USING NATURAL LANGUAGE PROCESSING TO CREATE REFERENCE TERM VECTORS: TF-IDF VS DOC2VEC

Shane J. Robinson[1, 2, †]

[1] Northwestern University
339 East Chicago Avenue, Chicago, IL 60611

[2] github.com/shanergit

[†] Address to which correspondence should be addressed:
shaner.mail@icloud.com

Abstract

One of the primary challenges in using natural language processing to analyze documents is choosing which algorithm best characterizes the important terms therein by producing a distillation of the data that's referred to as a Reference Term Vector (RTV). Term Frequency-Inverse Document Frequency (TF-IDF) is a legacy text vectorization algorithm that directly maps terms from a "bag of words" vector into a new reduced size TF-IDF vector. Doc2Vec is a more modern text vectorization algorithm that also produces a reduced size vector but was developed in 2013 by Tomas Mikolov and team to handle large corpora and big data. In this paper, we will illustrate how TF-IDF yields better results for a corpus of contained size, with homogeneous content that is able to be mentally modeled; whereas Doc2Vec yields better results for boundless corpora of varied content that is too broad for devising a mental model.

1 Introduction

Data analysis tools have long been utilized within U.S. politics, especially in field of public polling – candidate preference polling, issues polling, exit polling, etc. But in the modern era of exabytes of collectable data, actors in the political arena could be making use of NLP algorithms to better understand issues by distilling documents to the words/phrases that best characterizes their content. In so doing, political stakeholders would be equipped to better position candidates and more accurately frame and discuss issues.

Choosing between the TF-IDF and Doc2Vec algorithms is an important consideration when creating a reference term vector for corpora. In this effort to illustrate that the TF-IDF algorithm produces a more efficacious RTV than does Doc2Vec for the sizes of corpora used in political issue analysis, our team of nine researchers has produced a corpus of eighteen documents, two from each researcher, related to political matters faced by the Biden administration. These documents were trimmed to ~500 words, for the sake of producing a manageable document to be analyzed under time constraints, and then combined to create the corpus. Within this text, we will demonstrate that TF-IDF creates a more constructive reference term vector than does Doc2Vec for this corpus.

2 Related Work

Over the years, several word vectorization methods have been presented, as well as extensions of existing vectorization algorithms. A 2003 paper examined the results of using TF-IDF to determine which words in a corpus would be appropriate to use in a query, and provided evidence that the algorithm categorizes relevant words effectively and can bolster query retrieval (Ramos, 2003). A seminal moment for reference term vectors came when Word2Vec was created to produce a numeric representation for each word in a set using the Continuous Bag-of-Words model, Eq. (1):

$$\mathbf{Q = ND + Dlog_2V}$$

and the Continuous Skip-Gram model, Eq. (2):

$$Q = C(D + D\log_2 V)$$

where C is the maximum distance of the words (Mikolov et al, 2013). Doc2Vec (document vectors) is an extension of Word2Vec, which is an unsupervised algorithm created to learn fixed-length feature representations from variable-length pieces of texts or strings of words together like sentences, paragraphs, and documents (Thanaki, 2017). This algorithm makes up for the two weaknesses of the bag-of-words model, namely, the loss of the ordering of words and ignoring of the semantics of words (Mikolov et al, 2014). In a 2020 video, TF-IDF was compared with Doc2Vec, and it was posited that TF-IDF is the superior algorithm for creating a reference term vector from smaller corpora, whereas Doc2Vec produces a superior RTV for immensely sized corpora due to its dimensionality reduction (Maren, 2020). In a 2019 paper, co-training with TF-IDF/LDA and Doc2Vec/LDA was conducted to illustrate co-training as a semi-supervised learning (SSL) approach that aims to utilize various perspectives in respect of feature subsets for the same example (Kim et al, 2019).

3 Data

Our team of nine researchers each contributed two articles to include in the corpus. The parameters for articles selection were that they be from 2016 or later, and that they be related to issues the Biden administration is facing, either domestically or internationally. Upon selection, each article was preprocessed before being included in the corpus. This preprocessing included trimming each to ~500 words and cleansing them of stop words such as "the," "is," "are," etc. Other items that were filtered out of the articles were punctuation, short tokens, and non-alphabetics; each word was also converted to lowercase. Preparing the documents in this way allowed for a straightforward corpus creation within the Python programming language and filtered out words/tokens that were not useful to the algorithms. Finally, the corpus text was processed into lists and each document was given a file name in researcher_docnumber_description format for quick identification purposes while working with the corpus and the RTV outputs.

