Extracting categorical topics and topic likelihood from news articles using topic modelling techniques

Srikanth Reddy Nagidi
Department of Computer Science
Northern Illinois University
Dekalb, IL 60115
z1836478@students.niu.edu

Venkata Surya Vamsi Maddukuri
Department of Computer Science
Northern Illinois University
Dekalb, IL 60115
z1855404@students.niu.edu

Anurag Gudipati
Department of Computer Science
Northern Illinois University
Dekalb, IL 60115
z1862977@students.niu.edu

Santosh Raju Chintala Pallipata
Department of Computer Science
Northern Illinois University
Dekalb, IL 60115
z1862061@students.niu.edu

Manish Raj Goud Dusari
Department of Computer Science
Northern Illinois University
Dekalb, IL 60115
z1863779@students.niu.edu

***Abstract*— Over the past few years, news articles have been exploring vast topics which are happening around the world. They construct data with lot of stop words and noisy words. Topic modelling algorithms for social medias have been studied well but our exploration of topic analysis in the stream of news articles due to the exploitation in the form of noisy words and due to this only reason previous research was limited to analyze just one topic. The main problem for a digital medium to produce articles is nothing but choosing a right topic to hit the major audience. So, we would like to explore the data from the news articles using NLP techniques (e.g. data cleaning, data preprocessing, wiping out noisy words etc.) and topic modelling techniques (e.g. using LDA model) to produce a visualization which represents the major explored topics by different mediums. We use this model which incorporates the data from several news articles and reduce noisy words using traditional semantic analysis and wipe out noisy words and then operate on the data using Latent Dirichlet Allocation algorithm and produce a distinct visualization of the most explored topics around the world.**

***Index Terms*— Natural Language processing , Stemming and lemmatization, stop words, Bigram, Latent Dirichlet Algorithm**

I. INTRODUCTION

In these days of abundant information and abundant information producers, hitting the right audience with right content is most important thing now-a-days. So, we would like to propose a model which analyzes the data present in the news articles and produces the topic likelihood and categorization of topics with the help of visualization techniques. With the information we provide one can easily assess the topic categorization by analyzing graphs and produce the content of that topic particularly. For producing data visualization we used topic modelling methods and NLP techniques for data analysis.It is to be noted that, some publishers have turned to Altmetrics because they appear more rapidly than citations.

We wanted compute the fields which are currently being popular and has a big impact on people all over the world.

II. DATASET

The Altmetrics dataset was chosen as our dataset. Also, it was easier to collect the entire data. As altmetrics focuses on social media platforms that often provide free access to usage data through Web APIs, data collection is less problematic. The dataset was merged from different JSON files and then integrated into a single .csv file. The entire dataset was not taken at once and sampling was performed on the dataset. Sampling helps reduce the time taken to process the data and also gain information based on the subset of the data [14]. Around 100,000 tuples were considered. Out of 100k records in the data set we considered only the records which has news summaries in it. That narrowed down to 25,000 documents. So, basically we are considering 25k research papers for our topic modelling in this paper.

We extracted news summaries and its related scopus subject and written into a csv, which will help us in analyzing the data.

III. DATA PRE-PROCESSING

Nobody wants a raw chunk of unorganized data as it is hard for everyone to analyze. So first task

before topic modelling was to clean the data as finely as possible using several techniques. After converting the unorganized into perfectly organized csv file, it is up to mathematical functions to deliver. So the steps to be taken for data pre-processing are:

A. Removing non-English words

We have used Langdetect module to remove all the non-English words, so that only English news articles only be considered. After filtering we have found that 3000 non-English documents are present and removed all of them.

B. Data cleaning by removing characters and symbols

Although there will be some plenty of characters and symbols which describes more than words but that does not help for us to analyze. So we removed all the characters and symbols which comes of no use using regular expressions. So we are now left with the useful English text. And these texts which are in sentences are converted to a list of words.

C. Removing stop words

In the texts we will have so many stop words like the, a, for etc. They might describe the situation and meaning of the sentence but they are unnecessary while analyzing the data. So, we have removed those 174 stop words by importing a package and then added 17 more stop words to the list as we observed these words are most repetitive and gives no meaning.

D. Lemmatization and Stemming

The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. For example, in English, the verb 'to walk' may appear as 'walk', 'walked', 'walks', 'walking'. The base form, 'walk', that one might look up in a dictionary, is called the lemma for the word. After removing all those words we are left with lemmatization or stemming of the words.

In our paper we have used lemmatization to convert all the words in the list of documents to its base form.

