Cluster Analysis of Tweets of World Leaders to Compare Popularity of Topics Among World Leaders and the Efficacy of Different Clustering Methods

Group 11
Max Niebergall 160623100
Amardeep Sarang 160112080
Daniel Berezovski 160173040
Omid Ghiyasian 131462250
Anthony Sukadil 160593610

CP421 Group Research Project (Option B)
Professor Yang Liu
April 3rd 2019

Abstract

In taking up this topic we set out to determine if we could find effective methods of clustering english language Tweets of world leaders by their topics, as well as to compare the efficacy of different clustering algorithms in this task. We believe this is an important endeavour as Twitter is one of the largest social media platforms in the world, and many of the most influential and powerful people in the world use it to spread their messages. It is self evident that knowing the most popular topics among world leaders is useful and applicable knowledge for activists, political rivals, and lobbyists alike. Knowing which methods of discerning these topics are better than others is therefore of equal importance.

Our preliminary research lead us to the clustering methods k-means, LDA, and DBSCAN. In comparison of these methods DBSCAN did the worst at achieving our goal as it clustered our entire dataset, i.e. all of our tweets, into a single cluster. Otherwise, we found LDA to be the most useful clustering algorithm. As a soft clustering algorithm, LDA allowed us to get a better understanding of what it determined to be the topic of the cluster by giving us a list of words and weights for each cluster. Finally, k-means performed reasonably well since the tweets we found in each cluster did appear to have semantic similarities. However, our k-means results did not resemble a topic as closely as our LDA results.

I. Introduction

Twitter is one of the leading social media platforms in the world [1]. It offers microblogging as a service to let users publicly or privately post short messages known as Tweets. Almost every leader of every nation has a verified Twitter account and like other users with large followings, the topics they are interested in are also interesting to those who follow them, and therefore, there is no question that Twitter can be a powerful tool for leaders to influence their audience [2]. It is interesting and important for us to figure out what specific issues are popular among our leaders for this reason. In this report, we will try to find out which clustering method works best for classifying Tweets which share the same topic and to identify and group these topics.

On Twitter, users can post a Tweet almost from any mobile device or computer. Users can post tweets up to 280 characters talking about their opinions, sharing news, politics, economics, viral videos, and so on. In this way a Tweet is usually different from the conventional text data like full page blogs, news articles, journals, and etc. because of the limitation on how many characters is in the Tweet. This results in messages which are informal and short, which makes using algorithms that are typically used for conventional text much less effective [3], [4], [5].

Because of this restraint on message length, we have decided to use the density-based DBSCAN algorithm due to its ability to identify arbitrary shaped clusters, as well as because it is less susceptible to noise and outliers, and because it does not require the specification of number of expected clusters in the data [6]. This seems fitting because we are trying to find popular topics and we don't necessarily know how many topics there are. Additionally, we selected LDA as our second clustering method due to positive results in similar studies [7]. Finally we are also considering k-means clustering method due to its popularity and ability to detect topics on large amounts of information automatically [8], [9].

Throughout this paper, we will briefly introduce related works, explain how we collected the data, what pre-processing was done, which leaders we selected for this study and the application of the clustering methods mentioned here. Finally we will provide experimental results to verify the performance of each method and give a conclusion with suggested future works.

II. Related Works

Social media platforms such as Twitter generates terabytes of information every day [11]. On Twitter, users share short text messages with other users of the platform. In order to extract topics out of this kind of dataset, we need algorithms that can deal with short informal documents such as Tweets [12].

Applying clustering methods to documents generated on social media platforms is not an original idea on our part; it has been actively researched well before us. Wartena, Cutting and others [8] [10] have proposed clustering algorithms to organize and cluster topics in large documents such as wiki pages on Wikipedia and other large document collections online. However these conventional document clustering algorithms often do not work for short and informal texts. Wikipedia hardly follows a trend in Twitter where things evolve quickly. Wikipedia also does not contain informal messages like that of Twitter.