4 Methods

We are comparing the TF-IDF and Doc2Vec techniques for creating reference term vectors and demonstrating how TF-IDF is the preferable algorithm for use with smaller corpora, whereas Doc2Vec is better suited to immensely sized corpora. The results of each algorithm's resultant RTV will be compared with mentally mapped clusters – this is possible due to the corpus's limited size. Passes through three algorithms were conducted – K-means TF-IDF, LDA TF-IDF, and Doc2Vec – with numerous subsequent iterations for fine-tuning, in which parameters such as vector size and number of topics were adjusted to optimize RTV results.

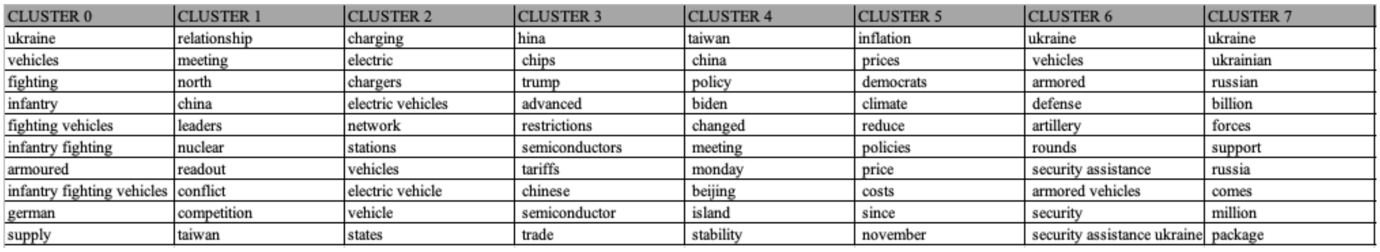
For our initial pass of K-Means TF-IDF, K was set to 8 and random state equal to 89. We then sorted cluster centers by proximity to centroid, saved the terms for each cluster and document to dictionaries to be used for plotting outputs, and created dictionaries to store terms and titles.

After an initial pass of K-means TF-IDF indicated fewer clusters were appropriate to categorize topics in the corpus, an LDA using TF-IDF algorithm was passed with the number of topics set to 5. While earlier passes of the TF-IDF algorithm variations (as well as initial mental modeling prior to running through code) indicated fewer than eight topics, our team found that Doc2Vec produced its best vectorization with K set to 8.

5 Results

5.1. Running K-means on the TF-IDF Matrix

The initial results from running the corpus through the K-means TF-IDF algorithm yielded eight clusters. It successfully produced clusters grouping documents together by the following content: electric vehicles, US/China semiconductors, aid for Ukraine, and Russia/Ukraine.

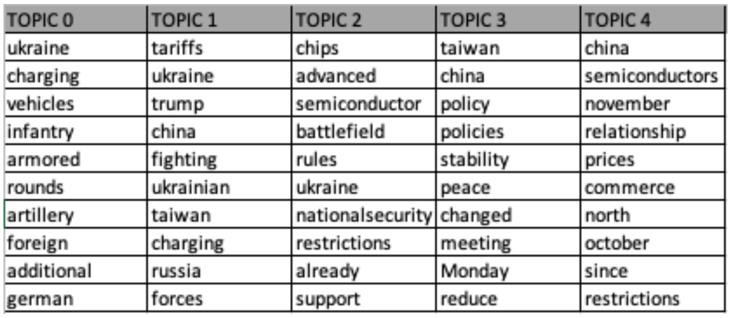


CLUSTER 0	CLUSTER 1	CLUSTER 2	CLUSTER 3	CLUSTER 4	CLUSTER 5	CLUSTER 6	CLUSTER 7
ukraine	relationship	charging	hina	taiwan	inflation	ukraine	ukraine
vehicles	meeting	electric	chips	china	prices	vehicles	ukrainian
fighting	north	chargers	trump	policy	democrats	armored	russian
infantry	china	electric vehicles	advanced	biden	climate	defense	billion
fighting vehicles	leaders	network	restrictions	changed	reduce	artillery	forces
infantry fighting	nuclear	stations	semiconductors	meeting	policies	rounds	support
armoured	readout	vehicles	tariffs	monday	price	security assistance	russia
infantry fighting vehicles	conflict	electric vehicle	chinese	beijing	costs	armored vehicles	comes
german	competition	vehicle	semiconductor	island	since	security	million
supply	taiwan	states	trade	stability	november	security assistance ukraine	package

The algorithm was able to drill down to produce a granular cluster: military aid for Ukraine. On first pass, however, not every article in the corpus with content related to military aid for Ukraine was included in this cluster – three total clusters contained such articles. Further adjustment of the code should rectify this issue and more accurately bin the data. Thusly, we hypothesized fewer than eight clusters should be generated to appropriately vectorize the corpus.