IV. Bi-gram model

Some English words occur together more frequently. For example - Sky High, do or die, best performance, heavy rain etc. So, in a text document we may need to identify such pair of words which will help in sentiment analysis. First, we need to generate such word pairs from the existing sentence maintain their current sequences. Such pairs are called bigrams. A bigram is an n-gram for n=2. The frequency distribution of every bigram in a string is commonly used for simple statistical analysis of text in many applications, including in computational linguistics, cryptography, speech recognition, and so on.Bigrams help provide the conditional probability of a token given the preceding token, when the relation of the conditional probability is applied:

$$P(W_n|W_{n-1}) = \frac{P(W_{n-1},W_n)}{P(W_{n-1})}$$

The probability $P()$ of a token W_n given the preceding token W_{n-1} is equal to the probability of their bigram, or the co-occurrence of the two tokens $P(W_{n-1}, W_n)$, divided by the probability of the preceding token.

Python has a bigram function as part of NLTK (Natural Language Toolkit) library which helps us generate tokens for these pairs. When we are dealing with text classification, sometimes we need to do certain kind of natural language processing and hence sometimes require to form bigrams of words for processing.

V. Document-term matrix

To generate an LDA model, we need to understand how frequently each term occurs within each document. To do that, we need to construct

a document-term matrix. The gensim package we used creates a dictionay, assigning a unique integer id to each token, in doing so collecting word counts and relevent statistics. This dicitionary is converted to bag of words called corpus - a list of vectors equal to number of documents. In each document vetor is a series of tuples (term ID, term frequency).

VI. Topic Modelling techniques

A. Latent Dirichlet Algorithm(LDA) model

In more detail, LDA represents documents as mixtures of topics that spit out words with certain probabilities. It assumes that documents are produced in the following fashion: when writing each document, you on the number of words N the document will have (say, according to a Poisson distribution).

Choose a topic mixture for the document (according to a Dirichlet distribution over a fixed set of K topics). For example, assuming that we have the two food and cute animal topics above, you might choose the document to consist of 1/3 food and 2/3 cute animals.

Generate each word within in the document by: First picking a topic (according to the multinomial distribution that you sampled above; for example, you might pick the food topic with 1/3 probability and the cute animals topic with 2/3 probability).

Using the topic to generate the word itself (according to the topics multinomial distribution). For example, if we selected the food topic, we might generate the word broccoli with 30 percent probability, bananas with 15 percent probability, and so on.

Assuming this generative model for a collection of documents, LDA then tries to backtrack from the documents to find a set of topics that are likely to have generated the collection.

Like the same way we produced the topic analysis model using news article summaries by determining the topics which are being discussed and their coherence values to produce the topic likelihood and mainly for topic categorization.

We suggest to take 14 as the number of topics for topic modelling as we find high coherence value at 14 and 32. But we are dealing with the sample documents and with minimal prototype so we took 14 and built model.

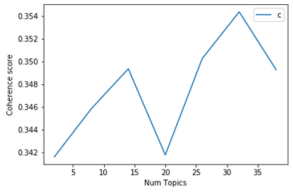

Fig. 1. Taking 14 as the number of topics for topic modelling

VII. Data visualizations on the developed model

A. Words in each topic

The output from the model is a 14 topics each categorized by a series of words. LDA model doesnt give a topic name to those words and it is for us humans to interpret them. Some of the topics and words in them are show in figure 2

Fig. 2. Words in each topic

B. Word counts versus weights of each keyword

It is also important to note the weights of each keyword in the topics and how frequently the words have appeared in the documents. Figure 3

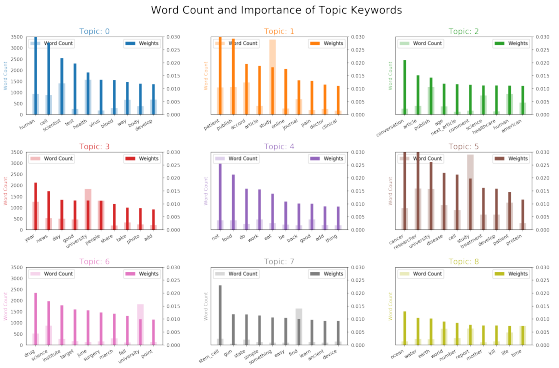

Fig. 3. Word frequency and weights in each topic

C. Number of topics per dominant topic

After building LDA model, this model is applied to each every document and tried to find out to which topic this document belongs. Its not necessary that a document belongs to only one topic. So, the model gives the percentage of the document to a topic. We also calculated which topic is more dominant among the others for a particular document. Example is shown in the Figure 4.

