Computer and Information Science; Vol. 11, No. 4; 2018
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Topic Modelling in Bangla Language: An LDA Approach to Optimize Topics and News Classification

Mustakim Al Helal[1] & Malek Mouhoub[1]

[1] Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada

Correspondence: Malek Mouhoub, Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2. E-mail: mouhoubm@uregina.ca

Received: September 20, 2018 Accepted: October 9, 2018 Online Published: October 31, 2018

doi:10.5539/cis.v11n4p77 URL: https://doi.org/10.5539/cis.v11n4p77

Abstract

Topic modeling is a powerful technique for unsupervised analysis of large document collections. Topic models have a wide range of applications including tag recommendation, text categorization, keyword extraction and similarity search in the text mining, information retrieval and statistical language modeling. The research on topic modeling is gaining popularity day by day. There are various efficient topic modeling techniques available for the English language as it is one of the most spoken languages in the whole world but not for the other spoken languages. Bangla being the seventh most spoken native language in the world by population, it needs automation in different aspects. This paper deals with finding the core topics of Bangla news corpus and classifying news with similarity measures. The document models are built using LDA (Latent Dirichlet Allocation) with bigram.

Keywords: topic modeling, classification, natural language processing

1. Introduction

During the last decade, the amount of data generated by people made history. Indeed, roughly 2.5 quintillion bytes of data is produced daily according to the study of DOMO and ninety percent of the data in the world has been created in the last two years alone [1]. As the amount of available data is increasing tremendously, it becomes difficult to access and process the data we are looking for. To retrieve the underneath meaning of the data automated process is required in an efficient way.

Topic modeling is a frequently used text-mining tool for discovering the hidden semantic structures in a text body. It provides a convenient way to analyze big unclassified text. A topic contains a cluster of words that frequently occurs together. The topic modeling approach can connect words with similar meanings and distinguish between uses of words with multiple meanings [2]. Unlike English, very few modeling tools have been developed for other languages. However, the internet content is only 51.2% in English [3] and this percentage might decrease over the coming years. Therefore, it is becoming essential to develop similar tools for other languages.

Bangla has become one of the most popular languages in the world after the announcement to observe February 21st as International Mother Language Day annually by UNESCO on November 17th, 1999 [4]. Over time, there is a good number of Bangla news portals, blogs, eBooks, web pages, search engines, … etc. Although the content is rich enough, the research in Bangla is not frequent due to insufficient datasets, unorganized grammar rules which is the core challenge to work with Bangla. Considering these challenges, we have created our own corpus and proposed the first ever topic modeling tool for Bangla. We are confident that this tool will be very useful to many Bangla speaking users. Several research works have been conducted with LDA for categorizing unclassified texts. The most recent of these works is a generative LDA model designed to categorize texts corpus in English [5]. An empirical result on application of the designed model was presented in text modelling, collaborative filtering and text classification. In this paper, each document consists of a mixture of topics. In this regard, a model is proposed to extract topics from a news corpus in Bangla. In [5], Blei evaluated a topic model with perplexity. Traditionally, perplexity has been used many times as an evaluation process for the extracted topics but it was found that it does not correlate with human annotations at times [6]. In [7], Blei worked with LDA to categorize research papers. This research considered several journals archived by JSTOR which is an organization for indexing journals in different fields. The objective was to find similar articles for a scientist out of millions of journals, conference papers, etc. This is also one kind of categorization of texts using LDA. Finding the underlying meaning of the huge amount of science journals was the main objective of this paper. In [7], the author

also discussed about the effective way on how to approximate the posterior with mean field variation methods. A formal text mining approach was proposed in [8]. Here, the authors worked with Wikipedia and twitter data. Two different perspectives were explored with these two data sets. From the Wikipedia data, a document topic model was achieved aiming to a topic wise document search [8]. On the other hand, with the twitter data a user topic model was explored to identify user¢¢s interest in twitter data. This idea can also be implemented on newspaper data to explore the news trends over a certain time. In this paper, similarity measure for the Wikipedia data was also calculated and demonstrated for different articles against a selected article. Another trend finding work on topic modelling with LDA was done in [9] where the goal is to investigate the research development and current trends from a collection of scholarly articles. A whole picture of LDA over the past 20 years has therefore been illustrated and the paper is more of a survey on LDA applications.