5.2. Running LDA on the TF-IDF Matrix

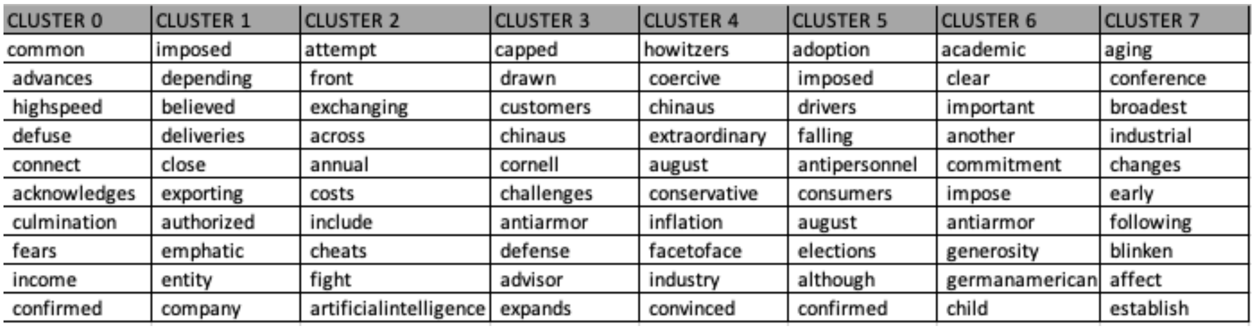
This algorithm yielded suboptimal results, with the only somewhat accurate vector being a cluster that grouped any foreign-to-the-US topics. Even with that, the topics are muddy with documents about electric vehicles and military aid to Ukraine included.



TOPIC 0	TOPIC 1	TOPIC 2	TOPIC 3	TOPIC 4
ukraine	tariffs	chips	taiwan	china
charging	ukraine	advanced	china	semiconductors
vehicles	trump	semiconductor	policy	november
infantry	china	battlefield	policies	relationship
armored	fighting	rules	stability	prices
rounds	ukrainian	ukraine	peace	commerce
artillery	taiwan	nationalsecurity	changed	north
foreign	charging	restrictions	meeting	october
additional	russia	already	Monday	since
german	forces	support	reduce	restrictions

5.3. K-means Clustering Doc2Vec

While earlier passes of the TF-IDF algorithm variations (as well as initial mental modeling prior to running through code) indicated fewer than eight topics, our team found that Doc2Vec produced its best vectorization with K set to 8.



CLUSTER 0	CLUSTER 1	CLUSTER 2	CLUSTER 3	CLUSTER 4	CLUSTER 5	CLUSTER 6	CLUSTER 7
common	imposed	attempt	capped	howitzers	adoption	academic	aging
advances	depending	front	drawn	coercive	imposed	clear	conference
highspeed	believed	exchanging	customers	chinaus	drivers	important	broadest
defuse	deliveries	across	chinaus	extraordinary	falling	another	industrial
connect	close	annual	cornell	august	antipersonnel	commitment	changes
acknowledges	exporting	costs	challenges	conservative	consumers	impose	early
culmination	authorized	include	antiarmor	inflation	august	antiarmor	following
fears	emphatic	cheats	defense	facetoface	elections	generosity	blinken
income	entity	fight	advisor	industry	although	germanamerican	affect
confirmed	company	artificialintelligence	expands	convinced	confirmed	child	establish

6 Discussion

Much time was spent on mentally modeling the corpus as its small size lent itself well to this mental preprocessing. By modeling the clusters, we believed the documents should bin into, we were able to adjust the “K” in the algorithm through our iterations. We settled on five: US/China semiconductors, China/Taiwan engagement, domestic fiscal policy, electric vehicles, and military aid for Ukraine. We initialized this process with eight for the first pass of TF-IDF to create a reference point from which to begin fine-tuning.

