Topic Modelling Using LDA and LDAMallet

Natural Language Processing

Aishwarya Kanakarajan
Freie Universität Berlin

Abstract

The aim of this project is to build an unsupervised machine learning model that can identify the summary or the title/topic of any content which is usually a large unstructured text. This will help in data labeling or information retrieving techniques. Applications of topic modelling includes the recommender systems, finding articles under the same topic etc. The topic modelling techniques used in this project are Latent Dirichlet Allocation (LDA) and LDAMallet. The performances of both algorithms on a particular dataset are be done. LDA is a generative probabilistic model that extracts the topics of the dataset under the pre-defined number of clusters. LDAMallet Model is the sampling-based implementation of Latent Dirichlet Allocation. The dataset identified for this project is “Cranfield Collection”. It is a very widely used dataset for information retrieval experiments. It contains 1400 abstracts of academic papers developed at the College of Aeronautics at Cranfield University and is publicly available for usage. Text pre-processing has been done and that involves tokenization, stop words removal and lemmatization. The python packages such as genism and pyLDAvis have been used. Model training has been done by tuning the hyperparameter. The evaluation metrics used in the model are perplexity and coherence.

Dataset and Preprocessing

The Cranfield Collection is a corpus usually used for information retrieval experiments. It contains 1400 abstracts of aerodynamics journal articles from the collection of academic papers of the college. The corpus is analyzed by a simple statistical method. There are 1400 abstracts of aerodynamic data in total. As a necessary preprocessing step, the unwanted columns and rows, the punctuations, the stop words and other signs that don't have any semantical significance have been removed. Further steps include the lemmatization and the tokenization of the cleaned data. The dictionary and the word corpus to be fed to the model have been created using the genism module.

Hyperparameter Tuning

Hyperparamter: number of topics along with number of passes through the passage.

Firstly, the number of topics has been set to 10 as fixed value and iteratively the values for the number of passes through the passage has been changed from 5 to 20 in steps of 5. It is observed that the results of the model with the number of passes set to 15 performed better relatively. This conclusion is derived from the coherence value.

Next, the number of topics has been increased from 10 to 20 in steps of 2 and it is observed that the model with the number of topics 12 performed well relatively.

Evaluation Metrics

Perplexity and Topic Coherence are used to evaluate the model performance

• Perplexity: shows how well the model can predict a sample; This is measured as the normalized log-likelihood of the held-out test set. The lower, the better.

• Coherence: Coherence looks at the most-frequently occurring words in each of the generated topics, rates the semantic similarity between them and finds the mean coherence score across all the topics in the model. Based on human-interpretability; the higher the value, the better is the performance of the model.

Overall, the coherence of the model with n=12 has better value w.r.t LDA(BoW) model.

The TF-IDF model has two values equivalent (n=12 and n=15) and the LDAMallet model has coherence values equally likely between the models with n=8 and n=12.

