INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 6)
Available online at: https://www.ijariit.com

Text classification of BBC news articles and text summarization using text rank

Abhishek Dutt
abhishekdutt1212@gmail.com
University of Massachusetts, Amherst

Kirk Smalley
kjsmalley@umass.edu
University of Massachusetts, Amherst

ABSTRACT

Automatic text classification is a crucial method to manage and process vast amounts of textual data in the digital form. This paper illustrates the text classification and text summarization using machine learning techniques and algorithms. An important part of this task is that the input data is in the form of natural language so we must do some preliminary analysis to transform the documents into a uniform structure so that we can train our data using numerical representations. Text Summarization is the process of obtaining the significant information from a text document. Information is extracted from the text document and the summarized report is created and displayed to the user.

Keywords: *Text rank, Text Summarization, BBC, BBC News, Text Classification, Machine Learning, Naive Bayes, Logistic Regression, SVM, Random Forest Classifier, Page rank*

1. INTRODUCTION

1.1 Text Classification

Automatic text classification has been a salient research topic and had important applications ever since the rise in digital documents. Text classification is now seen as a necessity due to the tremendous size of the text documents we see every day. A very significant application of text classification is topic-based classification. Intuitively, this is a task of classifying the text under a predetermined set of categories. If d_i is a set of D documents and there is a predetermined set of categories $\{c_1,c_2,c_3...c_n\}$, text classification then assigns one of the categories to c_j to d_i. Since a machine learning approach is used to perform classification, there is a requirement of an initial dataset that is used to perform training of the machine learning model. Figure 1 shows the graphical representation of the Text Classification process

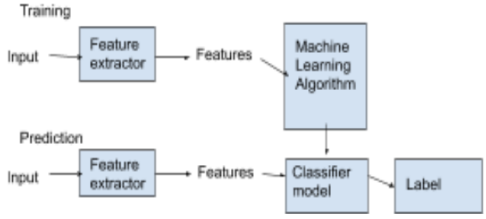

Fig. 1: Text classification process

One of the biggest issues in constructing a Machine learning classifier is the number of features that can easily reach very high numbers of tens of thousands, in turn making it extremely slow to compute the values of the labels.

To address this problem, there is a need of dimension reduction methods to be implemented. One of which is by selecting the subset of the original feature [1]. The other is by transforming the older features into new ones [2]. After which machine learning algorithms can be applied and then we can determine which algorithm produces better results. Typically, Support Vector Machines produce better accuracy and are more widely used for text classification. However, there are various other algorithms that can be used. A few of them being the Naive Bayes classifier, Random Forest classifier, and Logistic Regression which we implemented to compare the results.

1.2 Text Summarizer

Automatic Text Summarizer is a tool that helps us summarize the entire context and gives us a brief summary of the relevant and salient features of the context. The main advantage of a text summarizer is that the reading time of the user can be significantly reduced without decreasing the quality of the information absorbed. A marvelous text summarizer should reproduce the theme of the document efficiently by keeping repetitions to a bare minimum.

An extractive text summarizer attempts to summarize the articles by selecting the subset of the words that contain the most important information. This is done by weighing the important parts of sentences, ranking these sentences in the order of importance and similarity using the algorithm, and then selecting the sentences with the higher rank to form the summary. We use an algorithm called the TextRank algorithm which has some major advantages when it comes to text summarization that makes it the ideal choice for extractive text summarization [3].

- It is unsupervised, which means that it does not require training data and does not require any manually tagged corpus or other human intervention needed to produce actual data [7].
- It is language-independent which means that it works on almost all languages and is based on word concurrence [7].
- It is well developed and easily available for developers.

The diagram below (Fig. 2) shows the implementation of the TextRank algorithm.

