

Progress Report I - Peace Speech Project

Jinwoo Jung (jj2762@columbia.edu), Hojin Lee (hl3328@columbia.edu),
Hyuk Joon Kwon (hk3084@columbia.edu), Matt Mackenzie (mbm2228@columbia.edu),
Tae Yoon Lim (tl2968@columbia.edu)

October 20, 2020

1 Problem Definition and Progress Overview

As the world gets more polarized, one of the biggest challenges we face is hate speech. The harm of hate speech is an active area of research, and the general consensus in the recent literature is that hate speech tends to cause harm to societies. However, while active research is conducted around hate speech, we have not focused on the other side of the story: peace speech.

We were interested in whether peace speech could play a role in bringing peaceful society as some research suggests that peace speech is the DNA of peaceful societies. If the hypothesis is true, we could develop techniques to understand, measure, and track the power of peace speech, which will guide us toward building and maintaining more robust and peaceful communities.

We will look into articles from many different countries, analyze them using natural language processing (NLP) techniques, and study the relationship between the language used in the articles and the peacefulness of the country. This report is the first progress report, which addresses the data engineering, exploratory data analysis and pre-processing of the data.

2 Data

The data that we are working on is the *News on the Web* (NOW) dataset from *corpusdata.org*. According to the source, the data is “composed of 11.2 billion words from web-based newspapers and magazines from 2010 to present times” from 20 countries.

There were two stages in data accessing periods. First, while the project managers were closing the deal with the data vendor and figuring out a Non-Disclosure Agreement (NDA), we were given sample data to work with on September 28th, 2020. Sample data included a source file, which contained the news article’s ID, number of words, date of issuance, country, publisher, url, and title of an article. The other file was a text file, which contains the news article ID and the actual text content of the article. There were about 3000 news articles contained in the sample dataset. Please refer below for the screenshots for the format of the .

On October 6th, 2020, we then received access to the full dataset from NOW. The dataset is composed of multiple files, divided into the same two types as the sample data: sources and text files. For the source files, although the content’s structure is the same as the sample data, the files are divided into year-month combinations and some files combined into “part1” and

“part2”, in which many year-month sources are combined. On the other hand, text data are stored in subfolders of year-month format. There were about 20 million news articles contained in the full raw dataset. Please refer to Figure 4-1 and below screenshots for the structure.



textID	#words	date	country	website	url	title
11241	397	13-01-06	US	Kotaku	http://kotaku.com/5973495/author-of-the-warriors-cult-film-adapted-to-h	
11242	757	13-01-06	US	Michigan Radio	http://michiganradio.org/post/thats-what-they-say-dialect-socie	
11243	755	13-01-06	US	New York Daily News	http://www.nydailynews.com/life-style/eats/best-new-yor	
11244	1677	13-01-06	US	OregonLive.com	http://www.oregonlive.com/performance/index.ssf/2013/01/reflect	
21242	794	13-01-11	US	Ars Technica	http://arstechnica.com/gadgets/2013/01/ask-ars-does-facebook-au	
21243	690	13-01-11	US	NBCNews.com	http://worldnews.nbcnews.com/_news/2013/01/11/16463878-accused-	

