# INFORMATION RETRIEVAL - CS F469

## News Feature Extraction, Classification and Summarization using NLP, TF-IDF, SVM and Naïve Bayes


BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

November, 2017


| | |
|---|---|
| AMRITANSHU JAIN | 2015ABPS0831P |
| PARTHO SARTHI | 2015A7PS0088P |
| VARUN AGARWAL | 2015A7PS0052P |
| DEVANSH GHATAK | 2015A7PS0034P |

## 1. Problem Statement

We aim to achieve classification of news as well as summarising the articles in order to save precious time of readers as well as solve the problem of displaying news articles on increasingly popular mobile devices.

We have used SVM and Naïve Bayes as the classifiers of the processed news articles.

## 2. Background of the problem

With the advent of the World Wide Web, sources of information have increased drastically. News providers have risen in number and have a platform to reach millions of people which was not possible before. This has ultimately led to increase in news which has not been classified and hence either doesn't reach its targeted audience or reaches the wrong audience, which is beneficial to neither.

This study aims to classify news into various groups so that users can identify the most popular news group in the desired country at any given time. Based on Term Frequency-Inverse Document Frequency (TF-IDF) and Support Vector Machine (SVM), a news classification method was proposed. The proposed approach is comprised of three different steps: 1) text pre-processing, 2) feature extraction based on TF-IDF, and 3) classification based on SVM and Naïve Bayes. The proposed approach was evaluated using BBC datasets. The problem comes under the domains of text mining for summarization of the articles and classification, evaluation for categorisation of news.

A major problem faced by us during the work was many of the news article's content had a problem of encoding which hindered us to tokenize and pre-process the data. We also obtained an unexpected result in one of the cases while classifying. We generally observed the trend that title of the news article played a very important role to classify the article in the respected class. But when we summarized the data by excluding the title of the article we obtained a very good result based on evaluation metrics we used.

## 3. Related Work: Literature

Relevant research papers:

M. I. Rana, S. Khalid, and M. U. Akbar, "News classification based on their headlines: A review," in IEEE 17th International Multi-Topic Conference (INMIC), 2014, pp. 211-216.

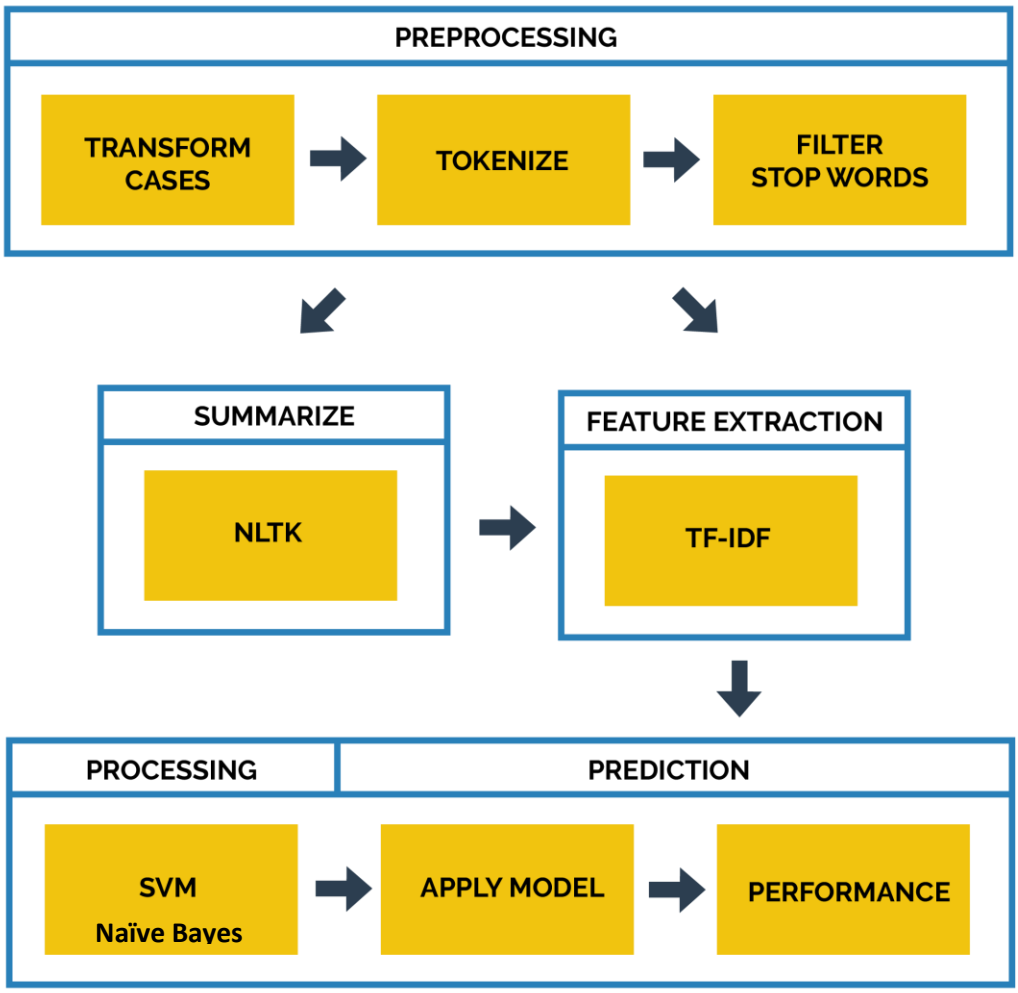
M. W. Pope, "Automatic classification of online news headlines," University of North Carolina at Chapel Hill, November 2007.