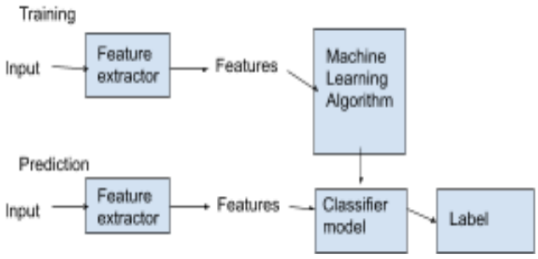

Fig. 2: Text Summarization process using TextRank

2. RELATED WORK

2.1 Use of Text Classification in Marketing Research and Medicine

Many fields now use automated text classification methods as there is a huge amount of textual data that they deal with. Marketing research uses automated text classification on a wide range of different research objectives. For e.g. how text classification is used to predict the defaults based on an online loan request [5].

Medicine research also uses text classification, for e.g. [6] where SVM is used for cancer genomic studies. Notably, many important factors like, data source, average text length, or classification task vary across all the studies. Thus, the strong reliance on a limited set of classification methods proves that computer science research on text classification performance has not been guiding the text classification choices. Rather an implicit assumption is seen to be followed that a method is similarly effective across all choices.

2.2 The popularity of the TextRank algorithm

Being a widely implemented algorithm for text summarization, it was not widely used in web content, while its application was limited to keyword and sentence extraction. It was shown in [7] that TextRank is an efficient method for text extraction and it does not require human supervision nor is it language-specific. It was shown to outperform many other supervised learning algorithms. However, a study conducted by [2] pointed out a more direct link between web content and text summarization algorithms, showing the effectiveness of an algorithm like TextRank.

3. DATA

3.1 Text Classification

For this research, we used the BBC world news dataset. The data consists of a total of 2225 text documents from the BBC world news website corresponding to the stories covered in 2004-2005. The total size of the data is around 5 MB.

The data is pre-classified into 5 different class labels. These class labels are Business, Sport, Tech, Entertainment, and Politics. It contains 510 text documents from the business category, 511 text documents from the sports category, 401 text documents from the tech category, 386 text documents from the entertainment category, and 417 articles from the politics category.

There are a total number of 456,542 words in all the documents while containing only 29,126 unique words. This means that on average every word appears average about 15 times throughout the whole corpus. The average document length is 205 words. Note however that all these statistics are reported after the removal of all stop words. Each document in reality is about 1500 words long so clearly stopwords comprise a great amount of the document.

3.2 Text Summarization

For text summarization, we used 30 texts collected from different web pages. These 30 texts were divided into 4 main categories.

- Finance (8 texts)
- Short stories (6 texts)
- Politics (8 texts)
- Entertainment (8 texts)

These texts were used to test how well the text summarization performs. There was no training data since TextRank does not need any training data.

4. METHODS

4.1 Text Classification

Text classification is a wide field. There are various methods that can be used to perform text classification. The diagram below shows the representation of different methods of text classification. It has always been a very important application and research topic. It is broadly divided into two sub-categories, Statistical and Machine Learning. There are many machine learning techniques that can be used for text classification, which are further divided into supervised, semi-supervised, and unsupervised learning methods. As shown in the diagram below.