Despite these past efforts, little has been done with LDA in Bangla language. Bangla having a completely different grammatical structure and stemming techniques, it is challenging to identify topics from this language. In [10], a text summarization technique was developed particularly for Bangla language. However, it is a heuristic model and LDA was not used for this text summarization. Two different approaches, namely Abstractive and Extractive, were discussed in the paper. However, the reported research deals with the extractive method only. A set of Bangla text analysis rules were developed based on a heuristic approach [10]. It uses a sentence scoring method to achieve the summarization goal. Although this paper is one of the very few papers that talks about text summarization in Bangla language, the power of topic modelling was not used in any means. Our goal is to apply the LDA and see how it works for news categorization in Bangla.

2. Data Preprocessing

Collecting the data for Bengali language has always been a challenge due to scarcity of resources and unavailability of publicly available corpus. Although various research works have been going on with Bangla, none of those datasets were made available for the public. The dataset that we used is a news corpus. It is collected from one of the most used and popular Bangla newspaper called "*ÅThe Daily Prothom Alo*". We designed a crawler with Python library called Beautiful Soup. The data that we collected has 7134 news with many different categories. The crawler scrapped all the news from January 1, 2018 to March 31st, 2018. The daily Prothom Alo has an online archive that is crawled each daybover the mentioned period of time. The news data are collected in a CSV file which is illustrated in Figure 1.

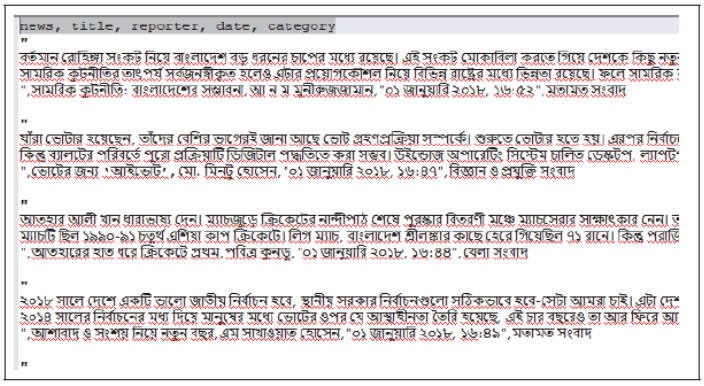

Figure 1. The news corpus: CSV file

Once the dataset is ready then we start the preprocessing which consists of the tasks of Tokenization, removal of stop words and creation of the Bigram. From the news, each word is tokenized and put into an array. Let us consider an example of Bangla word tokenization for a given sentence. Here, each word in the sentence is tokenized and from the implementation point of view these tokenized words are then appended into a Python list for our purpose. Like English, Bengali language has a lot of stop words. These are also the connecting words just like prepositions and conjunctions. However, Bangla being brought up into the NLP world quite recently, there is still no fully established list of stop words. Consequently, we used the stop words list from [11] and enhanced it with our program. In this regard, a list of 438 stop words are used. For the task of topic modelling with Bangla, Bigram creation is an important part. A bigram is a sequence of two adjacent tokens in the corpus occurring frequently [12]. A probability is then calculated for these words to occur one after another. If they have a threshold value then these word pairs are combined together and put into a new token in the dictionary. Basically, bigrams

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

are n-grams where n = 2. A conditional probability (W_n given W_{n-1}) is calculated for bigrams as follows.

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

3. Proposed Model

As we have our news corpus ready t ıes, we formalize the proposed model on how to train the Bangla news corpus ıc modelling. In this Section, we describe the proposed model in a step by step process. Our main goal from this research is to find a way to extract the topics from our news corpus. A methodology is proposed to find out the right topic a news belongs to. This way each news can be classified in its right category. The proposed model is illustrated in Figure 2.

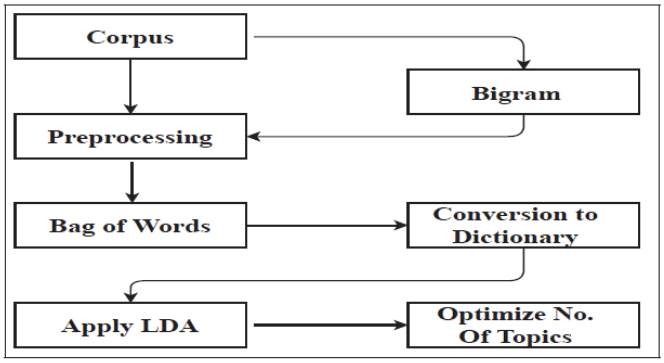

Figure 2. The proposed model

This is the basic structure on how the model works. Once we have the dictionary ready with the preprocessing already been performed on it, we apply the LDA algorithm. We have trained 7134 news. The dictionary is already set up in the preprocessing phase. So this whole dictionary goes into the model and it extracts a number of topics. However, LDA does not know how many topics it has to extract. We propose a coherence [13] based method to understand the optimal number of topics. From that experiment, we feed the right number of topics as a hyperparameter in our training model. One problem with LDA is that it can get over fitted if too many topics are extracted. So, finding the right number of topics is very important.