Fig 2-1. Screenshot of Sample Source File

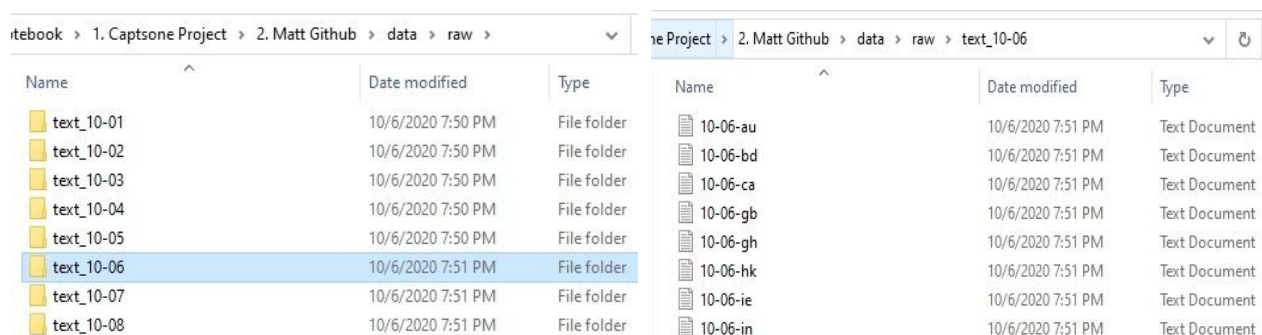


@@11241 <p> Sol Yurick , the writer whose 1965 novel " The Warriors " was adapted into a film 14 years later -- which then became one of the best adapted me invoking it , typified by the iconic " Baseball Furies " the protagonist Warriors fight in Riverside Park . After making their way through rival gangs of big hits . Primarily a brawler , with some limited open-world features , the game also served as a canonical prologue to the all-gang meeting in the B

@@11242 <h> That 's What They Say : Dialect Society chooses its words of the year <p> For this week 's edition of " That 's What They Say , " University egories ? <p> Anne : Marriage equality , which was in the runoff for word of the year , actually won most likely to succeed . And I thought it was a very a mixed-gender audience . <p> Rina : That 's never happened to me , has it to you , Anne ? <p> Anne : No , never . <p> The mansplaining was up against " I least likely to succeed ? <p> Anne : That was a tie . One of the first I remember . It was a tie between " YOLO , " which our listeners over about 25 or : is " binders full of women . " After presidential candidate Mitt Romney said that , it instantly became a hashtag , and many people have probably seen it

@@11243 <h> A sublime croissant at French Tart in Grant City , Staten Island . <p> French Tart chef Laurent Chavenet claims to work in the only authentic wo types of flour in its croissants . <p> In 1986 , pastry chef Hubert Colson first opened his famed pastry shop in Mons , Belgium . Barely 20 years later master pastry chef Laurent Dupal and his Spring St. shop Ceci-Cela . Since his quaint cafe opened in 1992 , Dupal has consistently made some of the flaki le slow . -- Tzivia M. <p> Stork 's Bakery , located at 12-42 150th St. in Whitestone , makes the perfect croissant . Its golden , flaky , light crust is GEST IT , WE 'LL TEST IT <p> We 're in search of the best of the city , but we need your help . Send your picks for the following , and we 'll try them !

Fig 2-2. Screenshot of Sample Text File



Name	Date modified	Type
text_10-01	10/6/2020 7:50 PM	File folder
text_10-02	10/6/2020 7:50 PM	File folder
text_10-03	10/6/2020 7:50 PM	File folder
text_10-04	10/6/2020 7:50 PM	File folder
text_10-05	10/6/2020 7:50 PM	File folder
text_10-06	10/6/2020 7:51 PM	File folder
text_10-07	10/6/2020 7:51 PM	File folder
text_10-08	10/6/2020 7:51 PM	File folder

Name	Date modified	Type
10-06-au	10/6/2020 7:51 PM	Text Document
10-06-bd	10/6/2020 7:51 PM	Text Document
10-06-ca	10/6/2020 7:51 PM	Text Document
10-06-gb	10/6/2020 7:51 PM	Text Document
10-06-gh	10/6/2020 7:51 PM	Text Document
10-06-hk	10/6/2020 7:51 PM	Text Document
10-06-ie	10/6/2020 7:51 PM	Text Document
10-06-in	10/6/2020 7:51 PM	Text Document

Fig 2-3. Screenshot of Full raw data structure

3 Data Engineering

There was a specific request from the project management team in terms of data structure. They wanted to have a structure which will enable them to easily transfer our work to other projects after the capstone is finished. They requested each article be extracted into its own file and placed in a folder following the convention of “[Country]/[Year].” When dealing with the sample data, this was almost trivial since our data was so little, however, it presented a real challenge when we had more than 60GB and 20 million articles to process.

As stated prior, our data came in two files that needed to be joined together, the sources and text files. All together there were 49 source files and 129 text folders, with each folder containing at least 20 text files (one for each country and some “na” countries). This data was too large to fit in memory all at once, and therefore needed to be exported to the requested format in batches. One addition we made to the file structure was to include the publisher within the folder hierarchy, so that the final output was “[Country]/[Publisher]/[Year]/[text_file].txt”. This has the added benefit of allowing us to query by publisher further upstream in our preprocessing pipeline.

