Text Classification using Topic Modelling

Narasimman Sairam
New York University
ns3184@nyu.edu

Niharika Kunaparaju
New York University
nk1851@nyu.edu

Kavitha Vishwanathan
New York University
kv668@nyu.edu

I. ABSTRACT -

In our project, we decided to experiment with text classification which is a widely used machine learning technique in many applications like email spam detection, automated categorization of documents etc. We aim to perform unsupervised part-of-speech (POS) tagging and build a model using Latent Dirichlet Allocation(LDA). We want to utilize the respective word probabilities generated using LDA in building the feature set for text classification and compare the results to find the accuracy of our model. We decided to use the Reuters dataset to train and test our model for our experimentations.

Keywords – POS tagging, Topic Modelling, LDA, Text Classification, Reuters Dataset, Naïve Bayes Classifier, NLTK Package

II. INTRODUCTION

Modern Information Technologies and Web-based services are faced with the problem of selecting, filtering and managing growing amounts of textual information to which access is usually critical. Information Retrieval (IR) is seen as a suitable methodology for automated management of information/knowledge as it includes several techniques that support an accurate retrieval of information and the consequent user satisfaction. Among the others, the classification of electronic documents in general categories (e.g., Sport, Politic, Religion,.) is an interesting mean to improve the performances of IR systems: (a) users can more easily browse the set of documents of their own interests and (b) sophisticated IR models can take advantages of the categorized data. As an example, the authoring of the textual documents is carried out using the document contents.

A preliminary classification step provides an indication of the main areas of interest. Text classification is, thus, playing a major role in retrieval/filtering and also in the development of user-driven on-line services. The purpose of this work is to try and identify a method for document classification that has good performance in terms of classification accuracy that is acceptable in practical applications. We aim to achieve this using text classification by topic modelling.

III. RELATED WORK

In preparation for this project we researched on a few papers which provided us with an insight on how to go about solving the problem of text classification and use topic modelling top achieve this. The paper [1] Improving text categorization using topic model, prompted us in researching more about LDA and using it instead of simple bag-of-words. LDA approach helps in clustering the words into a set of topics and the words assigned to the same topic are semantically related.

[2]TagLDA, gave us an insight into how POS tagging could be used in conjunction with LDA modelling to give us more accurate results. POS tagging identifies the order of words and filter out all the words that are not nouns. Figure 1 is the snippet of the pseudo-code for removing all the words other than nouns, stop list words and making all the words to lowercase

```
for each document as document {
    POS tag document
    split tagged novel into 1000 word chunks
    for each chunk as chunk {
        remove non-nouns from chunk
        lowercase everything
        remove stop list words from chunk
    }
}
run LDA over chunks
analyze data
```

Figure 1: Pseudocode

IV. DATASET

Currently the most widely used test collection for text categorization research is Reuters-21578. This dataset contains 21,578 documents collected from Reuters newswire articles which are assigned to 135 categories. These documents are classified across 135 categories. The ModAptè split subdivides the data set into a training and a test set of 9,603 and 3,299 documents, respectively. Once discarded all categories with no document in the test set, the

remaining classification scheme is made of 90 categories (R90) and the remaining training set consists of 9,598 documents. Of the 90 categories of R90, we consider the standard subset consisting of the 10 most frequent (R10).

We have used the R10 dataset to train our model which are the 10 most frequent categories by Debole 3 and this subset contains about 75% of the documents. The R10 categories and number of documents in each category for training and test respectively are listed below.

class	# train	# test		class	# train	# test
earn	2877	1087		trade	369	119
acquisitions	1650	179		interest	347	131
money-fx	538	179		ship	197	89
grain	433	149		wheat	212	71
crude	389	189		corn	182	56

Figure 2: R10 dataset

V. Design

Figure 3 below illustrates the framework of feature representations for learning the classification models.