Models

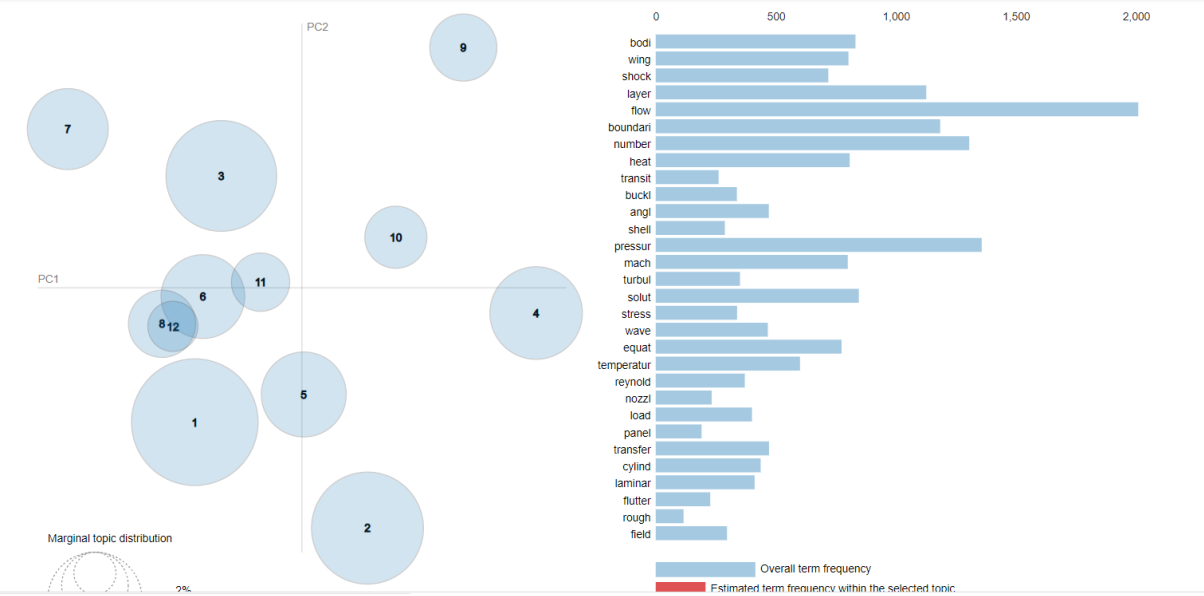
LDA with Bag of Words (BoW):

LDA is used to classify text to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions. Each document is modeled as a multinomial distribution of topics and each topic is modeled as a multinomial distribution of words. Every chunk of text that is fed into the model contains words that are related to each other. LDA works with the assumption that the documents are produced from a mixture of topics and those topics then generate words based on their probability distribution.

The tokenized and lemmatized text is converted to a bag of words — which is the dictionary where the key is the word and value is the number of times that word occurs in the entire corpus. The bag of words is created for every document and the sample result is as below

```
Word 14 ("effect") appears 1 time.
Word 26 ("investig") appears 1 time.
Word 35 ("ratio") appears 1 time.
Word 47 ("support") appears 2 time.
Word 48 ("theoret") appears 1 time.
Word 67 ("flat") appears 1 time.
Word 84 ("plate") appears 1 time.
Word 92 ("shear") appears 3 time.
Word 100 ("treat") appears 1 time.
Word 119 ("uniform") appears 1 time.
Word 184 ("number") appears 1 time.
Word 194 ("size") appears 1 time.
```

The model is built using varying number of topics and the optimal of the experimented values is 12. The visualization of the same is as below