A. A. Hakim, A. Erwin, K. Eng, M. Galinium, and W. Muliady, "Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TFIDF) approach," in 6th International Conference on Information Technology and Electrical Engineering (ICITEE), 2014, pp. 1-4.

We tried to summarize and process the news articles in various ways to obtain some interesting results in contrast with the above research papers. We tried to reduce the cost of processing the data while training by summarizing and only keeping the highest weighed sentences. Also generally most of the sentiment analysis tools have a certain word limit of articles to be processed, classifying the data after summarization pretty much solves this issue.

**4. System Description:**

Block diagram of the system and detailed description of each block/module, techniques, functions and GUI design (with minimal focus)



Obtaining the raw data of news articles of over 2400 different articles, we made a dataframe out of the csv file. We then cleaned the strings of the raw content of the articles to handle cases such as upper and lower case characters and more. We then performed tokenization on every article's content and removed stop words from the list of tokens obtained. For every article this process was repeated and the list of relevant words was obtained after pre-processing to make a single document. We tried to experiment with the data here and hence first the raw data wad used to extract the features using the most reliable technique in text mining i.e TF-IDF technique where we weighed the word of content of each article based on the term frequency and the inverse document frequency. We also took the raw data to summarize it first using context analysis (intent parsing) by NLTK to obtain the most important sentences. This data again went through the same process of feature extraction as mentioned above. We then made a vector/word embedding using the features extracted above to represent a single document in the form of a vector. After having n (total number of valid documents) vectors we classified those using Support Vector Machine and Naïve Bayes classifiers into five different classes and evaluated the results based on the metrics of f1_score, precision and recall.

**5. Evaluation Strategy:**

We used various strategies to evaluate the results – f1_score, recall and precision. Interestingly these evaluation metrics returned better results on the raw data as compared with the summarized data. We used five different classes to classify the news articles which are as follows – Business, entertainment, politics, tech, sports. We used cross validation technique to divide the data as 4:1 and 3:2. We evaluated the news articles based on their expected classes and the classified classes.