That being said there have been many trials to cluster these kinds of data. To name a few, Zhang and others [13] proposed a modified LDA

method to improve the performance of topic detection which found that the method shows promising results compared to all other baseline methods. Other works [9], [14] and [15] propose some variant of k-means or other methods that were compared to k-means in order to measure the performance with varying results. In summary, many papers either used a modified version of k-means or compared their own algorithm to k-means. A comparison of several methods was done by Ibrahim and others [16] which included DBSCAN as one of the proposed methods. The results show that DBSCAN is performs poorly compared to other clustering methods. In this paper, we will try to implement some of these methods and compare our results to verify the outcome.

III. Dataset & Pre-Processing

In this section, we describe how we obtained our dataset, what it looks like, the challenges, and how we processed it.

To obtain the data, we used Tweepy which is an easy-to-use Python library used to access the Twitter API. To begin, we specified which politicians we want to grab tweets from. Tweet data is set up as a JSON object consisting of many fields including the language, the date, was it a retweeted tweet, links, any hashtags, mentions and many more. For our models, we parse the JSON objects to extract the language and full text of the tweet which included links, hashtags, unicode characters and punctuation. We selected language because some politicians retweet in multiple languages which adds another layer of complexity.

When it came to pre-processing, we did this process differently for the three methods. For DBSCAN and k-means clustering methods we applied the same pre-processing methods to our dataset. The main aspect of this pre-processing method was the use of the Doc2Vec library from Gensim. According to Gensim, “The algorithms in Gensim . . . automatically discover the semantic structure of documents by examining statistical co-occurrence patterns within a corpus of training documents. These algorithms are unsupervised, which means no human input is necessary – you only need a corpus of plain text documents.”[2.1] We used trained Doc2Vec on our data set, and to produce a Document-Vectors for each tweet. We also used Doc2Vec to set the min-count to 2, eliminating some uncommon words, and to set the dimensionality (vector size) of our vectors to 500. result of this use of Doc2Vec was a data set of vectors, each with 500 dimensions. These vectors were then passed directly into our clustering algorithms.

Unlike the methods used by Doc2Vec in k-means, the LDA model cannot handle difficult text, notably, emojis, links, and other unicode characters. Because of this, we start by removing unicode characters and punctuation as they do not provide any useful information. After this, we parse each tweet and remove stopwords. This is done through Gensim’s list of stopwords as well as a large external list of 3000 stopwords that is also run through the dataset to reduce dimensionality. After this, we stem each token to reduce it to its root word, eg. community to commun. The pre-processed data is then passed in the LDA model.

We now will describe our procedure with each clustering method used.

IV. K-Means

First, k-mean is a clustering method that tries to separate the data into k clusters, where k is specified by the user. It does this while trying to minimize the distance to the center of each cluster.

In our experiment, we used the k-means method from the scikit-learn library. The main

parameter that we changed was the k-value (ie. the number of clusters). Further justification for why we choose this method can be credited to the popularity of the method and as mentioned before, the popularity of k-means is likely due to the fact that it can handle large dimensionality data. It is also good for topic modeling since you can specify how many clusters you require, which is especially useful when the number of topics in the data is known. Furthermore, the fact that k-means has a complexity of $O(n^2)$ adds to its appeal for topic clustering.

To start the implementation, we took the pre-processed Doc2Vec vectors and stored them in a PKL file. After this, we simply loaded the vectors from the PKL file and created a numpy array of the vectors. Afterwards, we defined our k-value to be 25 and used scikit-learn's k-means function.

V. LDA

Probabilistic models of text and topics are powerful tools in making inferences about the content of documents. Latent Dirichlet allocation (LDA) is one of the most important probabilistic models used today and is a method of soft clustering in natural language processing [17]. Soft clustering is important in document classification because documents can be classified under multiple themes or topics. As we've seen in the previous model, hard clustering of documents is not ideal as many documents can overlap in topic. The model returns a set of words, each with a weight associated with it which determines the significance of the word in that cluster.