Before we train our model and run the LDA algorithm, we run a coherence model with roughly 200 topics just to explore the graph. As we know that it is not possible to have 200 topics with about 7134 news, we just set the value to check the gradual coherence movement across different topics and found that it gets to the peak at around topic 47. So, we took that number and fed it into the algorithm. This way, the model does not under fit or neither over fits. Once we get the model trained with our corpus, we evaluate it through experiments. We have performed a cosine similarity check between different news. Some news were similar while some others were different. So, it was expected to have more similarity score between similar news and less similarity between news about different agenda. We achieved those scores for the trained LDA model. However, cosine similarity can also be achieved from Doc2Vec model. That is why we used the Doc2Vec model just to compare the cosine similarity score between LDA and Doc2Vec and gain an insight on how both models work in terms of cosine similarity. A comparative evaluation with other variations of LDA has also been performed and reported in the next Section.

4. Experimentation

The goal of the first experiment that we perform is to understand the number of topics that we need to infer from the trained model. LDA itself cannot understand the optimal number of topics. We performed an experiment to understand the optimal number of topics. This number depends on the dataset and the main research goal. Our purpose is to infer topics from an online newspaper with about 7,134 news instances. When too many topics are inferred from a LDA model, it may get over fitted which is not expected at all. On the other hand, extracting too few topics does not make sense too. A coherence based value is considered for understanding the right number of topics. We have experimented the model with 200 topics along with the aggregated coherence value for these topics as shown in Figure 3.

As can be seen, the model reaches its peak value at 47, that we consider as the optimal number of topics.

4.1 Similarity Measure

Since a trained LDA model already groups topics in terms of their keywords, we undergo through an experiment to explore the cosine similarity measure from our trained LDA model. We feed a couple of document pairs each time and see the cosine similarity value. However, similarity measure can also be performed with a Doc2Vec model. Consequently, we compare the similarity scores from both the LDA and the Doc2Vec. These scores are shown in Table 1.

As we can see, each time a pair of documents are fed into both of these models. For example, doc 1 and doc 2 are two highly related documents. They are both talking about a news on the Myanmar Rohingya issue in Bangladesh. As a human interpreter, someone will judge these two news as a highly related pair. LDA cosine similarity gives this pair a 95.15% similarity which would have been almost close to the human interpreter. On the other hand, Doc2Vec performed poorly and gave only a 68.54% similarity which demonstrates its poor performance. Document 1916 and document 1060 are talking about Technology. In this case, LDA performs better than the Doc2Vec again. Now, let us see how these models work on dissimilar news. It is expected that two dissimilar news are likely to have a low cosine similarity score. In this regard, we performed the similarity check for document 5 and document 9 where one is about sports and the other news is about foreign affairs respectively. LDA gave only a 19.07% match where Doc2Vec reported a score of 50.01%. This is a winning situation for LDA in all respects.

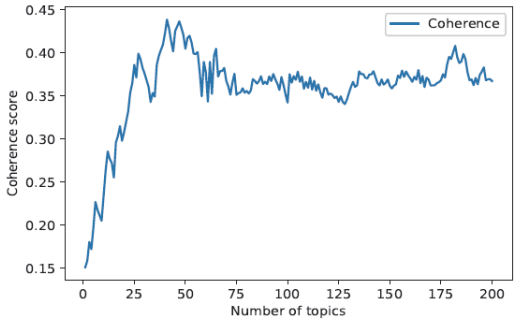

Figure 3. Coherence based on number of topics

Table 1. Showing cosine similarity score between different models

Document Pairs	LDA	Doc2Vec
(doc5, doc9)	19.07%	50.91%
(doc5, doc6)	71.63%	72.55%
(doc271, doc272)	68.68%	60.61%
(doc1, doc2)	97.15%	68.54%
(doc1, doc513)	72.45%	30.31%
(doc 1916, doc 1060	80.99%	37.91%