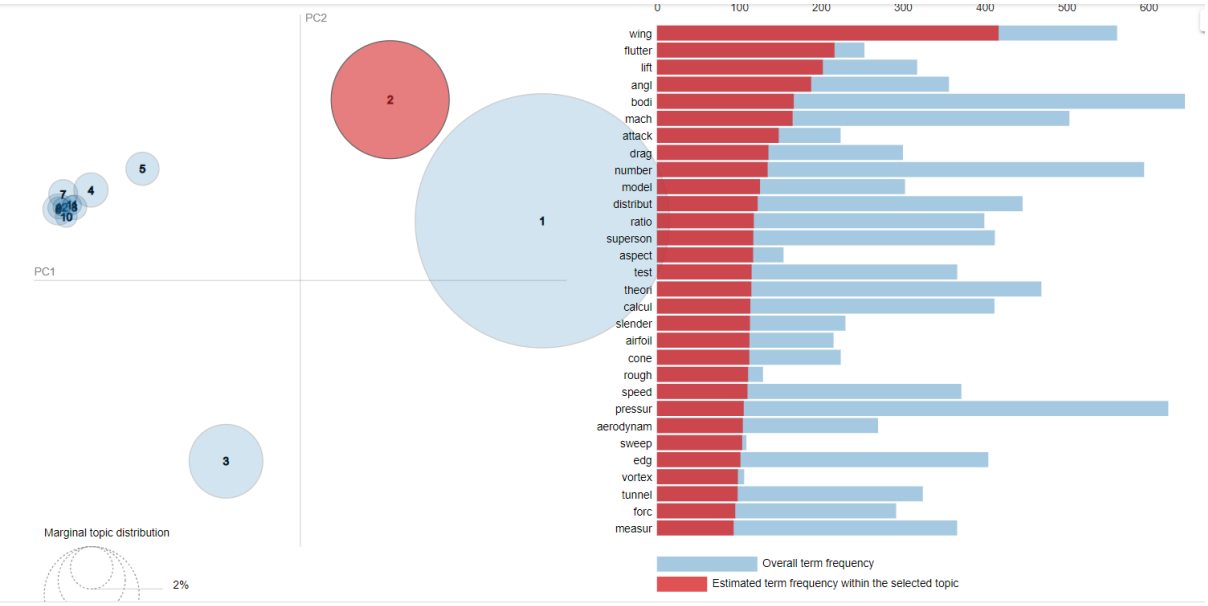
The pyLDAvis library module is used to visualize the performance of the model. From the above map, it is evident that the Topics represented as a bubble is proportional to its prevalence of the corpus. The bigger the bubble the better is the performance. Similar topics tend to overlap each other. The frequency of the words in the topic can be observed by selecting the corresponding bubble. Thus, among all the visualization maps, the map indicating the model with n=12 has larger bubbles that are not overlapping and spread apart indicating well defined topic identification.

LDA with TF-IDF:

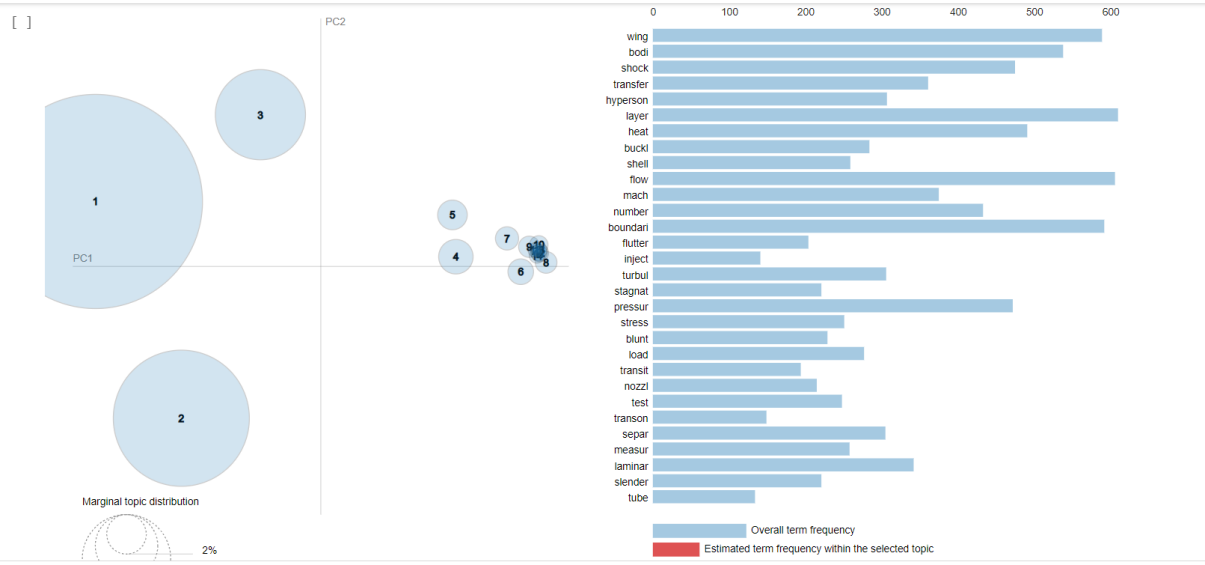
TF-IDF is Term Frequency-Inverse document frequency. For each word in the query, the occurrences in a single document and the occurrence document wide are calculated. These results are multiplied and that represents the overall relevance of a word throughout the documents. With the bag of words created in the previous section of the project, the tf-idf model is extended using the genism library from the created corpus.

