Quarantine Quibbles: A Sentiment Analysis of COVID-19 Tweets

Jason Nguyen* and Ritu Chaturvedi†
School of Computer Science,
University of Guelph, Guelph
Email: *jnguye21@uoguelph.ca, †chaturvr@uoguelph.ca

***Abstract*—With the advent of the COVID-19 pandemic, people have flocked to social media in order to stage their thoughts surrounding these unusual circumstances. This paper aims to uncover public sentiment regarding the novel coronavirus pandemic on the microblogging platform *Twitter*. This is done through a proposed algorithm that builds off of existing aspect-based sentiment analysis approaches and opts for a Naïve-Bayes route to classify existing Tweets that have been atomized into n-grams. This research concludes that overall sentiment regarding the COVID-19 outbreak over July 2020 is a combination of pessimism and dejection as our quarantine denizens take to their online platforms in airing their polemic opinions.**

Keywords: sentiment analysis, opinion mining, machine learning, supervised learning, data mining, text mining, microblogs, spam post exclusion

I. Introduction

Opinion mining (otherwise known as sentiment analysis) is a subset of natural language processing and computational linguistics that aims to identify, extract, measure, and analyze the affective states and subjectivity of a particular corpus of information. Within this sub-discipline lies the more granular field of *aspect*-based opinion mining, which seeks to atomize (break up) a corpus of text into pertinent keywords or features that represent the essence of the given topic. A fine-grained analysis is performed on these "aspects". A generic description of aspect-based opinion mining is:

1) Obtain a corpus of information that contains pertinent opinions to be mined, usually consisting of raw data from social media/product review sites (*Amazon*, *eBay*, *Facebook*, etc.)
2) Pre-process the data. This usually consists of removing redundant data ("noise") that would otherwise interfere with later analyses
3) Derive a list of candidate aspects from the data. Although some methodologies involve deriving n-grams from the data or probabilistic models that incorporate expectation-maximization algorithms, a simple method of deriving candidates is to examine the nouns [1], [2]. n-grams are continuous sequences of size-n from a given piece of text. For example, for the text "I went to work",
   - The 2-grams would be "I went", "went to", and "to work"
   - The 3-grams would be "I went to", and "went to work"
   - The 4-gram would be "I went to work"
4) Prune the list of candidate aspects in order to create a more-relevant list of aspects
5) Cross compare the pruned aspects to the original data in order to find sentimental connections between the data.

II. Related Work

Existing work in mining aspects include Das and Kannan's probablistic model for ranking aspects, which consists of indicators such as "uniqueness", "diversity", and "burstiness" and also aims to combat the noise, redundancy, and ambiguity of Twitter posts [1]. This article, written by two Microsoft researchers, highlights the obstacles involved with attempting to analyze the inherent noise of topical phrases dispersed among microblogging sites such as Twitter, as well as the probabilistic model used to combat such noise. A topical phrase is a sentence or sentence fragment that expresses the essential idea of a paragraph, referred to in this article as "aspects". A probabilistic model is a model that incorporates probability in its implementation. Characteristics of aspects were quantified by three pertinent metrics, 'uniqueness', 'burstiness', and 'diversity' which were used to discern relevant phrases without requiring domain knowledge or manual effort. A limitation of this approach is its inability to converge semantically similar aspects together into an overall theme, a problem that could in theory be rectified through other NLP techniques. Due to Twitter's inherent "noisiness", this article will prove helpful in finding COVID-19 related search phrases (aspects) in spite of that, and as such will result in more quality data returned from the mining.

Samuel et al. [3] conducted an exploratory analysis on over one million Twitter posts and, like Das and Kannan [1], analyzed the data using several different features such as "content", "motive", and "richness" in addition to the typical sentiment. The authors write about the insights and patterns discovered from a heavily filtered dataset of Tweets, sieved first to 40,000 lead Tweets and then finally to 18,000 filtered Tweets. Although 9 lexicons were originally used for sentiment analysis, only two metrics (Jockers and Sentiword) were kept in the final report. The article didn't go into too much detail about the sentiments themselves, but provided extremely valuable correlations between other metrics (such as between retweet count and sentiment) that shed an interesting light on potentially confounding variables later on in this COVID-19