4.2 Classifying News

We have gone through a document classification experiment according to topics. As we have a trained LDA model and also extracted the topics, we wanted to go further with the first ever news classification in Bangla language with LDA. Therefore, we propose a method for classifying news with our LDA model. At first, we extract a document vs topic matrix. In this matrix, each term is tagged with a probability to belong to a given topic. Let us illustrate this idea with a simple example. Let assume we have a document D = "Dogs and cats are cute" and eventually become $D_{preprocessed}$ = ["*dogs*", "*cats*", "*cute*"]. As a human interpreter, we can easily understand that this is a document with the topic ÅAnimal. We may also consider that we have topics k1 and k2. However, the matrix for document vs term probability distribution will look like the following:

$$\begin{Vmatrix} [(0, p1) & (0, p2)] \\ [(1, p3) & (1, p4)] \\ [(2, p5) & (2, p6)] \end{Vmatrix}$$

where $p_1, p_2, \ldots, p_6$ are all the probabilities. 0, 1 and 2 are the word indexes from our example sentence. For each word, we have:

$$\sum(p_1 + p_2) = \sum(p_3 + p_4) = \sum(p_5 + p_6) = 1 \tag{2}$$

In our proposed method, we extract the mean of words for each topic. To make the example more generic, let us assume we have n terms and m topics. Our matrix will be as follows:

$$\begin{Vmatrix} [(w1, p1) & (w1, p2) & ¢.. & (w1, p3)] \\ [(w2, p4) & (w2, p5) & ¢. & (w2, p6)] \\ \vdots & & & \\ [(w_n, p_{m-2}) & (w_n, p_{m-1}) & ¢. & (w_n, p_m)] \end{Vmatrix}$$

So the mean topic for each each word is calculated with the following equations:

$$mean_1 = \sum(p_1 \ldots p_3)/m \tag{3}$$

$$mean_2 = \sum(p_4 \ldots p_6)/m \tag{4}$$

$$mean_m = \sum(p_{m-2} \ldots p_m)/m \tag{5}$$

Finally, the document belongs to the topic having the largest probability value.

4.3 Topic Extraction

Figures 4 and 6 report some topics extracted from the newspaper in the English translation. As we can see, each topic consists of 10 words related to the topic. Each word has a corresponding probability and for each topic, all the probabilities of the words will sum up to one. Also, for each topic we are showing 10 highest probable words. However, some words might have no sense to that topic but most of them relevant. Some topics are a bit mixed and the meaning can be different but most the other topics make sense and can be classified as a category from the newspaper. However, the topics are not tagged automatically. By seeing the words grouping for each topic, the tags are made manually since LDA will only provide with the group of words/terms known as topics. With each word, we got a probability in a descending order. These probabilities sum to one.

4.4 Document Classification

In this section, we will visualize how the proposed model works for the document classification tasks. We will feed the model some random news and graphically explore how well the model can predict its topic. This is basically a topic vs document distribution. In Figure 5, we have fed a news about a movie.

Topic 36 is the most relevant topic for this news outperforming the other topics in terms of their probability score. Since the news is about a movie, we get a relevant answer from our model. It gives us a topic consisting of words related to movies. The first word is Cinema with the highest probability. This topic is illustrated in Figure 4.

(36, '0.026*(Movie) + 0.020*(Send) + 0.020*(Director) + 0.019*(statement) + 0.017*(Finishing) + 0.016*"(Actress) + 0.015*"(Next_movie) + 0.015*(Release) + 0.014*"(The movie) + 0.012*(Control))

Figure 4. Word clusters for topic: Movie

In the second experiment, we feed a news about Donald Trump talking about the USA and immigrants facts. As reported in Fig. 7, it is more likely to belong to the topic consisting of the word "Trump" itself with the highest probability and leading to all the other immigration and the USA related terms. This experiment reflects the fact that we can classify Bangla news successfully with this model. The topic for Trump news is illustrated in Figure 6.

5. Conclusion and Future Work

We have demonstrated how topic modelling can be extended with the Bangla language in a large scale. Bangla having a strong online presence over the past few years, there is a lot more to do with it regarding topic modelling and other NLP tasks. Online Bengali libraries can use this tool as a recommender system. This work can also be extended with a trending topic scopes which is yet to be explored for the Bangla news and media text data. Trending topics can play a vital role in predicting the corruption rate over different districts in Bangladesh. Moreover, it can be stretched to use in public sentiment analysis for prediction over diverse aspects through print and news media. In this research, we did not use LSI (a variation of LDA) which can be considered in the future work. Different similarity measures can also be explored for document classification.