The hyperparameter has been set variedly in the range from 10 – 16 and it is observed that the model with number of topics 12 and 14 both perform nearly equally but the visualization differs in such a way that the model with n=12 has bigger bubble and spread apart bubble indicating the recognition of the well-defined topics in comparison to the smaller bubbles in model with n=14. To check the precision of the model performance, the numbers have been set to 13 and 15 as well. It is observed that the model with n=15 also performed well. Both the models (with n=12 and n=15) have been tested on the unseen data. With the results the model with n=12 seemed to be better that the later.

N=12

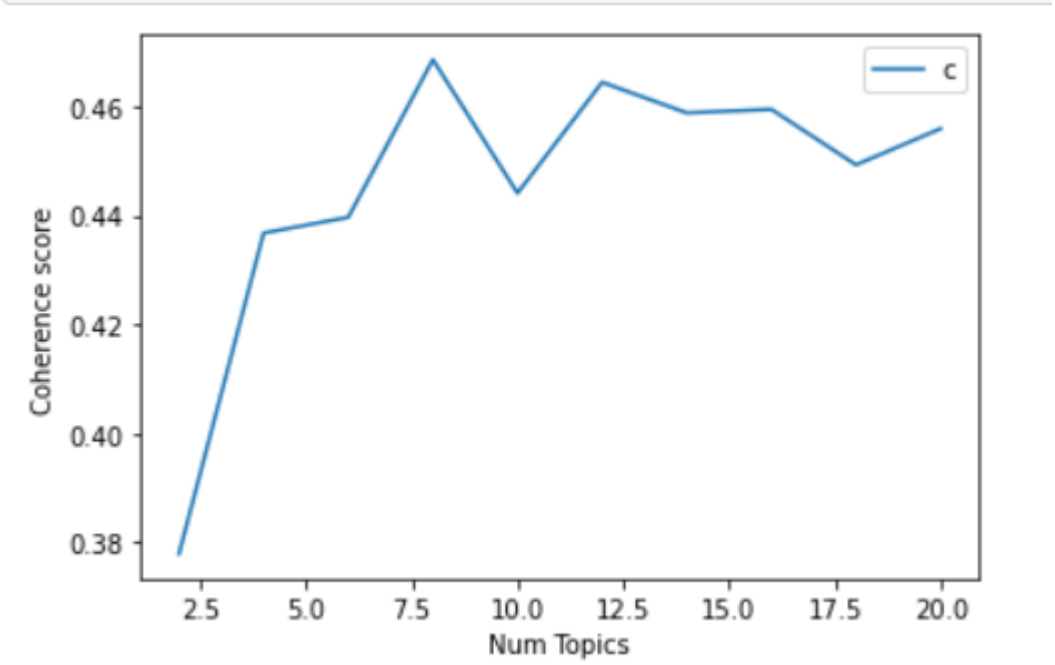


N=15



LDAMallet:

To implement the Mallet model, it is important to download the mallet file from http://mallet.cs.umass.edu/dist/mallet-2.0.8.zip, uncompressed and the path should be declared before building the model. It is also necessary to set the Java environment for the Mallet model to run on the google Collaboratory. With the model readily available using the genism module, to find the optimal number of topics, the number of topics have been charted as below against its coherence value,



It is observed that the model with n=8 and n=12 both performed well. However, with previous results in hand, the model with n=12 has been tested for its performance on the unseen data and it performed well compared to the other two versions.

Results and Conclusion

Based on the coherence values and the observed results on the unseen data, the models performed well when the number of topics is 12. Among the three models, LDAMallet performed better followed by the LDA model with tf-idf vectorizer and thus, making the bag-of-words model the least performing of all the three models.