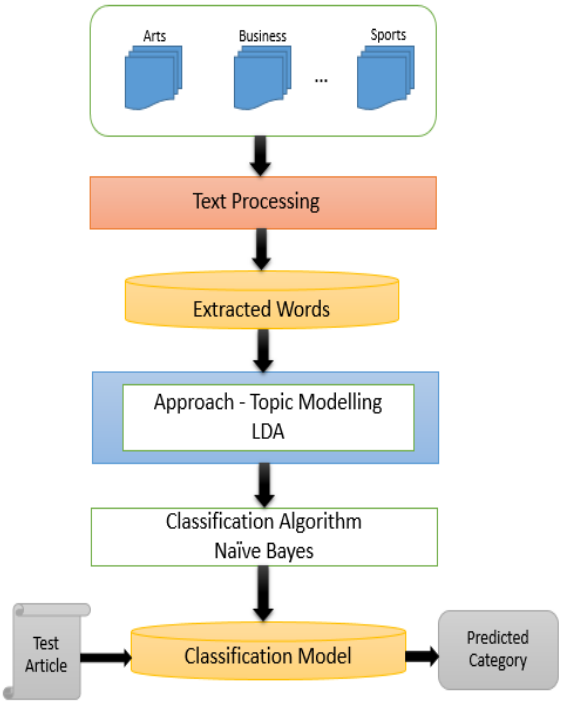

Figure 3: The Proposed Framework

The process of text processing is applied to extract terms. The set of terms is originated by applying the topic model based on the LDA algorithm. In LDA, we derive the proportions that each word constitutes in given topics. The process of checking topic assignment is repeated for each word in every document, cycling through the entire collection of documents multiple times. This iterative updating is the key feature of LDA that generates a final solution with coherent topics. The result is the topic probability distribution for each article. We apply the Naïve Bayes to learn the classification model. This model is used to estimate the performance of category prediction.

VI. Implementation

We experimented with the Reuters dataset and following are the series of steps in our pipeline.

1. R10 Dataset:

There are 5845 documents in the R10 set. Each line represents a single document of the corpus. First word in the line is category that the document belongs to and rest is the content of the document.

2. POS tagging:

We have tokenized R10 documents to get bag-of-words. We then used JET to get part-of-speech tags for each specific word present in the bag-of words.

3. Data Filtering:

Given a sentence, only words with POS tags as NN and NNS describe the topic that the document belongs to. Most of the other tags become irrelevant to the topic. Hence we wrote a script that runs through all the words and discards all the words with tags other than NN and NNS. Out of around 1 lakh tokens, only 13000 unique tokens were remaining that belonged to NN and NNS tag category.

4. Data Pre-processing:

Many measures were adapted to pre-process the data. Below is the list of some important ones:

- Elimination of the common stop words like the, a, it etc.
- Removal of high frequency words that overpower the corpus. Words that appear in more than 60% of the documents are not being considered.
- Removal of infrequent words which were in 5 or less documents.
- Conversion of all words to lowercase.
- Lemmatization to reduce derivationally related words to a common base form.

Natural Language Tool Kit (NLTK) contains a library of packages that can be utilized to pre-process data. NLTK dictionary package maps between ID and token. Further, corpus package takes bag-of-words as an input and gives out a corpus in a form that would be used for LDA modelling.

Figure 4: Data processing

5. Running LDA:

LDA takes 3 input parameters for generating a model. These parameter values are being given based on running the LDA multiple times to find the best possible fit.

a. Topics – Experimented by giving from 6 to 12 and we decided to use 10 topics.
b. Chunk size – A chunk should be such that there is a good mixture of topics in it. We have chosen to go with a chunk size of 2000.
c. Iterations – We started with 5 iterations and found that considerably better results were being generated when we use 20 iterations.

Results that were extracted from the model to be used further in the study include

a. Term to topic probability distribution
   To be used as an input for the classifier later in the analysis.
b. Top 10 words in each topic
   Used to generate word clouds for different set of topics to get a better picture of the categorization.

6. Running the classifier

Once we get the probabilities of the words being present in a particular topic category, we build a feature vector that is given as an input to the classifier. This feature vector contains a matrix of the words and topic probabilities generated by LDA taking into account all the words from the training corpus.

richardson	expands	onomichi	whitney	Topic
0.0000043	0.0000124	0.0000045	0.0000065	earn
0.0000570	0.0000023	0.0000019	0.0000056	acq
0.0000043	0.0000124	0.0000045	0.0000065	earn
0.0000043	0.0000124	0.0000045	0.0000065	earn
0.0000043	0.0000124	0.0000045	0.0000065	earn

Figure 5: Feature vector

In figure 5, each row is representing the word/topic probabilities of words in a single document and the last column represents the topic category the document falls in.

VII. Observations

1. LDA Probability Distribution

We have experimented using different set of input values while finding the probability distribution using LDA. Below is the perplexity plotted over 10 iterations of the entire dataset performed by the LDA.