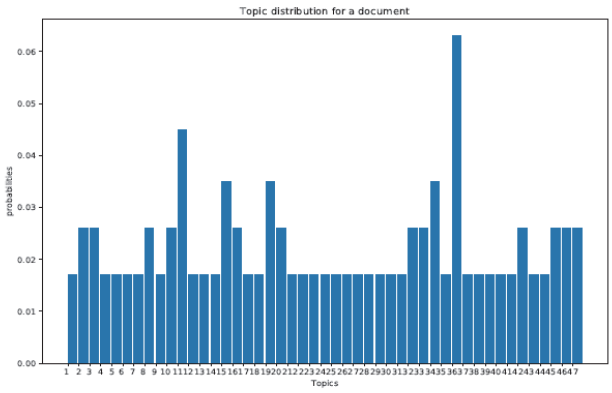

Figure 5. Document topic distribution for movie news

(9, '0.052*(Trump) + 0.044*(President) + 0.025*USA + 0.023*(Against) + 0.020*(Admin) + 0.019*(January) + 0.019*(complaints) + 0.017*(Statement) + 0.016*(Order) + 0.016*(Directly))

Figure 6. Word clusters for topic: Trump

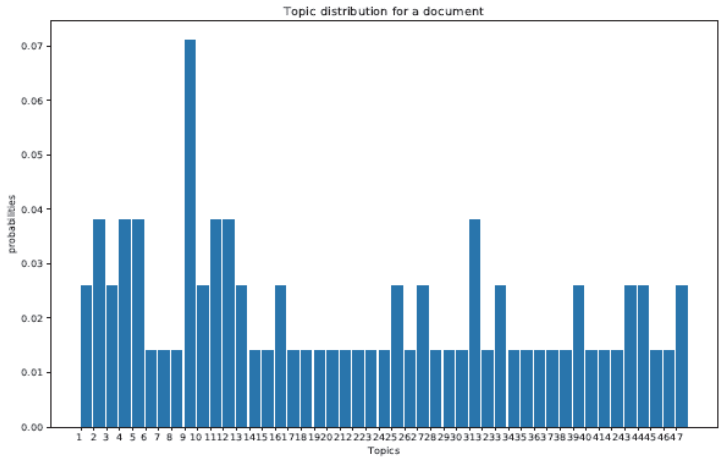

Figure 7. Document Topic distribution for Trump news

References

Abujar, Sheikh, et al. (2017). A Heuristic Approach of Text Summarization for Bengali Documentation." 8th International Conference on Computing, Communication and Networking (8th ICCCNT), 2017 8th International Conference on. IEEE. 2017.

Alghamdi, R., & Khalid, A. (2015). A survey of topic modeling in text mining. Int. J. Adv. Comput. Sci.Appl.(IJACSA), 6(1).

Blei, David M. (2012). Probabilistic topic models. *Communications of the ACM, 55*(4), 77-84.

Blei, David M., Andrew Y. Ng, & Michael, I. J. (2003). Latent dirichlet allocation. *Journal of machine learning research,* 3, 993-1022

Domo.com. (2018). Press Release - How Much Data Does The World Generate Every Minute? Retrieved June 23, 2018, from https://www.domo.com/news/press/how-much-data-does-the-world-generate-every-minute

GitHub. (2018). stopwords-iso/stopwords-bn. Retrieved June 23, 2018, from https://github.com/stopwords-iso/stopwords-bn

Mahmood, A. (2018). Literature Survey on Topic Modeling. Retrieved June 23, 2018, from https://www.eecis.udel.edu/ vijay/fall13/snlp/lit-survey/TopicModeling-ASM.pdf

Markroxor.github.io. (2018). *gensim news classification.* Retrieved June 23, 2018, from https://markroxor.github.io/gensim/static/notebooks/gensim news classification.html

Newman, David, et al. (2010). Automatic evaluation of topic coherence. Human Language Technologies: The 2010 An- nual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics.

Tong, Z., & Haiyi, Z. (2106). A text mining research based on LDA topic modelling. Jodrey School of Computer Science, Acadia University, Wolfville, NS, Canada, 10, 201-210.

Wallach, Hanna M. (2006). Topic modeling: beyond bag-of-words. Proceedings of the 23rd international conference on Machine learning. ACM.

Wikipedia contributors. (2108). International Mother Language Day." Wikipedia, the Free Encyclopedia. Wikipedia, the Free Encyclopedia, 1 Apr. 2018. Web. 13 Apr. 2018.

Wikipedia contributors. (2108). Languages used on the Internet." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 13 Apr. 2018. Web. 13 Apr. 2018.

Copyrights