Summarizing Shelter-in-Place Ordinances with NLP

Rowan Fitch, Kailen Shinmoto

March 2020

1 Question

With the declaration of COVID-19 as a global pandemic, numerous countries have began enforcing shelter-in-place ordinances to try and slow the spread of the virus. Using NLP, we plan to summarize the shelter-in-place ordinances among the United States, as well as ordinances from other countries, and compare the summaries to determine which countries/states are taking the most precautions regarding COVID-19.

In the United States, there has been some confusion as to what a stay-at-home ordinance entails. A summary of a stay-at-home ordinance can help citizens by highlighting the main points of the ordinance. Along with providing a summary, this project can also highlight which countries/states are taking more precautions, and see if there is any correlations between those taking more precautions and the current infection rate in those respective countries/states.

2 Text Summarization

Text summarization is not a new topic in NLP. Infact, there are different approaches to summarizing text.

2.1 Latent Semantic Analysis

Latent Semantic Analysis is an unsupervised model that uses vector semantics to determine relations between sentences. For example, two sentences that contain a high number of the same words would be semantically related[2]. The first step in Latent Semantic Analysis is to convert the input document D into a matrix M where the rows of M represent words and the columns represent sentences. The cells in M can be a metric such as the Tfidf-weight of a word or the Term-frequency of a word. The second step of Latent Semantic Analysis is to incorporate Singular Value Decomposition, an algorithm that decomposes the matrix M into three matrices M_1, M_2, M_3. After M has been decomposed, the third step is to select important sentences based on M_1, M_2, M_3.

2.2 Bayesian Classifier

A Bayesian Classifier calculates the probability of a sentence s being included in a summary S given k features using Bayes Rule[3]. The probability is modeled as:

$$P(s \in S | f_1, f_2, ..., f_k) = \frac{P(f_1, f_2, ..., f_k | s \in S) P(s \in S)}{P(f_1, f_2, ..., f_k)} \quad (1)$$

2.3 Lexical Chain

The Lexical Chain model starts by first using POS tagging to collect all the nouns from a source document. Then, for each noun n, all possible lexical chains L are formed and given a score, then stored in an array. The final step involves a nested loop where for each noun n, every chain L such that $n \in L$ is evaluated by determining the change in the score of L if n is removed. The chain whose score is most greatly affected by removing n then becomes the placeholder for n, and this process is repeated until a list of chains $L_1, L_2, ..., L_n$ is produced[4].

2.4 Weighted Frequency

The Weighted Frequency model first tokenizes a document, and calculates the weighted frequency F of a word w by dividing the frequency of the word in the document f_w by the maximum f_w of all the words in the document. Sentences are then represented numerically as the sum of the weighted frequencies of the words in the sentence. The sentences with the highest sums are the summaries of the document[5].

2.5 Graph Theoretical Approach

The Graph Theoretical Approach model displays all the sentences of a document as vertices in an undirected graph G. Two sentences, s_1 and s_2 are connected by an edge e if their cosine similarity is above a certain threshold[6]. The sentences with the highest cardinality, or the most edges, are the sentences that form the summary of the document.

References

[1] Mervosh, Lu, Swales, "See Which States and Cities Have Told Residents to Stay at Home", The New York Times

[2] Ozsoy, Alpaslan, Cicekli, "Text summarization using Latent Semantic Analysis", Journal of Information Science, vol 37, 2011, pages 405-417

[3] Kiyani, Tas, "A survey automatic text summarization", Pressacademia, vol 5, 2017, pages 205-213

[4] Silber, Mccoy, "Efficient Text Summarization Using Lexical Chains"

[5] Malik, "Text Summarization with NLTK in Python", https://stackabuse.com/text-summarization-with-nltk-in-python/

[6] Gupta, Lehal, "A Survey of Text Summarization Extractive Techniques", Journal of Emerging Technologies in Web Intelligence, vol 2, 2010, pages 258-262