The intuition behind the model stems from the idea that documents cover only a small set of topics and these topics use only a small set of words frequently. For example, Tweets that contain words such as "finance", "helping", "community", "Canadians", and "growing" have a larger commonality in Canadians and growth, than other subjects, which suggests a topic focused around the Canadian economy doing well. In theory, this results in a more precise assignment of documents to topics. However, one disadvantage of LDA is that it is not able to derive context because the pre-processing step removes most of this. When looking at Tweets of politicians specifically, the platform is used to share news with a limit on characters. This means, users want to fit as much information in as few words as possible, focusing heavily on visuals, links, and emojis to emphasize meaning.

To start the implementation, we begin with the pre-processing steps by stripping unwanted characters, lemmatizing and then stemming. We then use Gensim to create a dictionary, which "encapsulates the mapping between normalized words and their integer ids."[2.1] We then take all our pre-processed text data and convert it into a bag-of-words format using doc2bow and this becomes our corpus. This turns every tweet into a list of tuples indicating the number of instances of a token in a tweet which will be used in the model. After this, we use the LDA function in Gensim and proceeded with analyzing our findings.

VI. DBSCAN

DBSCAN is a density-based clustering method. scikit-learn describes DBSCAN as a "Density-Based Spatial Clustering of Applications with Noise. [It] Finds core samples of high density and expands clusters from them. Good for data which contains clusters of similar density." This implementation has two main parameters. First, eps is the maximum distance between two samples for them to be considered as in the same neighborhood. Second, the number of samples in a

neighborhood for a point to be considered as a core point.

For DBSCAN, we used the same Doc2Vec pre-processing used is the k-means method. We decided to use DBSCAN due to a number of factors, including that it does not require a predetermined number of clusters, its ability to make clusters of arbitrary shapes, and its robustness towards outliers. It was also appealing that DBSCAN had an average efficiency of $O(n \log n)$.

It should also be noted that DBSCAN is a hard clustering method which means that it will strictly assign the tweet to a single cluster rather than assigning a probability of the tweet is in the cluster like soft clustering would.

The implementation used the same Doc2Vec vectors used in k-means and used scikit-learn's implementation of DBSCAN.

VII. Results

Overall, our primary evaluation method was qualitative. We looked at the clusters that were made by each technique and which words were present in each cluster. An indication of a well performing clustering technique would be given when a cluster made by that technique had several words which related to a single high level topic. For example seeing words like 'Canada', 'children', 'family', 'community', and 'achieve' in a single cluster might indicate that the topic of the cluster is well defined and about Canadian people. However, if the cluster also had words like 'climate', 'healthcare', and 'business', it is less clear what the topic is, but would indicate that this method was at least somewhat successful at clustering topics.

Using this method of evaluation, we were able to conclude that LDA gave the best results since several clusters derived by LDA contained words and weights which indicated a single overarching topic[1]. However, we found that as a result of the pre-processing step required to properly train the model, clusters have many overlapping words, as well as words that are difficult to interpret. For example, we see 'ha' and 'wa' appearing frequently in clusters with relatively high weights which don't represent any important meaning. We also have the names of the politicians appearing frequently in each cluster with high weights. However, this does not necessarily mean that this is the topic of the cluster, as figures like Justin Trudeau appear in Tweets about many topics. Overall, the benefits of LDA are that we are given a list of words and their weights, and we can generally identify a small set of topics.

We also concluded that DBSCAN performed exceptionally poorly as it clustered all tweets into a single cluster and thus it failed to split tweets into clusters which each had a recognizable topic.