**6. Experimental Results and Evaluation:**

We evaluated the results by processing the raw data using different strategies –

- Raw content data of the article
- Summarizing by including both title and content in the raw data
- Summarizing by including title and the whole content(including title) which lead to repetition of data
- Summarizing by including only the content(without title)
- Summarizing the data by including the content(without title) and the title(compulsory)

We used 2 different classification methods to classify the articles in the given classes –

- Support Vector Machine
- Naïve Bayes

Below are the results we obtained by cross validating the data by a ratio of 3:2-

**Raw Content (Naïve Bayes)**

| class | f1_score | precision | recall | support |
|---|---|---|---|---|
| business | 0.97 | 0.98 | 0.97 | 208 |
| entertainment | 0.98 | 0.99 | 0.97 | 137 |
| politics | 0.96 | 0.94 | 0.99 | 163 |
| sport | 0.99 | 0.99 | 1 | 222 |
| tech | 0.96 | 0.97 | 0.95 | 160 |
| | | | | |
| avg / total | 0.98 | 0.98 | 0.98 | 890 |

**Raw Content (Support Vector Machine)**

| class | f1_score | precision | recall | support |
|---|---|---|---|---|
| business | 0.91 | 0.89 | 0.94 | 209 |
| entertainment | 0.94 | 0.96 | 0.92 | 165 |
| politics | 0.91 | 0.92 | 0.89 | 170 |
| sport | 0.97 | 0.96 | 0.99 | 204 |
| tech | 0.91 | 0.93 | 0.89 | 142 |
| | | | | |
| avg / total | 0.93 | 0.93 | 0.93 | 890 |

**Summarizing by including both title and content in the raw data (Naïve Bayes)**

| class | f1_score | precision | recall | support |
|---|---|---|---|---|
| business | 0.84 | 0.77 | 0.93 | 216 |
| entertainment | 0.89 | 0.97 | 0.83 | 157 |
| politics | 0.84 | 0.85 | 0.84 | 155 |
| sport | 0.96 | 0.95 | 0.97 | 197 |
| tech | 0.89 | 0.96 | 0.82 | 165 |
| | | | | |
| avg / total | 0.89 | 0.89 | 0.88 | 890 |

**Summarizing by including both title and content in the raw data (SVM)**

| class | f1_score | precision | recall | support |
|---|---|---|---|---|
| business | 0.86 | 0.81 | 0.91 | 216 |
| entertainment | 0.89 | 0.96 | 0.83 | 163 |
| politics | 0.88 | 0.9 | 0.87 | 165 |
| sport | 0.97 | 0.97 | 0.98 | 214 |
| tech | 0.89 | 0.89 | 0.89 | 132 |
| | | | | |
| avg / total | 0.9 | 0.9 | 0.9 | 890 |

**Summarizing by including title and the whole content (including title) (Naïve Bayes)**

| class | f1_score | precision | recall | support |
|---|---|---|---|---|
| business | 0.87 | 0.8 | 0.95 | 217 |
| entertainment | 0.89 | 0.96 | 0.83 | 143 |
| politics | 0.87 | 0.87 | 0.88 | 145 |
| sport | 0.97 | 0.95 | 0.99 | 215 |
| tech | 0.87 | 0.96 | 0.79 | 170 |
| | | | | |
| avg / total | 0.9 | 0.91 | 0.9 | 890 |

**Summarizing by including title and the whole content (including title) (SVM)**

| class | f1_score | precision | recall | support |
|---|---|---|---|---|
| business | 0.84 | 0.77 | 0.92 | 203 |
| entertainment | 0.89 | 0.96 | 0.83 | 168 |
| politics | 0.88 | 0.9 | 0.86 | 176 |
| sport | 0.96 | 0.94 | 0.97 | 190 |
| tech | 0.89 | 0.94 | 0.85 | 153 |
| | | | | |
| avg / total | 0.89 | 0.9 | 0.89 | 890 |

**Summarizing by including only the content (without title) (Naïve Bayes)**

| class | f1_score | precision | recall | support |
|---|---|---|---|---|
| business | 0.9 | 0.85 | 0.96 | 201 |
| entertainment | 0.93 | 0.99 | 0.87 | 142 |
| politics | 0.93 | 0.92 | 0.94 | 163 |
| sport | 0.97 | 0.96 | 0.99 | 207 |
| tech | 0.9 | 0.96 | 0.86 | 177 |
| | | | | |
| avg / total | 0.93 | 0.93 | 0.93 | 890 |