To perform the data export as efficiently as possible, our process started with sequentially reading in each source file. Using pandas, we performed multiple groupings to iterate over each country, year, and month, within the source file. Using our groupings we can locate the corresponding text file and begin exporting the text to the preferred output.

One challenge faced here is the text file folders and names do not always follow the same convention. For example, there are text folders *text-17-01* and *text_15-12*, there are text files *16-03-us.txt*, *19-12-us1.txt*, *text_18-01-US.txt*, and also other formats. To make sure we are always capturing the correct data, we need to split the folder and file names by either a dash or hyphen, and match the current country, year, and month to the correct file.

4 Exploratory Data Analysis

After fully parsing the full raw dataset from the sources, we started to analyze the breakdown of articles by the number of articles per publisher. The total number of articles from the full dataset contains about 20 million articles. In order to perform text-preprocessing and further analysis, we believed that we must come up with a way to sub-sample the whole dataset. Since the number of articles from publishers that have more than 1000 articles accounts for around 85% of the full data, we are planning to use articles from these publishers that have more than 1000 articles for future analysis. Please refer to the table below where it shows a breakdown of the volume of data by publisher size.

Number of Articles Published	Publishers		Articles	
	Count	%	Count	%
1	6,762	18.1%	6,762	0.03%
2	3,256	8.7%	6,512	0.03%
3	1,968	5.3%	5,904	0.03%
4-5	2,540	6.8%	11,250	0.06%
6-10	3,312	8.9%	25,581	0.13%
11-25	4,171	11.2%	69,968	0.35%
26-100	5,487	14.7%	292,704	1.45%
101-500	5,321	14.2%	1,266,238	6.27%
501-1000	1,586	4.2%	1,124,211	5.57%
1000+	2,994	8.0%	17,381,823	86.09%

Fig 4-1. Breakdown of full dataset by size of publisher

Furthermore, we grouped articles by country and year to see if there exists any patterns. We observed that the number of articles across the countries are uneven, and some countries such as the United States have significantly more articles than other countries. On the other hand, the distribution of the number of articles by year stays relatively unchanged. For most countries, we get a significantly larger amount of data from 2016 and onward.

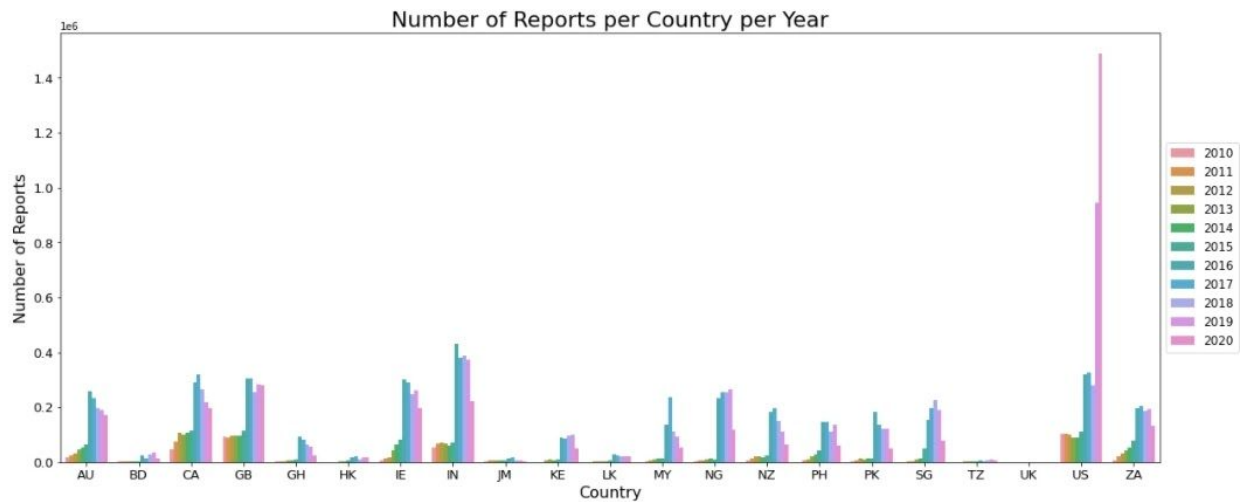


Fig 4-2. Number of Articles per Country per Year. Y-axis scale is $1e6$.

