Clustering For Classification Using KMeans With Term Frequency-Inverse Document Frequency And Latent Dirichlet Allocation

Seth Gory
sethgory@gmail.com

I. Introduction

This paper explores the utilization of clustering techniques as a means of dimensionality reduction performed in preparation for a classification task. A collection of stand-up comedy transcripts and their corresponding Internet Movie Database (IMDb) ratings were scraped from the web and used as feature data and targets respectively. Latent Dirichlet Allocation (LDA) is a widely-used NLP technique and was chosen to perform topic modeling [1][2]. The results of modeling returned a vector of probabilities indicating the distribution of topics within each document in the dataset. A KMeans algorithm was then used to cluster the documents based on each topic vector [3]. KMeans is used again to form a different set of clusters based on Term Frequency-Inverse Document Frequency (TF-IDF) matrices computed for each document [4]. Quality of each application of the clustering algorithm is determined by silhouette score [5]. Comparisons of topic distribution, comedian names, years, and IMDb rating were also made between each set of clusters.

The accurate prediction of IMDb rating based on transcript text data would be of interest to any media company or producer who is looking to maximize their investment in a production by being able to forecast the relative success of that production. Being able to predict a crowd-sourced rating could be akin to predicting the universality or accessibility of a particular comedian's performance style. This could prove useful in deciding which shows to host on a company's media platform if the goal is to create content that will earn high marks on IMDb and thus cater to the largest market. Random Forrest, Stochastic Gradient Descent, XGBoost, and an ensemble method were tested in a classification task to predict a binary representation of IMDb rating [6][7][8][9]. Derived features, including cluster assignments and topic distributions, were used to train the models in each of three rounds of testing with varying results.

The rest of this document contains sections that detail the entire process of analysis and all findings of the work. Section II describes the dataset and the scraping process that was used to build it. Steps taken to preprocess the text and perform feature engineering are also discussed. Topic modeling, clustering, and classification models are described in section III. Section IV provides a detailed analysis of the results of clustering and predictive modeling. Conclusions, weaknesses, and future directions are discussed in section V.

II. Data Description

The website Scraps From The Loft hosts transcripts and other resources relating to media and pop culture. In total, there were 330 transcripts collected at the time the script was run for this project's application. Additional data such as runtime and IMDb rating are also gathered for the initial dataset (Table 1). Out of the 330 total transcripts obtained, 272 titles had IMDb data available. Some descriptive statistics about the data are given in Table 2.

Binary labels, indicating above or below average IMDb rating, were created and assigned to each data point that had a rating value (Fig 1). This attribute was used as a target to train classifier models later on in the project. The technique of binning or discretizing continuous attributes, while yielding coarser results, can help to decrease the amount of complexity needed in forming a decent predictive model [10].

TABLE I. Data Description

Attribute	Type	Example Value	Description
TITLE	Nominal (string)	"23 Hours To Kill"	Title of special
NAME	Nominal (string)	"Jerry Seinfeld"	Name of comedian
YEAR	Numeric (integer)	1992	Year the special was filmed
IMDB RATING	Numeric (real)	7.6	IMDb rating given for a particular special

Attribute	Type	Example Value	Description
RUNTIME	Numeric (integer)	72	Length of special in minutes
RATING TYPE	Binary (integer)	1	Discretized ratings as a binary. 1 is above the mean, 0 is below the mean.
LINK	Nominal (string)	"https://www…"	Link to individual transcript pages
TRANSCRIPT	Nominal (string)	"Hello, thanks for coming to the show…"	Transcript of stand-up comedy routine
LANGUAGE	Nominal (string)	"en"	Language of the routine

TABLE II. DESCRIPTIVE STATISTICS

Attribute	Range	Mean	SD	Mode
RUNTIME	20.0 – 123.0	68.26	14.85	60.0
IMDB RATING	2.9 - 8.8	7.30	0.90	7.4
YEAR	1963 - 2020	2011	10.24	2018
WORD COUNT	230 - 7396	3819	1540.64	3834
DIVERSITY RATIO[a]	0.18 – 0.80	0.36	0.10	0.52

a. Unique words / Total word count.