The k-means method did split the tweets into our specified 25 clusters, however, most clusters formed contained a set of words from several topics which we classified as an only satisfactory result. One of the disadvantages of k-means is the fact that like DBSCAN it is a hard clustering method hard which means that it will strictly assign each tweet to a single cluster rather than assigning a probability of the tweet is in the cluster like soft clustering would. k-means also works well only on globular clusters, since the data had so many dimensions it was impossible to visually confirm if the data is globular. If this was not the case, it could have led to a negative impact on the performance of k-means. Likely due to these disadvantages, k-means did not perform exceptionally well. For example, while a k-means cluster may have contained words belonging to the topic of Canada such as 'trudeau' and 'canada' it would also contained words belonging to the topic of the U.S. such as 'trump', 'america', and 'congress'. Furthermore,

it was not evident from the other words in the cluster that these two topics were actually related in an overarching topic. This mixing of different topics in a single cluster means that k-means did not perform as well as LDA. However since it was able to create clusters that indicated some common themes, it is considered to have a better performance than DBSCAN.

Although the ability of a technique to get a logical and correct answer is one of the main metrics considered when evaluating clustering methods such as the ones used, it should not be the only consideration. Additionally, one should also consider the type of clustering (soft or hard), the ability to control the number of clusters, as well as the time and space efficiency of the clustering algorithm.

Firstly, we believe that soft clustering is superior in this application because it's ability to assign a tweet to multiple topics and assign weights based on similarity to the topic allowed for a clearer understanding of the meaning of each cluster. The reason for this is likely due to the fact that DBSCAN does not work well with high dimensionality data or sparse data which was exactly what our data entailed. A second problem that was encountered, which was likely due to the first was the fact that it was hard to estimate the eps parameter since the dataset we used so many dimensions. Because DBSCAN is a hard clustering method, it was not a great method for clustering tweets because a tweet could be in multiple clusters (topics) and hard clustering methods like DBSCAN are unable to reflect this. Thus for all the reasons above it can be concluded that DBSCAN is unsuited for the task of clustering tweets into topics.

A second consideration when evaluating clustering methods is whether one can specify the number of clusters to be made such as in k-means. This ability can be desirable in topic clustering especially when the number of topics present in the data is well known. Although k-means did not give the best performance for our purposes its ability to create a specific number of clusters may be one of the reasons why it is popular in many topic clustering papers that we reviewed. However, using clustering methods that automatically choose the number of clusters can also have undesirable results such as clustering all data points in a single cluster. This was the main downfall of DBSCAN in our experiment.

One final consideration when evaluating clustering methods is time and space complexity. Neither time nor space complexity was a problem for us since our dataset had a modest number of tweets and would finish in under 30 seconds on a machine with 8GB of system memory. However, this may be a concern in larger data sets. k-means and DBSCAN both have a time complexity of $O(n^2)$ where n would be the number of records in the data set. In comparison LDA has a time complexity of $O(nd^2)$, where d is the dimensionality, this means that in datasets were there is a high dimensionality k-means and DBSCAN is more efficient than LDA. Despite this for our purposes LDA's better performance was worth the trade off in complexity.

All in all, LDA gave the best results. Although k-means did split the file into clusters where there were words belonging to certain topics like the topic of Canada such as 'trudeau' and 'canada' it would also contain words belonging to the topic of the U.S. such as 'trump', 'america', and 'congress'. This mixing of different topics in a single cluster meant that the performance of k-means was less than optimal and LDA performed better. Of course by producing only one cluster for our entire data set, we take DBSCAN out of the running.

VIII. Conclusion

In taking up this topic we set out to determine if we could find effective methods of clustering english language Tweets of world leaders by their topics, as well as to compare the efficacy of different clustering algorithms in this task. Our preliminary research lead us to the clustering methods k-means, LDA, and DBSCAN. Of these methods DBSCAN, did the worst at achieving our goal as it clustered our entire data set, all of our tweets, into a single cluster. Otherwise, we found LDA to be the most useful clustering algorithm. As a soft clustering algorithm it allowed us to get a better understanding of what it determined to be the topic of the cluster by giving us a list of words and weights for each cluster. Finally, k-means performed reasonably well as we were able to select the number of clusters, and the tweets we found in each cluster did appear to have semantic similarities. However, our k-means results did not resemble a topic as closely as our LDA results, so we can conclude that LDA was superior on our testing.