The below is the analysis of the word count. Figure 4.3 shows the total word count of all the countries. We can observe that the average number of words per article are similar across countries and years. These observations seem to arise from how the raw data is scraped from websites in that each news article and its content in terms of number of words are capped at a certain limit. This is explained more in detail in part 4 when we talk about text-preprocessing, but in short, we observed that some parts of articles got removed and instead it was replaced with a sequence of 10 @ signs instead.

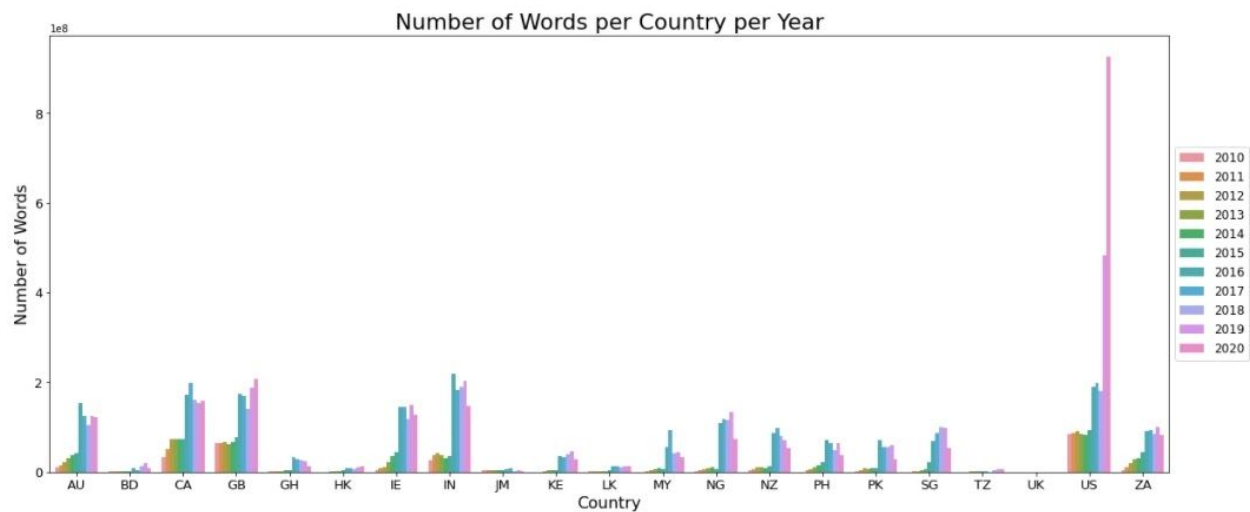


Fig 4-3. Number of words per Country per Year

4 Text Pre-processing

When we initially inspected the sample raw files, we identified unnecessary information that may affect our analysis later in the project. For example, the scraped text data contained phrases and sentences that are asking readers to subscribe to their publisher or directing readers to the related articles. Since these sentences may involve words that are later recognized as the lexicons that represent peaceful or hateful societies, we decided to filter out the noisy information that are not related to the news articles. Since the ways the noisy data are presented differ by each article and publisher, we could not manually detect each noisy pattern individually. Accordingly, we decided to rely on models to automatically remove the unnecessary contents.

This text-preprocessing procedure is composed of two procedures. The first procedure applies generally to all models (N-Gram, Cosine Similarity, and the merged model), and it cleans the scraped news article into easily readable sentences. The second procedure will be chosen based on both the accuracy of how well each model removes the noisy data and efficiency of how quickly we would be able to perform the cleaning.