research project. This and Esuli & Sebastiani's article both take a look at SentiWordNet, though the "Sentiword" library used in this article is a more-applied module derived from SentiWordNet. Both solutions will be examined for efficacy in determining a suitable sentiment analytics tool for gauging the disposition of COVID-19 Tweets. Samuel et al. also proposed a different sentiment analysis methodology using n-grams, Syuzhet, and sentimentr to classify COVID-19 microblog posts in order to paint a more optimistic picture regarding several re-opening scenarios [4]. This article provides a fresh point of view with regard to sentiment analysis with the COVID-19 pandemic in that it analyzes American sentiments towards reopening the U.S. economy and transitioning back to almost-normal life. A sentiment in this context refers to the emotional polarity (or general disposition) of a particular post on a social media site such as Twitter. The article takes on a simple, intuitive approach to collecting data by means of word frequency and n-gram analyses in order to quantify Tweets based on varying sentiment metrics. An interesting conclusion drawn from this is that even though there were more positive Tweets, the negative Tweets that did surface were quantifiably more extreme than the positive ones. This, along with the other article authored by Samuel et. al. will prove useful in this upcoming research project regarding COVID-19 Tweets in that it provides a useful data model to start from with respect to which metrics to measure from, though an area to improve in with the annotator's project is to perform more post-processing on the n-grams in order to hypothetically derive more meaningful data (especially with the changing times).

Ejieh et al. [5] proposed a three-step "Microblog Aspect Miner" algorithm to find relevant aspects. This article highlights a 'Microblog Aspect Miner' algorithm that aims to identify sentiments and opinions of product aspects present on the microblogging platform Twitter in spite of noisy data. An aspect (otherwise known as a "topical phrase") is a phrase or sentence that summarizes the essence of a particular topic of interest (it would be COVID-19 in this research project, for example). By using a three-step model that continuously refines the dataset ("Tweets") until only subjective polarity data remain, these findings improve upon Das and Kannan's article [1] by grouping together similar aspects accordingly, though in theory one could streamline the preprocessing step by utilizing more expressive API queries in order to remove extraneous raw data such as retweets. This research project and cited article both draw on the point that objective data serve no purpose in analyzing the sentiments of microblog users. Whereas the Das and Kannan article [1] was very theoretical in nature, this article was very applied and relevant to the research being conducted and will likely be an asset in the future. The insightful methodologies in this article and Das & Kannan's article complement each other and both are used in this research to ensure a consistent-but-accurate corpus of mined data with regard to COVID-19.

III. Proposed Methodology

This algorithm takes in public Twitter posts and, after rigorous preprocessing to remove redundant noise (hashtags, mentions, URLs), classifies them as either "positive" or "negative". It then derives n-grams (from 1- to 4-) and tallies up the number of positive and negative Twitter posts against them in order to derive aspect polarity.

Algorithm 1 COVID-19-Tuned Sentiment Analysis

Input: A keyword to search the relevant aspects of (in this case, it would be "coronavirus" as it is the most popular keyword used to refer to the virus as per *Google Trends* at the time of writing).

Output: A list of n-grams along with their occurrence counts, absolute positive score, absolute negative score, relative positive score, and relative negative score

1. **Data Extraction**: collect all Twitter microblog posts regarding the given keyword over a period of time

2. **Data Preparation**: call the preprocessing module. In addition to cleaning up the data, this converts posts into a normalized form using a lemmatizer, (a tool used to convert all inflected forms of a word (e.g. swam, swum) back into the canonical/base form), and then prunes the posts that are either too objective or are otherwise unfit for continued analysis.

3. **Data Mining**: call the tokenization module to mine the n-gram aspects from the aforementioned microblog posts. This step also uses a Naïve-Bayes classifier to determine whether each of the original Twitter posts is positive or negative.

4. **Data Analysis**: relate earlier Tweet classifications with the n-gram aspects in order to derive an understanding of where sentiments lie with each topic

A. The Dataset

Three corpora of Tweets were collected during the weeks of late-June to late-July, each consisting of about 800,000 microblog posts (Tweets) and represent the search terms "covid", "coronavirus", and "corona". They were scraped using the freely-available Tweepy API on a Twitter developer account.

B. Data Extraction

The Tweepy library was used to extract Twitter posts over an approximately one-month period. Three candidate search terms were taken into account, resulting in three corpora:

- "covid"
- "coronavirus"
- "corona"

In order to extract the full 280 characters (originally 140), the search was conducted using "extended search mode", a Twitter API call extension that allows applications to make use of the new 280 character limit without breaking legacy systems. In addition to that, the specifier "-filter:retweets" was used in addition to the query in order to get rid of any possible retweets from the result, as these typically do not provide anything novel to sentiment analyses.