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13	d
0	0	0	0.0803124	0	0	0	**0.435914**	0	**0.101481**	0	0	0	**0.289508**	0.0880821	
1	0	0	**0.478583**	0	**0.129247**	0	0	**0.26022**	0	0	0	0	**0.121009**	0	
2	0	0	**0.540138**	**0.145717**	0	0	0	0	0	0	0	**0.104997**	**0.204182**	0	
3	0	0	0.0891375	0	**0.129225**	0	0	0	0	0.0579656	0	0	0	**0.718447**	
4	**0.468194**	0	0.0582991	0	0	0	**0.177804**	0	0	0	0	0	**0.290479**	0	
5	**0.171622**	0	0	0	0	0	0	0	**0.819246**	0	0	0	0	0	
6	0	**0.530471**	0	0	0	0	0	0.0858736	0	**0.375284**	0	0	0	0	
7	0	0	0	**0.481963**	0	0	**0.374666**	0	0	**0.136105**	0	0	0	0	
8	0	0	0	0.0585025	**0.600905**	0	0	0	0.0745557	**0.116075**	0	0	0	**0.14526**	
9	0	0	**0.223442**	0	0	0	0	0	**0.763429**	0	0	0	0	0	

Fig. 4. Dominant Topic for each Document

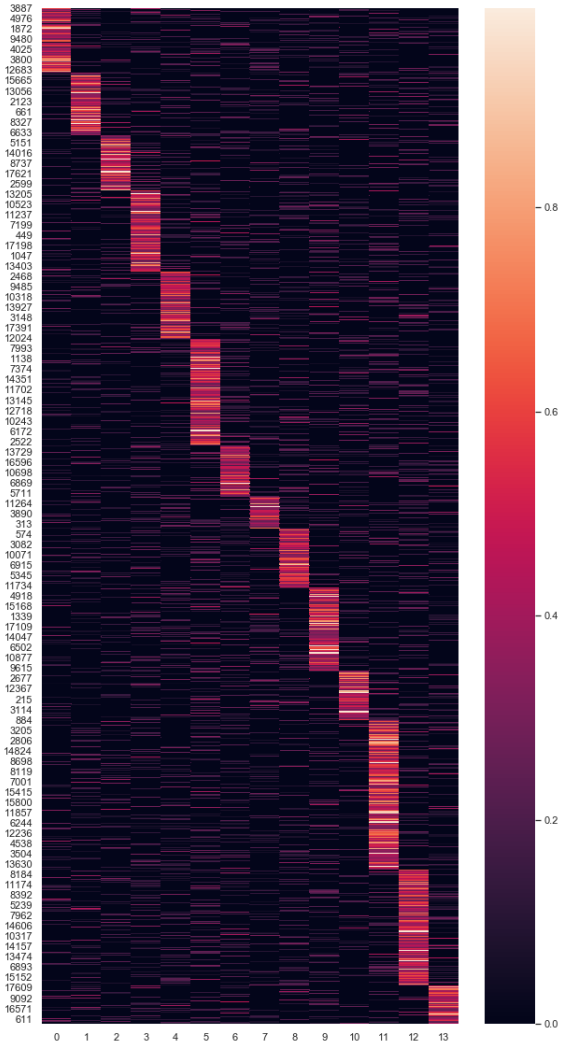

Fig. 5. Topics vs Documents

D. Visualizing documents and topics

Visualizing dominant topics with sorted order of topic vs document numbers generated for LDA.

E. Visualizing words frequency in each topic

The Fig.6 will display the overall term frequency and estimated term frequency within a selected topic of a word. For example it displays the word frequencies in topic 4.

F. Visualizing the clusters of the documents in a 2D space

Lets visualize the clusters of documents in a 2D space using t-SNE (t-distributed stochastic neighbor embedding) algorithm. t-SNE converts the similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional

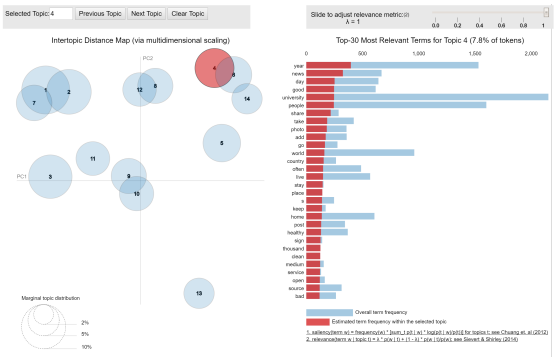

Fig. 6. Frequency of words in a topic

data. t-SNE has a cost function that is not convex, i.e. with different initialization we can get different results. Along with the t-SNE we also used Principal component analysis method to reduce the dimensionality. This will suppress some noise and speed up the computation of pairwise distances between samples. It is shown in Fig.7

VIII. Future work

Here from the data set we have only considered the summary of the news articles for a given paper. If the entire text of the news articles is considered and by extracting full text-features of the text in the news articles a deep analysis can be done. Full text features are like list of contents in the index, author and his H-index, title, meta data features, finding out entity for each sentences.

