortly after being taken to	Four others were shot and wounded, including a man who police believe fired a gun before police

a hospital following the shooting early on Sunday at Manny's Blue Room in Robbins, just south of Chicago.	arrived, Cook County sheriff's spokeswoman Sophia Ansari said.
Four other people were shot and wounded during the incident, including a man who police believe fired a gun before police arrived, the Cook county sheriff's spokeswoman, Sophia Ansari, said.	Attorney Gregory Kulis filed a civil rights lawsuit seeking more than $1 million on behalf of Roberson's mother, Beatrice Roberson.
When police arrived at the scene, Roberson was holding "somebody on the ground with his knee in his back, with his gun in his back," witness Adam Harris told WGN-TV.	Kulis also echoed witness reports that Roberson was holding down another man outside the bar when the officer arrived and shot him.
Roberson was working to "enough money together for a deposit on a new apartment", said Hunter, the great uncle of Laquan McDonald, a black teenager fatally shot in 2014 by a white Chicago police officer.	Roberson was trying to "get enough money together for a deposit on a new apartment," said Hunter, the great uncle of Laquan McDonald, a black teenager fatally shot by a white Chicago police officer in a high-profile 2014 case.

Table 10 . Top-5 summaries (pre-ClaimBuster filtering) from The Guardian and the Daily Mail on 'Accidental Shooting'. Similarity threshold of 0.85 is used. False positives are introduced.

When a similarity threshold of 0.8 is used, there is only one topical cluster, as seen below.

The Guardian	**Daily Mail**
A police officer fatally shot an armed security guard who witnesses say was trying to detain a man following a shooting at a suburban Chicago bar, authorities said.	A police officer fatally shot an armed security guard who was wearing a hat with "security" emblazoned across the front and holding a man down following a shooting inside the suburban Chicago bar where the guard worked, an attorney for the guard's family said Monday after filing a federal lawsuit.
Investigators said 26-year-old Jemel Roberson was pronounced dead shortly after being taken to a hospital following the shooting early on Sunday at Manny's Blue Room in Robbins, just south of Chicago.	Four others were shot and wounded, including a man who police believe fired a gun before police arrived, Cook County sheriff's spokeswoman Sophia Ansari said.
Four other people were shot and wounded during the incident, including a man who police believe	Attorney Gregory Kulis filed a civil rights lawsuit seeking more than $1 million on behalf of

fired a gun before police arrived, the Cook county sheriff's spokeswoman, Sophia Ansari, said.	Roberson's mother, Beatrice Roberson.
When police arrived at the scene, Roberson was holding "somebody on the ground with his knee in his back, with his gun in his back," witness Adam Harris told WGN-TV.	Kulis also echoed witness reports that Roberson was holding down another man outside the bar when the officer arrived and shot him.
Roberson was working to "enough money together for a deposit on a new apartment", said Hunter, the great uncle of Laquan McDonald, a black teenager fatally shot in 2014 by a white Chicago police officer.	Roberson was trying to "get enough money together for a deposit on a new apartment," said Hunter, the great uncle of Laquan McDonald, a black teenager fatally shot by a white Chicago police officer in a high-profile 2014 case.

Table 11 . Top-5 summaries (pre-ClaimBuster filtering) from The Guardian and the Daily Mail on 'Accidental Shooting'. Similarity threshold of 0.8 is used. A single topical cluster is found.

The table below shows the detailed results of topical grouping results using different similarity thresholds.

News subject	**Similarity threshold**	**Number of clusters (cluster sizes)**			
		Top-5 summaries	**Top-5 summaries (with ClaimBuster filtering)**	**Top-10 summaries**	**Top-10 summaries (with ClaimBuster filtering)**
Accidental shooting	0.90	3 (2,2,5)	3 (2,2,5)	8 (2,2,2,2,2,3,4,5)	8 (2,2,2,2,2,3,4,5,)
	0.85	3 (2,2,14)	3 (2,2,13)	8 (2,2,2,3,3,3,4,22)	7 (2,2,3,3,3,4)
	0.80	1 (32)	1 (31)	2 (3, 59)	2 (3, 55)
Florida midterm elections	0.90	1 (3)	1 (3)	2 (2,3)	2 (2,3)
	0.85	4 (2,4,6,11)	4 (2,4,6,11)	5 (2,2,2,6,21)	5 (2,2,2,6,21)
	0.80	2 (2,32)	2 (2,32)	2(5,56)	2 (2,50)

Synagogue shooting	0.90	0	0	0	1 (2)
	0.85	3 (2,2,2)	3 (2,2,2)	5 (2,2,2,3,3)	5 (2,2,2,3,3)
	0.80	2 (2,25)	3 (2,2,18)	5 (2,2,2,3,40)	3 (2,2,28)

Table 12 Number and size of clusters based on different similarity thresholds

It is interesting to see how the larger cluster mainly consist of the first sentences in the summaries, this could be expected as the first sentences are likely to contain the most important facts that are known about the subject. In the following sentences influence of the journalist becomes more apparent as he makes decisions on the angle and extra facts that will be included in the article.

These were also the clusters that for the most part remained when we distilled the summaries from the top ten sentences to the top 5. Many of the clusters consist of direct quotes, this was expected as the sentences are nearly identical.