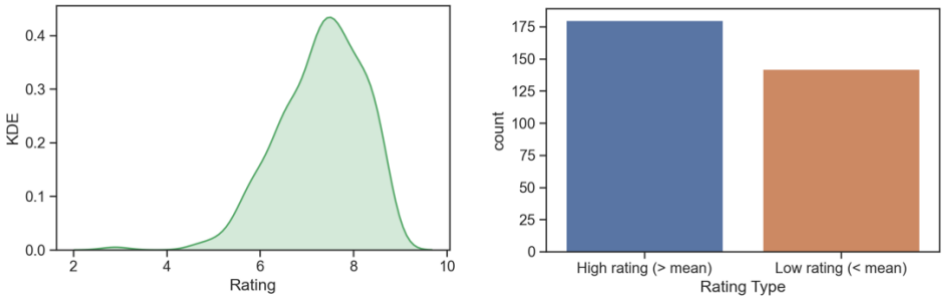

Fig. 1. Kernel Density Estimation plot for Rating (left). Bar plot of counts of each rating category (right).

III. METHODOLOGY

The process of feature preparation and analysis is outlined in Fig 2 and includes the following steps: data collection, cleaning, clustering, predictive modeling.

A. Data Collection

The transcripts were gathered by applying the Python libraries BeautifulSoup and Requests. A directory page was first scraped in order to gather the title and link attributes [11]. Each iteration of a custom scraping algorithm then visited a link that pointed to the transcript and scraped the contents. Google Chrome Dev Tools was used to discover the HTML organization of directory and transcript pages. An IMDb API was used to send requests for runtimes and ratings to the movie information database [12]. Frequent errors occurred when using the entire title as a search term. A workaround was found by instead using only the first 30 characters of each title for the search. The langdetect library was used to determine language of each transcript [13]. Although most of the transcripts were in English, it was necessary to add a language attribute because the few transcripts that were in Italian or Spanish confused several attempts at analysis.

Fig. 2. Flow chart of processes.

B. Cleaning and Feature Engineering

The HTML text that was initially scraped from the directory page included only the title, tags, and link for each special. From the title, regex was used to parse the year, which conveniently always appeared in the format of four digits between parenthesis. Filtering the tags through a short list of removable terms yielded the names. As would be typical for most NLP procedures, the documents were stripped of all punctuation, non-alphabetical characters, and line breaks. Lemmatization and parts of speech tagging and was done using the Python library Spacy and words that were not nouns, adjectives, verbs, or adverbs were filtered out. Optional bigrams and trigrams (common 2 and 3-word sequences) were created using Gensim [14][15][16].

C. Clustering

There were two sets of clusters created, each using a different abstraction of the input data to find distances between documents. LDA was used to form the first abstraction and was selected because of its recent popularity in the topic modeling domain [1]. Each document is assessed for the presence of k topics based on which words are in the text and given a vector that represents how each topic is proportionally present in the document. LDA algorithms detect which words are usually present together and combines them into a defined number of topics that a human can then attempt to interpret. The number of topics could be chosen based on coherence scores [17], but interestingly for this application, more interpretable topics were found when choosing $k = 7$ which produced a low coherence score of 0.36. KMeans was used to group the topic vectors of each document into 7 clusters. For clustering, $k = 7$ was selected by performing a grid search and comparing silhouette scores and the sum of squared errors (SSE), also called inertia, produced by each k (Fig 3).

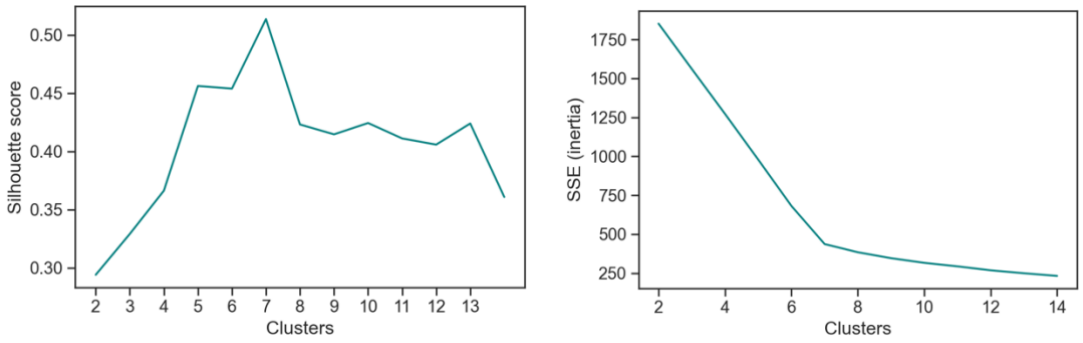

Fig. 3. Clusering strength when using LDA topic vectors as input. Silhouette scores per cluster (left). Plot of SSE per cluter (right).