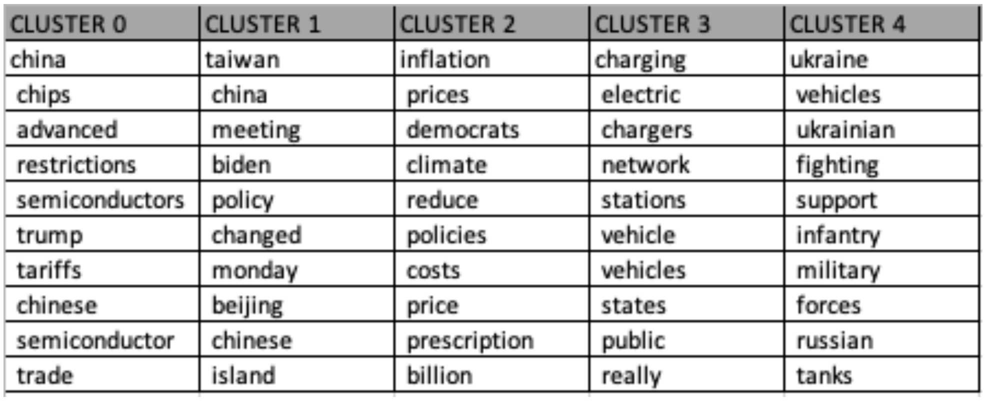
After the first pass of K-means TF-IDF, we transitioned to an LDA run. As this was still using TF-IDF, we again settled on five for the number of topics. This produced a result not as accurate as the K-means run but still with some fairly accurate clusters, as the algorithm successfully grouped documents about Ukraine fighting vehicles and US/China semiconductors. The remaining three clusters were not as coherent, however, and the weights for each word in the five topics indicated further tuning would be necessary to have this algorithm produce an RTV as accurate as the K-means TF-IDF – nearly all the words had a weight of .001, with few exceptions of .002 and one instance of .003. It proved to be more cumbersome to wrangle LDA for this corpus than was K-means on TF-IDF.

The Doc2Vec run failed to produce any clusters that accurately grouped the documents, and it failed to identify all the important words for each document that TF-IDF identified – there were no overlapping words in the competing reference term vectors. The many-to-many mapping nature of Doc2Vec proved too complex for a small corpus.

7 Conclusions

The resulting reference term vectors for each of the three algorithms clearly indicate a hierarchy of optimal outputs. The least characteristic RVT for the corpus was produced by Doc2Vec, as regardless which parameters were chosen and adjusted, it did not adequately cluster the documents nor accurately predict the important words in the clusters. This was to be expected, as Doc2Vec is better suited to immense corpora.

While LDA is a more modern algorithm than simple K-means, running LDA on the TF-IDF matrix yielded weak results. It was only able to produce one somewhat accurate cluster containing the foreign policy-related documents. Our team of researchers found the K-means TF-IDF algorithm to be the optimal choice for creating a reference term vector for a corpus of this volume and subject matter. As illustrated below, the five clusters produced accurately grouped the documents by topic, with minimal instances of unimportant words as part of the vectorization (see "really" in Cluster 3 as an example).



CLUSTER 0	CLUSTER 1	CLUSTER 2	CLUSTER 3	CLUSTER 4
china	taiwan	inflation	charging	ukraine
chips	china	prices	electric	vehicles
advanced	meeting	democrats	chargers	ukrainian
restrictions	biden	climate	network	fighting
semiconductors	policy	reduce	stations	support
trump	changed	policies	vehicle	infantry
tariffs	monday	costs	vehicles	military
chinese	beijing	price	states	forces
semiconductor	chinese	prescription	public	russian
trade	island	billion	really	tanks

This RVT was the most accurate produced of all the algorithms and could serve political actors well in identifying the most important words for each document, standing as proxy for an issue; and aiding in the positioning of candidates, the framing of issues, and navigating to solutions.

8 Future Work

There are several tunings our team would implement in future work with the TF-IDF algorithm for this corpus and others of its ilk, once of which is creating more granular equivalence terms for singular/plural forms of words. We would also adjust our stop words filtering to include adjective/adverb tokens, eliminating words such as "really" and "public." A cleaner corpus may lead to a more accurate reference term vector.

References

Ramos, Juan. "Using TF-IDF to Determine Word Relevance in Document Queries." In *Proceedings of the First Instructional Conference on Machine Learning*, vol. 242, no. 1, pp. 29-48. 2003.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space." *arXiv preprint arXiv:1301.3781* (2013).

Thanaki, Jalaj. "Advanced Feature Engineering and NLP Algorithms." In *Python Natural Language Processing*. Packt Publishing, 2017.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. "Distributed Representations of Words and Phrases and Their Compositionality." *Advances in Neural Information Processing Systems* 26 (2013).

Maren, Alianna J. “NLP: TF-IDF vs Doc2Vec – Contrast and Compare.” YouTube video, 9:28. February 23, 2020. https://www.youtube.com/watch?v=iSkbq6Tjkj0

Kim, Donghwa, Deokseong Seo, Suhyoun Cho, and Pilsung Kang. "Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec." *Information Sciences* 477 (2019): 15-29.