In future, we would likely want to increase the variety of Twitter users, as well as to change the number of tweets per user in our data set. A more diverse input might result in clusters which are more semantically different and would give us clearer results. Additionally, we might want to change the number of clusters generated by K-Means and LDA. The number of clusters is supposed to represent the number of topics among our tweets. If we were able to select a number of clusters which more closely approximates the true number of topics we would likely get better results. We might want to adjust other parameters as well, such as the min-count and vector-size in Doc2Vec.

IX. References

[1]. Alhabash, Saleem, and Mengyan Ma. “A Tale of Four Platforms: Motivations and Uses of Facebook, Twitter, Instagram, and Snapchat Among College Students?” *Social Media + Society*, Jan. 2017, doi:10.1177/2056305117691544.

[2]. Parmelee, John H., and Shannon L. Bichard. *Politics and the Twitter revolution: How tweets influence the relationship between political leaders and the public*. Lexington Books, 2011.

[3]. O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. Pages 46-54, 1998.

[4]. O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to Web search results. Computer Networks: The International Journal of Computer and Telecommunications Networking. 31(11):1361-1374, 1999.

[5]. J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi. Simple semantics in topic detection and tracking. Information Retrieval, 7(3-4):347-368, 2004

[6]. Baralis, Elena et al. “Analysis of Twitter Data Using a Multiple-level Clustering Strategy.” *MEDI* (2013).

[7]. Huang, Bo, et al. "Microblog topic detection based on LDA model and single-pass clustering." *International Conference on Rough Sets and Current Trends in Computing*. Springer, Berlin, Heidelberg, 2012.

[8]. Wartena, Christian, and Rogier Brussee. "Topic detection by clustering keywords." *2008*

19th International Workshop on Database and Expert Systems Applications. IEEE, 2008.

[9]. Nur'aini, Khumaisa, et al. "Combination of singular value decomposition and k-means clustering methods for topic detection on Twitter." *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, 2015.

[10]. Cutting, Douglass R., et al. "Scatter/gather: A cluster-based approach to browsing large document collections." *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1992.

[11]. Huberman, Bernardo A., Daniel M. Romero, and Fang Wu. "Social networks that matter: Twitter under the microscope." *arXiv preprint arXiv:0812.1045* (2008).

[12]. Kim, Sungchul, et al. "Finding core topics: Topic extraction with clustering on tweet." *2012 Second International Conference on Cloud and Green Computing*. IEEE, 2012.

[13] Zhang, Chengde, et al. "A Novel Hot Topic Detection Framework With Integration of Image and Short Text Information From Twitter." *IEEE Access* 7 (2019): 9225-9231.

[14]. Inouye, David, and Jugal K. Kalita. "Comparing twitter summarization algorithms for multiple post summaries." *2011 IEEE Third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*. IEEE, 2011.

[15]. Cheong, Marc, and Vincent Lee. "A study on detecting patterns in twitter intra-topic user and message clustering." *2010 20th International Conference on Pattern Recognition*. IEEE, 2010.

[16]. Ibrahim, R., Elbagoury, A., Kamel, M.S. et al. Knowl Inf Syst (2018) 54: 511. https://doi.org/10.1007/s10115-017-1081-x

[17] Sontag, David, and Dan Roy. "Complexity of inference in latent dirichlet allocation." *Advances in neural information processing systems*. 2011.

X. Appendix

1. In one LDA cluster, the top weights and words were [‘canadian’ (0.0432), american’ (0.0363), ’women’ (0.021) , ‘trudeau’ (0.017), ‘famili’ (0.013), ‘cdnpoli’ (0.011), ‘trump’ (0.01), ‘healthcare’ (0.008), ‘opportun’ (0.006), ‘commun’ (0.005)]
2. Libraries
    1. Gensim
    2. scikit-learn
    3. Tweepy
    4. Pandas