The second KMeans clustering was performed using a large sparse matrix of TF-IDF scores to represent each document. TF-IDF has long proven useful in the field of NLP for discovering the relevance and importance of certain terms both within a document and within a corpus of documents [18]. A TF-IDF vectorizer from the Sklearn Python library was used to compute a matrix of scores for each term in the document. KMeans was employed to find documents with the least distance between them and make cluster assignments. The number of clusters was chosen to be 7 in light of the fact that silhouette scores and SSE would

not converge to an optimum (Fig 4). By selecting the same number of *k*, comparisons could be made to the clusters formed with LDA topic vectors.

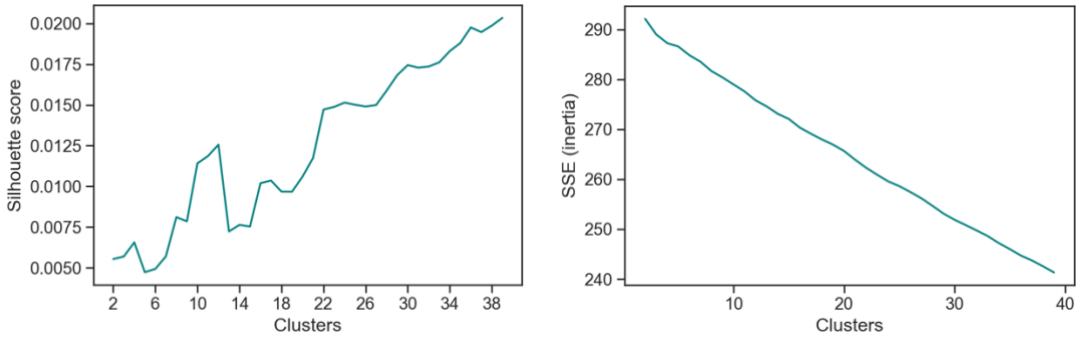

Fig. 4. Clustering strength when using TF-IDF matrices as input. Silhouette scores per cluster (left). Plot of SSE per cluter (right).

D. Predictive Modeling

The target of the models tested was a binary classification for each comedy special (either a higher than average or lower than average IMDb rating). Topic vectors created by LDA modeling, along with both cluster assignments, were added to the feature space for each document. In three rounds of testing, model input features were variable while the target remained the same. The methods used in each round were Random Forrest, Stochastic Gradient Descent, XGBoost, and Sklearn's Voting Classifier ensemble method combining the previously mentioned three models [6][7][8][9]. 15% of the data was held for testing and the remaining used for training the models. Training data in each of the three rounds consisted of: LDA topic vectors only, LDA Cluster and TF-IDF Cluster assignments (split into one-hot representations), and finally a combination of LDA topic vectors with both sets of cluster assignments.

IV. Results and Discussion

Topics derived using the LDA model are given in Table 3 along with the interpreted labels that were chosen to identify them and the mean proportions of each topic across the corpus (Fig 5). Most topics are relatively easy to identify, however, topic 1 was a bit more nebulous. The fact that there is a much higher portion of this topic across the corpus, combined with the non-specific nature of most of the high probability terms, could indicate that this is generally just clean filler material, or perhaps even an artifact of the language. Along these lines of conjecture, topic 2 seems to highlight the speaking mannerisms of a typical English person. Topics 1 and 2 are not so much of an actual topic, still the distinction proved useful for identifying and clustering the documents. Filtering names for those with a higher LDA percentage of the "UK" topic than the "Clean" topic reveals nearly all British, Australian, or South African comedians (Table 4).