8 Conclusion

We were pleasantly surprised by the Neutralizer's performance. With parameters tweaked by trial and error, the neutralized summaries and reference maps are both reasonably accurate and user-friendly. From the results provided in the project package and summarized above, we believe that we have demonstrated its potential as a tool to help users inform themselves on the news subjects in our dataset.

That said, our dataset is very limited. It would be interesting to see how well the Neutralizer performs on a larger dataset. It may even be that a larger dataset would allow the Neutralizer to showcase its abilities more. For example, we did not find topically similar sentences which contradict each other in our dataset, which theoretically should be identifiable by the Neutralizer, and would have been especially interesting from a fact checking point of view.

In fact, we found a surprising level of consistency between the different news sources from across the political spectrum, even though we picked news subjects that we thought were contentious. This may be one of problems with identifying fake news. If even news sources with unreliable reputations sometimes produce accurate news, how does the unsuspecting reader know not to trust a source for story X when then can trust it for story Y?

9 Future work

There are a number of areas for potential future work. Regarding the data, possible improvements include automating the mining for new articles and automatically scraping them. To bring the two pipelines together we could use the reference mapper as a source for new articles, this will include referenced articles into our original dataset. The newspaper software library could be adapted to make it usable for our purposes. Furthermore it would be interesting to see the results of a larger dataset, either manually composed or composed by the improvements listed earlier. Another point of future work is solving our initial goal: comparing sentences in the same topical cluster. This is very difficult and would require a lot more work. Finally, many small tweaks could be implemented to increase the performance of the summarizer such as adding lemmatization and positional weights or using a bigger background corpus.

10 References

1. Claimbuster, ‘ClaimBuster: Automated Live Fact Checking’. ClaimBuster, 2018.

   https://idir-server2.uta.edu/claimbuster/
2. PolitiFact. ‘Statements about Fake News’ (2019). PolitiFact, 2019.

   https://www.politifact.com/subjects/fake-news/
3. FactCheck.org. ‘Our Mission’. FactCheck.org, 2019.

   https://www.factcheck.org/about/our-mission/
4. Snopes. ‘About Us’. Snopes, 2019.

   https://www.snopes.com/about-snopes/
5. Adam Mosseri. ‘Working to Stop Misinformation and False News’. Facebook, 2017.

   https://www.facebook.com/facebookmedia/blog/working-to-stop-misinformation-and-false-news
6. The Straits Times. ‘War on fake news: How some countries are fighting misinformation with the law’. The Straits Times, 2018.

   https://www.straitstimes.com/world/war-on-fake-news-how-some-countries-are-fighting-misinformation-with-the-law
7. Lucas Ou-Yang. ‘newspaper3k’. newspaper3k, 2018.

   https://pypi.org/project/newspaper3k/
8. Yohei Seki. ‘Sentence Extraction by tfidf and Position Weighting from Newspaper
   Articles’. Proceedings of the Third NTCIR Workshop, 2002.

   http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings3/NTCIR3-TSC-SekiY.pdf
9. Rada Mihalcea and Paul Tarau, ‘TextRank: Bringing Order into Text’. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004.

   https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf
10. Paco Nathan, ‘A Python implementation of TextRank for text document NLP parsing and summarization’. PyTextRank, 2016.

    https://github.com/ceteri/pytextrank/

11. BBC News. ‘Outcry after police shoot African-American security guard hero’. BBC News, 2018.

    https://www.bbc.com/news/world-us-canada-46187460
12. TensorFlow Hub. ‘universal-sentence-encoder’. TensorFlow Hub, 2019.

    https://tfhub.dev/google/universal-sentence-encoder/1
13. Sanjeev Arora, Yingyu Liang and Tengyu Ma. ‘A Simple But Tough To Beat Baseline for Sentence Embeddings’. ICLR, 2017.

    https://openreview.net/forum?id=SyK00v5xx
14. Yves Peirsman, ‘Comparing Sentence Similarity Methods’. NLP Town, 2018.

    http://nlp.town/blog/sentence-similarity/
15. Google. ‘Facts about Google and Competition’. Google, 2011.

    https://web.archive.org/web/20111104131332/https://www.google.com/competition/howgooglesearchworks.html
16. Hassan, Naeemul, et al. ‘ClaimBuster: the first-ever end-to-end fact-checking system.’ Proceedings of the VLDB Endowment 10.12 (2017): 1945-1948.
17. "Fake News," Lies and Propaganda: How to Sort Fact from Fiction. University of Michigan, 2018.

    https://guides.lib.umich.edu/fakenews
18. T. Sri Rama Raju and Bhargav Allarpu. ‘Text Summarization using Sentence Scoring Method’. IRJET, 2017.

    https://www.irjet.net/archives/V4/i5/IRJET-V4I5493.pdf
19. Elena Lloret and Manuel Palomar. 2012. Text summarisation in progress: a literature review. Artif. Intell. Rev. 37, 1 (January 2012), 1-41.

UNIT
Street no. bus 0000
3000 LEUVEN, België
phone + 32 16 00 00 00
fax + 32 16 00 00 00
@kuleuven.be
www.kuleuven.be