IX. Conclusion

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

References

[1] Thomas L. Griffiths and Mark Steyvers, Finding scientific topics, 1998.

[2] David Newman and Edwin V. Bonilla and Wray Buntine, Improving Topic Coherence with Regularized Topic Models, CA 2004

[3] Yanggiu Song and Shimei Pan and Shixia Liu and Michelle X. Zhou and Weihong Qein, Topic and keyword re-ranking for LDA-based topic modeling, China.

[4] Adie E, Roe W (2013) Altmetric: enriching scholarly content with article-level discussion and metrics. Learned Publishing 26: : 1117. Available: http://figshare.com/articles/Enrichingscholarlycontentwith article level discussion and metrics/105851. Accessed 2013 February 19.

[5] Japkowicz, N., Stephen, S. (2002). The class imbalance problem: A systematic study. Intelligent data analysis, 6(5), 429-449.

[6] Xin Guo and Yang Xiang and Qian Chen and Zhenhua Huang and Yongtao Hao, LDA-based online topic detection using tensor factorization.

[7] Jey Han Lau and Karl Grieser and David Newman and Timothy Baldwin, Automatic labelling of topic models.

[8] Marshall, M. N. (1996). Sampling for qualitative research. Family practice, 13(6), 522-526

[9] Tengfei Ma, Lexicon extraction from non-parallel data.

[10] Daniel Ramage and Susan Dumais and Dan Liebling, Characterizing Microblogs with Topic Models.

[11] Blei, D., Ng, A., Jordan, M. (2003), Latent Dirichlet Allocation. Journal of Machine Learning Research.

[12] Asuncion, A., Smyth, P., Welling, M. (2008). Asynchronous distributed learning of topic models. NIPS 2008.

[13] D. Blei and J. Lafferty. Topic models. Text Mining: Theory and Applications, 2009

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. The Journal of Machine Learning Research, 3:9931022, 2003.

[15] T. L. Griffiths and M. Steyvers. Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America.

[16] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In WebKDD/SNA-KDD 07: Proceedings of the 9th WebKDD and 1st SNA-KDD Workshop on Web mining and Social Network Analysis.

[17] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text web with hidden topics from large-scale data collections. In WWW 08: Proceedings of the 17th International Conference on World Wide Web.

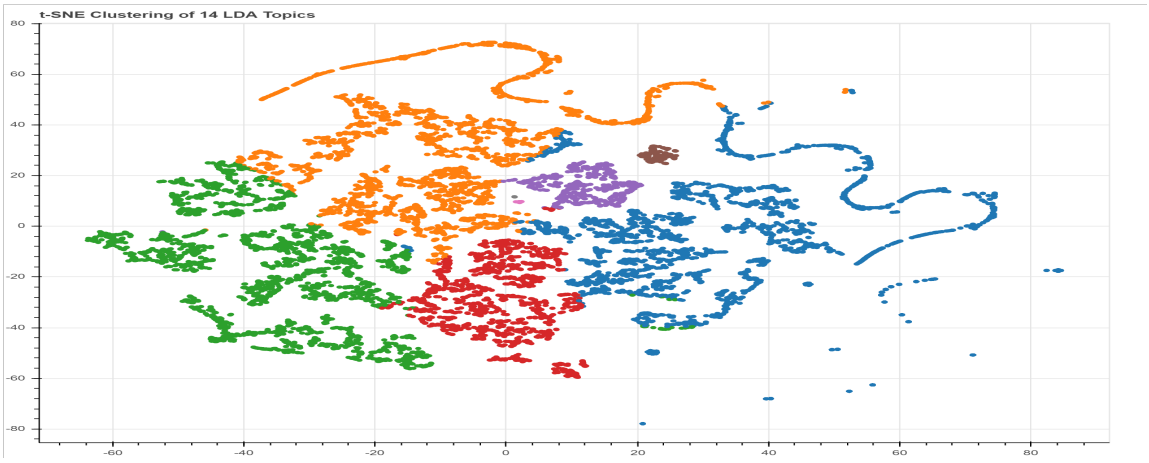

Fig. 7. This is a tiger.