TABLE III. Topics Modeled With LDA

Topic Index	Top Terms	Interpreted Topic Name
0	black_people, ni**er, police, jail	Policing and African Americans
1	awesome, restaurant, daughter, store, bathroom, yell	Clean humor
2	mate, lovely, brilliant, mum, bloke	UK
3	boyfriend, pregnant, abortion, husband, wedding, kiss	Relationships
4	cat, panda, frog, fish, sheep, bird	Animals
5	Trump, immigrant, president, racist, vote, government, terrorist	Politics
6	War, planet, religion, language	Big Picture

Another segment that can be discovered by partitioning the data according to the LDA topic vector are African American comedians. By selecting only rows that contain more than 25% of the "Policing and African American" topic, we get a list of almost all African Americans. Those included in the list of names with a greater than 25% portion of the "Big Picture" topic happen to be comedians who like to include a lot of philosophy in their routines and who are thought of as being more cerebral (Table 4).

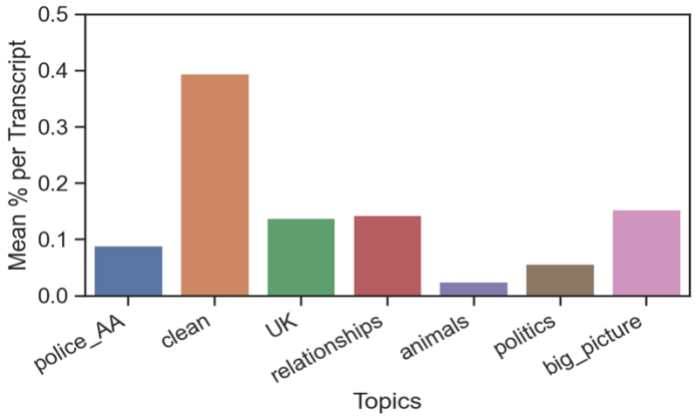

Fig. 5. Mean percentages per topic across the entire dataset.

TABLE IV. OBSERVATIONS REGARDING VARIOUS FILTERS

Filter	Comedians	Observation
Only names where UK percentage higher than Clean	Dylan Moran, Eddie Izzard, Jim Jeffries, Jimmy Carr, Michale McIntyre, Russel Brand, Sara Pascoe	British comics
Only names where Policing topic portion higher than 25%	Arsenio Hall, Cedric the Entertainer, Chris Rock, DL Hughley, Dave Chappelle, DeRay Davis, Kevin Hart	African American comics
Only names where Big Picture topic portion higher than 25%	Lenny Bruce, Bill Hicks, Doug Stanhope, George Carlin, Stewart Lee, Dick Gregory	Philisophical comics

Although this is not the intended application of topic vectors in this case it is interesting to note how they can be used to identify race or personality features relating to each comedian.

Coincidentally, the number of clusters shown to produce the highest silhouette score, when clustering based on LDA topic vectors, was equal to the number of topics chosen for the LDA model (Fig 3). This is confirmed to be optimal by plotting the SSE for each value *k* and using the elbow method to asses where the bend in the line occurs. It may be that this is a result of the LDA model parsing the documents in exactly 7 ways in the first place and that the latent clusters that are desired are actually more forced than intended. This same strong clustering was not achievable with the extremely large sparse data that was fed to the second clustering algorithm (Fig 4). This was partially expected due to the high dimensionality problem discussed in [19].

TABLE V. ACCURACY SCORES FOR EACH MODEL

Model Type	Training input 1: LDA Topic vectors	Training input 2: Cluster assignments	Training input 3: Topic vectors and cluster assignments
Random Forrest	0.71	0.73	0.68
Stochastic Gradient Descent	0.78	0.80	0.85
XGBoost	0.61	0.63	0.66
Ensemble of all three models	0.83	0.71	**0.88**

Most of the mean topic vectors in each set of clusters are quite different. For this reason, it appears that different qualities in the original data are being represented in each set of clusters, therefore, both sets of cluster assignments were included in training for the classification task. Table 5 gives the accuracy scores of each model attempted during testing rounds. The model with the highest accuracy, at 0.88, was found to be an ensemble (Random Forrest, Stochastic Gradient Descent, and XGBoost) soft voting classifier which took in LDA topic vectors and both sets of cluster assignments as feature data.

The average ratings received for members of each cluster in each set of clusters is shown in Table 6. According to the clustering based on topic vectors, comedy specials that focus on big picture ideas are more likely to get higher ratings than those that focus on policing issues (Fig 6). According to clustering based on TF-IDF matrices, specials that focus on politics are more likely to get

higher ratings than those that focus on animals but because of the weak clustering when using TF-IDF as input, it would be assumed that this relationship is not as strong.