C. Data Preparation

The inherent issue with Twitter posts is what defines them in the first place: their brevity. This makes it easy to marr attempts to analyze its data due to the "noisiness" in the corpus. It is therefore important to remove such noise, which includes:

- Hashtags
- Mentions
- Emojis and smileys
- URL links (all posts that use a URL are removed, as these typically are objective posts that serve no use to the sentiment analysis. This usually results in 80% of the input data being purged, unfortunately [2].)

This is achieved in this paper through Said Özcan's tweet-preprocessor library, which automatically cleans all of the aforementioned "noise" [6].

The next part uses the Natural Language Toolkit, a powerful and well-known Python NLP (natural language processing) library [7]. Some commonly-used terminology relevant to this paper's usage of NLTK follows:

- **Part of speech**: the grammatical role that a word has within the context of a sentence, including (but not limited to):
  - **Cats** are notorious little creatures — the bolded word is a noun
  - His **cat** fell down in the garage — the bolded word is a noun
  - I just **lost** the game — the bolded word is a verb
  - She is very **pretty** — the bolded word is an adjective
  - They ate very **quickly** — the bolded word is an adverb
- **Part of speech (PoS) tag**: a given library's representation of the part of speech. For example, NLTK uses the tag **NN** for singular nouns and **VBD** for past tense verbs.
- **Lemmatization**: the process of converting a word into its canonical (or base) form using the context of the words surrounding it. For example, "I ate an apple" could be lemmatized to "I eat an apple".
- **Stemming**: a similar process to lemmatization, but devoid of any surrounding context. Stemming typically is done using raw, contextless rules (such as dropping a suffix regardless of whether the word exists).
- **Stop words**: function words that serve no direct meaning in a sentence, such as "the", "of", "a", "to", among others.

After the initial cleaning is finished, each post is lemmatized, which provides the contest for NLTK's pos_tag function. This allows NLTK's WordNetLemmatizer to accurate convert each word to the sought-after base form; this context given by the PoS tag differentiates lemmatization from "stemming", which attempts to reduce a word to its base form regardless of the context around it [7].

D. Data Mining

The first part of the data mining step involves pruning any non-subjective posts. Using a Naïve-Bayes supervised learning approach, a corpus of 5,000 positive and 5,000 negative Twitter posts was used to train a classifier based on two features:

- Positivity – a measure of how much of a positive sentiment was emitted
- Negativity – a measure of how much of a negative sentiment was emitted

This model is then used to classify microblog posts within a certain threshold. As the training data, sourced by NLTK's corpora database, does not include neutral microblog posts, the metric to measure objectivity was decided to be the absolute difference between the positivity feature and the negativity feature, with posts having a difference of less than 0.30 being removed from the set.

It is typical for researchers to remove "stop words" from their corpora, but this research errs on the side of caution that important holistic context would be lost in the event of removing stop words. Ejieh et al.'s MAM [5] was relatively granular and involved using the SentiWordNet [8] to individually sum up the posts' positive and negative scores.

However, with microblog posts as short as Twitter's, stop words sometimes provide much-needed context, especially since the training data provided by NLTK consist entirely of raw Twitter posts: mentions, stop words, hashtags and all. This makes it especially powerful against the typo-ridden, slang-heavy nature of the platform. Naïve-Bayes has also been shown to be a strong classifier for document analysis, beating out neural networks and decision trees in a number of situations [9], [10].

The next part of the data mining step takes the subjective microblog posts and creates aspects out of n-grams. This is what differentiates ordinary sentiment analysis from *aspect*-based sentiment analysis – by tokenizing texts into smaller parts, one can derive a more-granular analysis of the corpus. As topics regarding COVID-19 comprised compound phrases (e.g. "wearing masks", "new cases", "get tested for"), it was fitting to expand on single-keyword aspect-based sentiment analysis by experimenting with n-gram analysis.

E. Data Analysis

The polarity of the n-gram aspects is determined in this step. Iterating over each microblog post in the corpus, the algorithm counts the number of posts where the aspect was classified as positive or negative. These two numbers are added together to derive the sentiment of the aspect.

A consequence of not removing the stop words from earlier is that a large number of aspects have meaningless tokens. However, this decision did end up creating some fruitfal aspects such as "because of covid" and "the covid crisis", the former potentially being reduced to "because covid" and "covid crisis", aspects that could have nuanced differences.

IV. Results and Evaluation

Due to time constraints, a more-detailed analysis could not be conducted. The n-gram aspects still contain some pertinent information, however.