**Summarizing by including only the content (without title) (SVM)**

| class | f1_score | precision | recall | support |
|---|---|---|---|---|
| business | 0.91 | 0.92 | 0.9 | 198 |
| entertainment | 0.95 | 0.95 | 0.94 | 150 |
| politics | 0.93 | 0.92 | 0.94 | 168 |
| sport | 0.98 | 0.97 | 0.98 | 214 |
| tech | 0.96 | 0.96 | 0.96 | 160 |
| | | | | |
| avg / total | 0.94 | 0.94 | 0.94 | 890 |

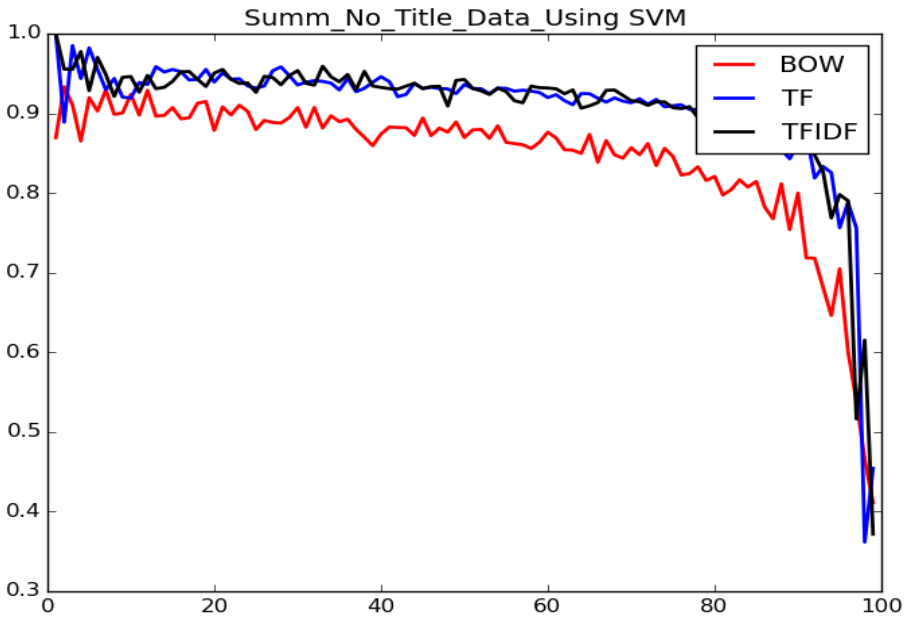
**Summarizing the data by content (without title) and the title exclusively (Naïve Bayes)**

| class | f1_score | precision | recall | support |
|---|---|---|---|---|
| business | 0.88 | 0.82 | 0.95 | 201 |
| entertainment | 0.93 | 0.99 | 0.87 | 146 |
| politics | 0.91 | 0.89 | 0.92 | 156 |
| sport | 0.97 | 0.98 | 0.97 | 217 |
| tech | 0.9 | 0.95 | 0.85 | 170 |
| | | | | |
| avg / total | 0.92 | 0.92 | 0.92 | 890 |