TABLE VI. MEAN RATING PER CLUSTER

Cluster	Mean IMDb Rating
LDA cluster 0	7.09
LDA cluster 1	7.63
LDA cluster 2	7.05
LDA cluster 3	7.31
LDA cluster 4	**8.38**
LDA cluster 5	7.76
LDA cluster 6	7.37
TF-IDF cluster 0	7.21
TF-IDF cluster 1	7.28
TF-IDF cluster 2	7.06
TF-IDF cluster 3	7.73
TF-IDF cluster 4	7.20
TF-IDF cluster 5	6.90
TF-IDF cluster 6	7.76

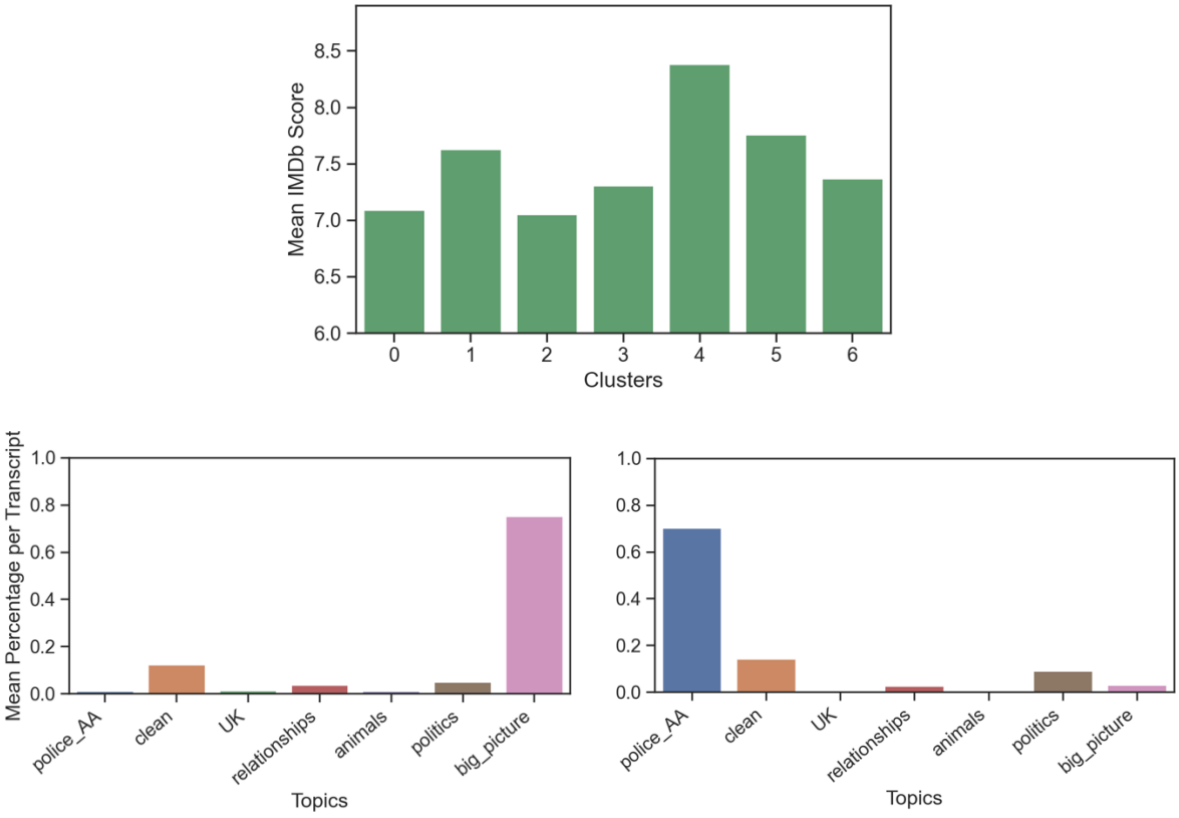

Fig. 6. Mean IMDb rating per cluster for topic-vector-derived clusters (top). Topic distribution for highest rated cluster 4 (bottom left). Topic distribution for lowest rated cluster 2 (bottom right).