These tables are curated from the raw output of the algorithm, as many n-grams were filled with stop words. The following tables have the following columns:

- **Aspect**: the n-gram aspect (1- to 4-gram)
- **Count**: the number of times the aspect appeared in any particular corpus (the table has curated rows from all three search queries; in the event of a conflict, the row featuring the highest count was preferred)
- **Positive**: the number of positive Twitter posts that contained this aspect
- **Negative**: the number of negative Twitter posts that contained this aspect
- **Pos%**: the percentage of Twitter posts containing this aspect that were classified as positive
- **Neg%**: the percentage of Twitter posts containing this aspect that were classified as negative
- **Sentiment**: the overall conclusion of the aspect's sentiment, which is the difference between the Pos% and Neg%. "+" denotes "Positive", "–" denotes "Negative", and the number of either symbol is determined by the magnitude of the difference:
  - 0.00 – 0.25 difference: "+" or "–"
  - 0.26 – 0.50 difference: "++" or "– –"
  - 0.51 – 0.75 difference: "+++" or "– – –"
  - 0.76 – 1.00 difference: "++++" or "– – – –"

A. General Overview

A preliminary analysis of some popular aspects shows that most sentiments are relatively negative. There is some inherent risk in indexing the aspect "corona", due to its ambiguity (astronomers and beer fans may chime in here), though surprisingly it shares the same sentiment features as "covid" and "coronavirus".

B. 2-grams

Examining the 2-grams paints a similar story, with noticeable defiance against aspects "the government" and "white house". "post corona" shows positive sentiment, likely owing to some Twitter users' optimism in putting the pandemic behind them.

C. 3-grams

The prevalence of 3-grams thins out faster than the 2-grams, presumably because of how uncommon it is to match three words in a row. There is a noticeable air of fear present here: "because of covid", "die of corona", "get the coronavirus", and "stay far away" being some notable examples. Interestingly, "suffer from corona" skews strongly positive.

There are too few 4-grams to draw meaningful discussion on.

Aspect	Count	Positive	Negative	Pos%	Neg%	Sentiment
corona	462983	169204	293779	0.365	0.635	– –
covid	358091	133533	224558	0.373	0.627	– –
coronavirus	107611	44834	62777	0.417	0.583	–
virus	96390	39378	57012	0.409	0.591	–
corona virus	79523	32309	47214	0.406	0.594	–
people	56925	17084	39841	0.300	0.700	– –
the corona	44880	17616	27264	0.393	0.607	–
test	36216	13410	22806	0.370	0.630	– –
case	29822	18677	11145	0.626	0.374	+ +
time	28940	10594	18346	0.366	0.634	– –
mask	28143	13895	14248	0.494	0.506	–
the coronavirus	27377	11578	15799	0.423	0.577	–
test	25692	7374	18318	0.287	0.713	– –
year	22501	6997	15504	0.311	0.689	– –
die	20563	4266	16297	0.207	0.793	– – –
trump	19744	7849	11895	0.398	0.602	–
spread	15918	12606	3312	0.792	0.208	+ + +
home	15870	4599	11271	0.290	0.710	– –
country	13016	7870	5146	0.605	0.395	+
hospital	11951	4069	7882	0.340	0.660	– –
government	11749	3321	8428	0.283	0.717	– –
death	11616	4694	6922	0.404	0.596	–
test	10198	3894	6304	0.382	0.618	–

TABLE I
POPULAR ASPECTS OF COVID-RELATED SENTIMENTS

Aspect	Count	Positive	Negative	Pos%	Neg%	Sentiment
wear mask	4070	1953	2117	0.480	0.520	–
the government	3099	830	2269	0.268	0.732	– –
spread corona	2980	2436	544	0.817	0.183	+ + +
corona crisis	2519	726	1793	0.288	0.712	– –
post corona	1348	837	511	0.621	0.379	+
donald trump	1034	482	552	0.466	0.534	–
white house	913	170	743	0.186	0.814	– – –
death rate	710	121	589	0.170	0.830	– – –

TABLE II
2-GRAMS OF COVID-RELATED SENTIMENTS

V. Limitations

Due to time constraints, a lot of due research was not conducted, nor were there any quantifiable measurements done against the methodologies this algorithm was based on.