4.1 General Text Pre-processing

As can be seen on the Fig 2-2 above, the raw texts both from sample and full data which contains news articles contain various kinds of unnecessary information. In order to apply the specific models such as N-Gram, Cosine Similarity, the merged model to filter out the noisy data, we first had to transform the raw scraped sentences into a cleaner form. We removed all the html tags such as <p> and <h>, and symbols such as {, }, <, >, \, (,), \n, and @. Furthermore, Since we apply a sentence tokenizer to further preprocess data, we convert symbols such as :, ;, ?, ! to periods to facilitate dividing the news articles into sentences. One thing to note is that the majority of the raw text data contains the sequence, "@ @ @ @ @ @ @ @ @ @", which seems to indicate some parts of original news articles are removed from the 'raw' data itself. This is validated not only by looking at some actual news articles on websites, but also by our data analysis performed with the sample data in that the number of words in each article are all similar and do not have much variation.

4.1 N-Gram

From our initial observations of the textual data, we identified that although the way in which the noisy text is presented differs between news articles, there exists recurring noisy patterns of advertisements and slogans when we group the data by publisher. For example, sentences that prompt readers to subscribe to newsletters may have different variations of words for each publisher. However, when we look at each particular publisher, specific patterns such as "Please subscribe to our newsletter" recur throughout multiple news articles by the same publisher. Therefore, we initially thought of applying an N-Gram model, which looks at combinations of N words in each sentence of articles. After detecting publisher-specific patterns, we tried to make a systematic way to check for overlapping phrases.

First, we grouped articles by publishers. As we mentioned above, advertisements and slogans are publisher-specific, so we only had to figure out which sentences to remove within publishers. Thus, we tokenized each sentence within the article. By tokenizing each sentence, it made classification tasks easier as we could remove whole sentences that contain unnecessary phrases. After tokenizing sentences, we used the N-gram method, specifically 5-grams, to create phrase lists. One of the key assumptions we had in this method is that it is uncommon to have a phrase consisting of 5 exact same words that shows up multiple times across different articles. After creating a list of 5-gram phrases, we measured frequencies of the phrases across the documents. Then, if the frequency is found to be larger than 25% of the total documents, then we removed sentences that the particular phrases are associated with.

4.2 Sentence Embedding Cosine Similarity

Unlike the N-Gram model which works only when we group the data into each publisher, we also thought of performing text preprocessing using Cosine similarity with sentence vectors, which works by each individual article regardless of whether we group the data or not.

How this model works is as follows: Once we tokenize a document into sentences, we embed each sentence in a document using a pretrained model. We utilized sentence transformers from HuggingFace (Wolf et al., 2020), and we particularly used Sentence-Bert (Reimers & Gurevych, 2019), which has the highest benchmark scores. The Sentence-BERT model allows us to embed sentences into 768 dimensional vectors. We then used the model to embed a document and each sentence associated with it. Key aspect of this model is that once we compute cosine similarity between each sentence and document, we can then remove sentences that are found to have low level of similarity measure. For the hyperparameter, we decided to use 0.95, which means that we filter out sentences that have similarity measures higher than 0.95 (i.e. cosine similarity of 0 indicates a sentence to be most similar to a document while 1 indicates the opposite).

4.3 Result

As can be seen below in Fig 4-1 and Fig 4-2 both N-grams and Cosine similarity are successful in removing unwanted text. However, both methods have trade-offs in the way they detect unwanted sentences.

Since we are using 5-grams to detect repeated sentences the N-Gram model is unable to delete repeated sentences that are less than 5 words long. This is the reason why the sentence ‘from around the web.’ is not deleted in Fig 4-1. Also the N-gram model fails to delete phrases that are repeated but not exactly repeated such as sentences that contain the current date. The Cosine similarity on the other hand is able to detect sentences that contain the current date and is able to delete sentences that are less than 5 words long. The N-Gram model is although safer in what it chooses to remove as sentences that have repeated 5 grams in them are most likely to be spam. If the document is clean and doesn’t have any spam the N-Gram will leave the text as it is. On the other hand the Cosine similarity model often takes out sentences that are not spam and there is no way to control this as we are using a fixed threshold to control the results. This happens especially when the article is long and doesn’t have spam sentences to remove to begin with. This serves as a problem as a lot of the articles didn’t contain spam to begin with.