V. Conclusions

Clustering has been proven useful in a variety of domains. The normal application is to group data points into clusters in order to discover hidden patterns and natural groupings. The approach used for this work relies on these hidden patterns to achieve an immense reduction in the amount of memory usage needed for training a classifier. The most successful model was ultimately trained on 3 distinct sets of features that were created using unsupervised methods. The representation of each data point is reduced from hundreds of thousands of characters, to a few thousand words, to even fewer important words, and finally to two cluster assignments and a 1x7 topic vector.

The scraping algorithm used in the work was designed to collect as many transcripts as are available and new transcripts are being added frequently. It would be interesting to see how any insights might change over time with the addition of more data. Future research on the subject could include an analysis that tracks changes in ratings of similar material over time. For example, one subject that is more likely to earn a high rating today, may not have been very interesting 10 years ago. Using sentiment or subjectivity analysis as the basis for clustering might be another interesting direction to take. One major weakness of this work is that the dataset is so small. With a few thousand transcripts, more accurate topic modeling and rating predictions could potentially be made.

Finding ways to use less data to form accurate predictions is a requirement many machine learning practitioners face. This becomes an especially hard problem when working with high-dimensional data [20]. The results of this work suggest that clustering, and in particular, stacking cluster assignments in the feature space, may be a potent way to reduce dimensionality and allow for computationally simpler model training.

References

[1] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.

[2] Crain, Steven P., et al. "Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond." *Mining text data*. Springer, Boston, MA, 2012. 129-161.

[3] Steinley, Douglas. "K-means clustering: a half-century synthesis." *British Journal of Mathematical and Statistical Psychology* 59.1 (2006): 1-34.

[4] Singh, Vivek Kumar, Nisha Tiwari, and Shekhar Garg. "Document clustering using k-means, heuristic k-means and fuzzy c-means." *2011 International Conference on Computational Intelligence and Communication Networks*. IEEE, 2011.

[5] Rousseeuw, Peter J. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." *Journal of computational and applied mathematics* 20 (1987): 53-65.

[6] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." *R news* 2.3 (2002): 18-22.

[7] Bottou, Léon. "Stochastic gradient descent tricks." *Neural networks: Tricks of the trade*. Springer, Berlin, Heidelberg, 2012. 421-436.

[8] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

[9] Sagi, Omer, and Lior Rokach. "Ensemble learning: A survey." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4 (2018): e1249.

[10] M. Kelechava, *Using LDA Topic Model as a Classification Model Input*, Medium – Towards Data Science, Apr. 2019. Accessed on: Jun. 6, 2020. [Online]. Available: https://towardsdatascience.com/unsupervised-nlp-topic-models-as-a-supervised-learning-input-cf8ee9e5cf28

[11] *Stand-Up Comedy Directory Page*, Scraps From The Loft, Jun. 2020. Accessed on: Jun. 6, 2020. [Online]. Available: https://scrapsfromtheloft.com/tag/stand-up-transcripts/

[12] *IMDbPY Homepage*, Jun. 2020. Accessed on: Jun. 6, 2020. [Online]. Available: https://imdbpy.github.io/

[13] *Langdetect Homepage*, pypi.org, Jun. 2020. Accessed on: Jun. 6, 2020. [Online]. Available: https://pypi.org/project/langdetect/

[14] *Gensim Homepage*, pypi.org, Jun. 2020. Accessed on: Jun. 6, 2020. [Online]. Available: https://pypi.org/project/gensim/

[15] Johnson, David, Vishv Malhotra, and Peter Vamplew. "More effective web search using bigrams and trigrams." *Webology* 3.4 (2006): 35.

[16] Khudanpur, Sanjeev, and Jun Wu. "Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling." *Computer Speech & Language* 14.4 (2000): 355-372

[17] Syed, Shaheen, and Marco Spruit. "Full-text or abstract? Examining topic coherence scores using latent dirichlet allocation." *2017 IEEE International conference on data science and advanced analytics (DSAA)*. IEEE, 2017.

[18] Ramos, Juan. "Using tf-idf to determine word relevance in document queries." *Proceedings of the first instructional conference on machine learning*. Vol. 242. 2003.

[19] Kleinberg, Jon M. "An impossibility theorem for clustering." *Advances in neural information processing systems*. 2003.

[20] Tadjudin, Saldju, and David Landgrebe. "Classification of high dimensional data with limited training samples." (1998).