Ejieh et al.'s MAM [5] algorithm programmatically resolves a list ranking each of the one-word aspects using Word2Vec, a neural network model used to create word embeddings [11]. This methodology streamlines the aspect reporting process. Due to time and scope constraints, ranking the n-grams based on relevance could not be done for two reasons:

- n-grams aren't directly compatible with Word2Vec, as Word2Vec models are trained on words, not tuples consisting ofwords

Aspect	Count	Positive	Negative	Pos%	Neg%	Sentiment
because of covid	4239	1167	3072	0.275	0.725	– –
die of corona	1878	381	1497	0.203	0.797	– – –
time of corona	1104	572	532	0.518	0.482	+
the covid crisis	965	257	708	0.266	0.734	– –
the corona crisis	955	365	590	0.382	0.618	–
suffer from corona	605	436	169	0.721	0.279	+ + +
get the coron-avirus	580	164	416	0.283	0.717	– –
people be die	220	38	182	0.173	0.827	– – –
amidst the corona	152	93	59	0.612	0.388	+
tell the truth	116	39	77	0.336	0.664	– –
want to travel	110	26	84	0.236	0.764	– – –
stay far away	105	3	102	0.029	0.971	– – – –

TABLE III
3-GRAMS OF COVID-RELATED SENTIMENTS

- Word2Vec only supports words within its vocabulary. "COVID", "coronavirus", and "corona" would either give no data, or meaningless data (coronavirus is a misnomer for COVID-19, as it refers to a type of virus. Tweets in Word2Vec's embeddings would likely not reflect the recent, more topical meaning; corona is slang and has many meanings)

Another scope drawback encountered during this research was lacking the time to cross-compare different demographic identifiers (follower count, "retweets" on a post, account age, etc.) with respect to polarity. Furthermore, Twitter does not provide gender or age information from "Tweets" (age isn't even required to register an account).

In addition to this, the dataset used to train the Naïve-Bayes model consisted of only 5000 positive and negative Twitter posts. As there were hundreds of thousands of collected posts, this could have resulted in a loss of accuracy. *sentiment140*, which consists of of 1.6 million Twitter posts, was a prospective dataset, but its usage did not come to fruition due to time constraints.

VI. Conclusions and Future Work

This paper proposed a modified algorithm for aspect-based sentiment analysis of COVID-19 Twitter posts based on existing methodologies. These proposed augments highlighted possible improvements in extraction, aspect generation, and post classification in lieu of noisy platforms such as Twitter. Some next steps proposed are:

- Train/use a larger and/or more recent Twitter post dataset
- Use data mining methods other than Naive Bayes.
- Examine the fastText library, which is touted as a successor to Word2Vec made by *Facebook*'s AI research lab [12]. fastText improves on Word2Vec by viewing words as summations of n-grams (so "apple" could be seen as "ap" + "pp" + "le"), allowing its model to analyze unknown words (and potentially Twitter misspellings)

References

[1] A. Das and A. Kannan, "Discovering topical aspects in microblogs," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 860–871, 2014.

[2] C. Ejieh, "Aspect-based opinion mining of product reviews in microblogs using most relevant frequent clusters of terms," 2016.

[3] J. Samuel, M. Garvey, and R. Kashyap, "That message went viral?! exploratory analytics and sentiment analysis into the propagation of tweets," *arXiv preprint arXiv:2004.09718*, 2020.

[4] J. Samuel, N. Ali, M. Rahman, Y. Samuel, and A. Pelaez, "Feeling like it is time to reopen now," *COVID-19 New Normal Scenarios based on Reopening Sentiment Analytics. ResearchGate researchgate. net/publication/341478625*, 2020.

[5] C. Ejieh, C. I. Ezeife, and R. Chaturvedi, "Mining product opinions with most frequent clusters of aspect terms," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 546–549, 2019.

[6] Said Özcan, "Preprocessor."

[7] Bird, Steven and Loper, Edward and Klein, Ewan, "Natural language toolkit."

[8] A. Esuli and F. Sebastiani, "Sentiwordnet: a high-coverage lexical resource for opinion mining," *Evaluation*, vol. 17, no. 1, p. 26, 2007.

[9] S. Ting, W. Ip, and A. H. Tsang, "Is naive bayes a good classifier for document classification," *International Journal of Software Engineering and Its Applications*, vol. 5, no. 3, pp. 37–46, 2011.

[10] J. Samuel, G. Ali, M. Rahman, E. Esawi, Y. Samuel, *et al.*, "Covid-19 public sentiment insights and machine learning for tweets classification," *Nawaz and Rahman, Md. Mokhlesur and Esawi, Ek and Samuel, Yana, COVID-19 Public Sentiment Insights and Machine Learning for Tweets Classification (April 19, 2020)*, 2020.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[12] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.