As for the time complexity of these tasks the Cosine Similarity model took an average of six minutes per each publisher which contained 250 articles each. The N-Gram model takes an average of 11 seconds to process a publisher with the same amount of articles. However, the Cosine similarity model is expected to increase linearly with the increase in documents as the method processes documents one at a time. But as the N-Gram model processes the documents by publisher it’s complexity will increase exponentially dependent on the number of articles in a

specific publisher. For example, it is very difficult to create this same sort of N-Gram model using over 10000 articles.

treat obesity through modification of the gastroin
testinal tract and reduction of nutrient intake .
related. from around the web.
more from the times of india. recommended by colom
bia. from around the web. more from the times of i
ndia. recommended by colombia. comments. character
s remaining 3000. or proceed without registration.
share on twitter. sign in with. facebookgoogleemai
l. refrain from posting comments that are obscene
defamatory or inflammatory and inciting hatred aga
inst any community . help us delete comments that
do not follow these guidelines by marking them off
ensive . let's work together to keep the conversat
ion civil .

treat obesity through modification of the gastroin
testinal tract and reduction of nutrient intake .
from around the web. characters remaining 300
0. or

Fig 4-1. Comparison between initially preprocessed news article vs pre-processed version using N-Gram, 2010.01.02 Times of India's article, ID: 1335637

on of nutrient intake . related. from around the w
eb. more from the times of india.
recommended by colombia.
from around the web. more from the times of india.
recommended by colombia. comments. characters rema
ining 3000. or proceed without registration. share
on twitter. sign in with. facebookgoogleemail. ref
rain from posting comments that are obscene defama
tory or inflammatory and inciting hatred against a
ny community .
help us delete comments that do not follow these g
uidelines by marking them offensive
. let's work together to keep the conversation civ
il
.

on of nutrient intake . related. from around the w
eb. more from the times of india.
from around the web. more from the times of india.
sign in with.
help us delete comments that do not follow these g
uidelines by marking them offensive
.

Fig 4-2. Comparison between initially preprocessed news article vs pre-processed version using Sentence Embedding Cosine Similarity, 2010.01.02 Times of India's article, ID: 1335637

5 Goals and Next Steps

Although both the sentence embedding cosine similarity model and N-Gram model seem reasonable heuristically, we thought we would need it would be great if we have metrics to objectively measure accuracy. We are in the process of building such models, and we would include the results in future write-ups.

Once we finish pre-processing of the data, we are planning to start actual analysis of the articles. We will start from the Bag of Words (BoW) approach using existing lexicons of peace speech and hate speech. We will analyze this question from multiple dimensions by segmenting data by countries, time and publishers.

Next, we will see how external events affect the sentiment of the articles. We will choose a couple of major events happening in each country, and try to assess whether these major events

are early indicators of changes in the overall sentiment of the articles and choice of vocabulary. For further analysis, we will discuss with Dr. Coleman, the project owner, and update it later on.

6 Contribution

- **Jinwoo Jung** (team captain): Main contributor on review, analysis, and implementation of N-gram model for data pre-processing.
- **Hojin Lee**: Co-set up meetings, milestones and manage progress. Contributed to building pre-processing models.
- **Hyuk Joon Kwon**: Cleaned and merged source codes. Main contributor in installation and implementation of each model, comparing efficiency and results.
- **Matt Mackenzie**: Developed source codes to parse and organize both sample and full raw dataset, performed data engineering, and outputted summary statistics of dataset.
- **Tae Yoon Lim**: Main contributor on research and literature review of sentence embedding cosine similarity model for data pre-processing

References

- [1] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019, May 24). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Retrieved 2020, from <https://arxiv.org/abs/1810.04805>
- [2] Lin, Y., Michel, J., Aiden, E., Orwant, J., Brockman, W., & Petrov, S. (2012, July 01). Syntactic annotations for the Google Books Ngram Corpus. Retrieved 2020, from <https://dl.acm.org/doi/10.5555/2390470.2390499>
- [3] Reimers, N., & Gurevych, I. (2019, August 27). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Retrieved 2020, from <https://arxiv.org/abs/1908.10084>
- [4] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., . . . Rush, A. (2020, July 14). HuggingFace's Transformers: State-of-the-art Natural Language Processing. Retrieved 2020, from <https://arxiv.org/abs/1910.03771>