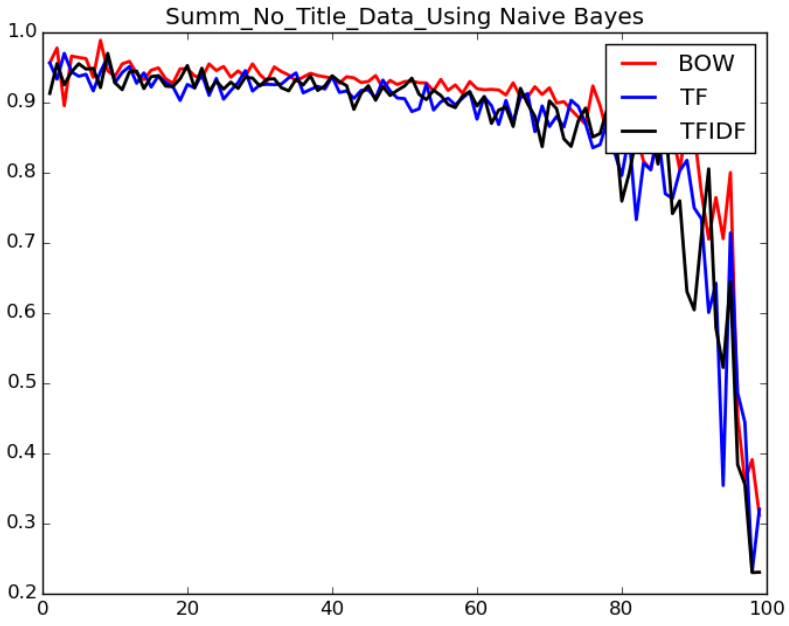
**Summarizing the data by content (without title) and the title exclusively (SVM)**

| class | f1_score | precision | recall | support |
|---|---|---|---|---|
| business | 0.91 | 0.89 | 0.94 | 209 |
| entertainment | 0.94 | 0.96 | 0.92 | 165 |
| politics | 0.91 | 0.92 | 0.89 | 170 |
| sport | 0.97 | 0.96 | 0.99 | 204 |
| tech | 0.91 | 0.93 | 0.89 | 142 |
| | | | | |
| avg / total | 0.93 | 0.93 | 0.93 | 890 |