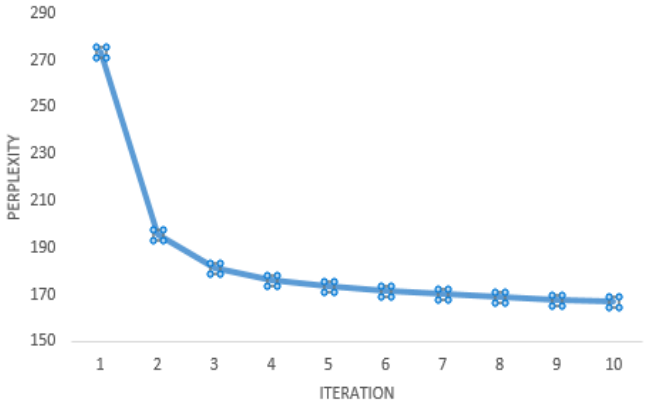

Figure 6: Perplexity vs iteration

We generated word clouds for different topic categories which provide good visual representation. Each word cloud gives the top 10 words ranked by their probabilities.

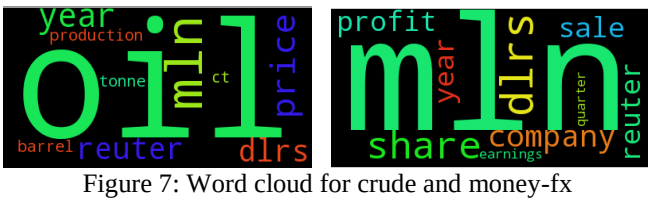

Figure 7: Word cloud for crude and money-fx

One of the ways for finding how LDA performs is by inferring from the topics that the model outputs. After

performing several iterations, we can say that the final model gave an accuravy of 70% approximately.

2. Classifiers

We needed a multi-level classifier since we have more than 2 topic categories. Some of the multi-level classifiers include Support Vector Machine, Naïve Bayes, KNN, Logistic Regression. Among these we chose Naïve Bayes as it works well for sparse matrices.
We ran the Naïve Bayes classifier on the feature vector that was derived from LDA output. In order to compare our model's performance, we ran Naïve Bayes using Bag-of-words model. Figure 8 and 9 represent the performance of our model using precision, F-1 measure and accuracy for Naïve Bayes using Bag-of-words and LDA respectively.

F1 Accuracy = 0.9458
Accuracy = 0.9461
Classification Report:

	precision	recall	f1-score	support
acq	0.95	0.95	0.95	696
crude	0.91	0.97	0.94	121
earn	0.97	0.97	0.97	1083
grain	1.00	0.90	0.95	10
interest	0.92	0.75	0.83	81
money-fx	0.84	0.87	0.85	87
ship	0.92	0.64	0.75	36
trade	0.76	0.97	0.85	75
avg / total	0.95	0.95	0.95	2189

Figure 8: Performance metrics for Naïve Bayes using Bag-of-words

F1 Accuracy = 0.7340
Accuracy = 0.8127
Classification Report:

	precision	recall	f1-score	support
acq	1.00	1.00	1.00	696
crude	0.00	0.00	0.00	121
earn	0.73	1.00	0.84	1083
grain	0.00	0.00	0.00	10
interest	0.00	0.00	0.00	81
money-fx	0.00	0.00	0.00	87
ship	0.00	0.00	0.00	36
trade	0.00	0.00	0.00	75
avg / total	0.68	0.81	0.73	2189

Figure 9: Performance metrics for Naïve Bayes using LDA

VIII. Conclusion

In this project, we learnt how unsupervised topic modelling can be used to improve the performance of a classifier. Text Classification is one of the most challenging in Natural Language Processing domain and this project gave us good exposure on different techniques prevailing in the real world scenario. We do consider this project to be going in the right direction. There are few avenues where we can fine tune to improve the results.

IX. Future Work

Although we have used LDA and got considerably accurate probabilities distribution, we plan do more extensive research on ways to improve the probability distribution of topic terms. We also would like to fine tune the parameters of all the classification algorithms used, to find the most appropriate set of parameters that give us highest accuracy given the model.

X. References

[1] Improving test categorization by using a topic model by Wongkot Sriurai

[2] TagLDA: Bringing document structure knowledge into topic models by Xiaojin Zhu, David Blei, John Lafferty

[3] Unsupervised Part-of-Speech Tagging in Noisy and Esoteric Domains with a Syntactic-Semantic Bayesian HMM by William M. Darling, Michael J. Paul, Fei Song

[4] Reuters – 21578 dataset published by David D Lewis

[5] Reading Tea Leaves: How Humans Interpret Topic Model by Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, David M. Blei

[6] Introduction to Information Retrieval – NLP Stanford website.

[7] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[8] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3:993–1022, 2003.

[9] https://github.com/amueller/word_cloud by Arturo Mueller

[10] Scikit-learn: Gensim