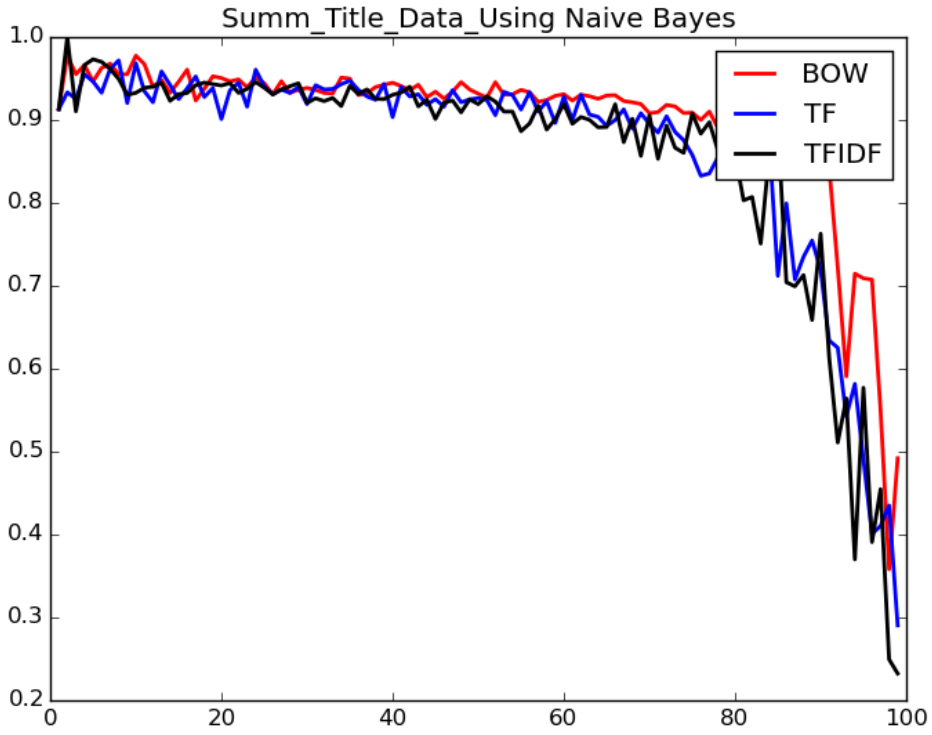
**Graphs**

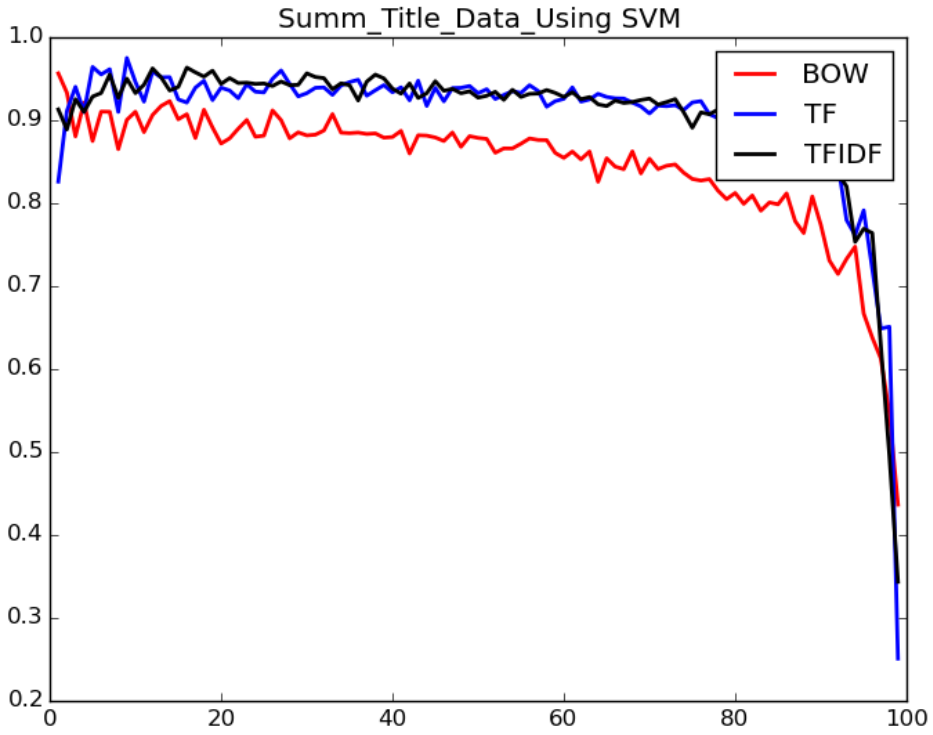



This is the graph of classifying the summarized data without including the title of the article. We can clearly see the superiority of the TF.IDF feature extraction method as compared to the bag of words model as the former method didn't depend only on the words of the content while the latter one does. This method of summarization brings about the contrast in the nag of words model and the TF.IDF model. Note that this is only with the SVM classifier and not with the Naïve Bayes classifier since the latter one complements the BOW model hence returning a greater accuracy in that case as given below



The same trend is followed by the summarization method including the title exclusively, where Bag of words model gives poor results with SVM classifier but better results with Naïve Bayes classifier in contrast to TF.IDF model of feature extraction as shown below



## 7. Conclusion and future work

Given the high dimensions of the data involved, text classification proved to be a challenging task. In this project, an approach was followed to classify news texts. 1) Pre-processing the raw data and cleaning the content of the articles 2) Processing the data and summarization of the data 3) Feature extraction from the raw data as well as the summarized data of various types 4) classification the vectorised documents (news articles) using SVM Classifier as well as Naïve Bayes Classifier 5) Evaluation of the results obtained by classification of the articles in 5 different classes of a)Sports b)Business c)Tech d)Politics e)Entertainment, by using the evaluation metrics – a) f1_score b) recall c) precision

We plan to use deep learning by using the popular model of **Hierarchical Attention Networks** to classify the documents/news articles based on the context/intent of the articles and not just by the term frequencies or the individual weights of the tokens in the article.

(i) It has a hierarchical structure that mirrors the hierarchical structure of documents

(ii) It has two levels of attention mechanisms applied at the word and sentence-level, enabling it to attend differentially to more and less important content when constructing the document representation. Experiments conducted on six large scale text classification tasks demonstrate that the proposed architecture outperform previous methods by a substantial margin. Visualization of the attention layers illustrates that the model selects qualitatively informative words and sentences.