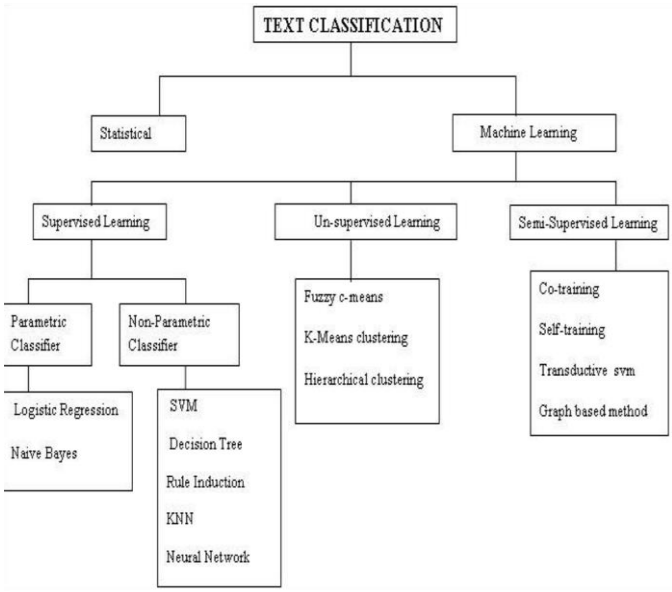

Fig. 3: Representation of different classifiers

Out of these many different techniques of text classification, we focus on Supervised Machine Learning techniques and pick two techniques each from parametric classifiers and non-parametric classifiers. We use both Logistic Regression and Naive Bayes classifiers as parametric classifier models and Support Vector Machine and Random Forest Classifiers as non-parametric classifiers.

To tokenize each document from the data set, we used the Sklearn CountVectorizer class which delimits tokens by spaces, removing all punctuation and non-ASCII characters. We also removed all the stop words using the stop_words parameter. Stop words are the most commonly seen words in a language, whose occurrence does not affect the model. We then implemented a BOW logistic regression classifier that directly computes the posterior probability of a class given the

document by associating features of the document with the labeled class. This is one of the supervised learning techniques as previously discussed since we had an output to adjust the weight values accordingly. Each class has the same type of weights associated with the common features for each class but each one has different weight values that best predict that class. When fitting the training data the classifier looks at the class label and weight values. Those weight values are then tuned accordingly by comparing the features of the document with the label so that the weights may better predict the given class. The table below shows the highest weighted words for all the five different categories

Table 1: List of top-weighted words in each category

Business	Politics	Tech	Sport	Entmt.
saudi	blunkett	mobile	athletic	bbc
higher	government	sony	race	album
market	mr	user	season	films
business	mps	games	coach	band
economy	committee	digital	rugby	star

Using K-fold cross-validation we split our training data into 5 separate groups and compute metrics separately for each subset. We used the CountVectorizer class to represent each document as a vector alongside each index containing word counts for every term in the vocabulary, giving us the BagOfWords representations of our data. We used the transform function to aggregate our training data samples from vectors into a matrix that can be fitted with the labeled test data to return the coefficients for the word vectors of each class. Using these we then predict the probability class for each instance of the test data. The total of these 5 probabilities is then returned, which represents one probability for each class.

The category which has the highest probability is then chosen to be the 'predicted' class. A similar strategy was adopted for other algorithms of text classification, i.e. Naive Bayes, SVM, and Random Forest Classifiers.

4.2 Text Summarization

The implementation of TextRank used is dependent on the similarity of sentences in the text that needs to be summarized. Initially, we start off with cleaning the text by removing the stop words, un-capitalizing all the letters, and removing any non-alphanumeric letters. We then split the text document into individual sentences and then split the sentences into words. After this, we create the similarity matrix which stores values of similarity between all sentences. We then use the PageRank algorithm to compute a score known as the page rank score and then rank the sentences according to this rank score. The PageRank algorithm is similar to the TextRank, the difference being that PageRank is typically used to rank web pages in an online search result. After the sentences are ranked, the desired number of sentences then form the summary.

Initially, only nouns and adjectives are considered by assigning the parts of speech to words. From this, a graph is created, with the words being the nodes. These nodes are then connected if the words are similar, or close in text. After creating the graph, the algorithm is run on the graph. It iterates over the nodes which are initially given equivalent weights of 1. The influence is calculated by summing up the value of the connected vertex. Scores are then normalized, and the algorithm repeats until the scores remain constant. This is the process of obtaining the score.

4.3 Document Clustering

4.3 1 Document Similarity: Another task we wanted to look at is the unsupervised learning task of clustering the BBC articles by similarity solely based on their content without consideration of their labels. The purpose of this is to measure the similarity between the documents in terms of their word counts to get a better sense of which documents may share similar content without being directly related to each other on a general basis. Consider for example an article concerning technology in general that contains a section dedicated to realistic sports video games. Also, observe a sports article strictly dedicated to the history of baseball. The former document would most likely be labeled under technology while the latter sports, and although the general topics of the two articles are different, they do share mutual information surrounding sports.

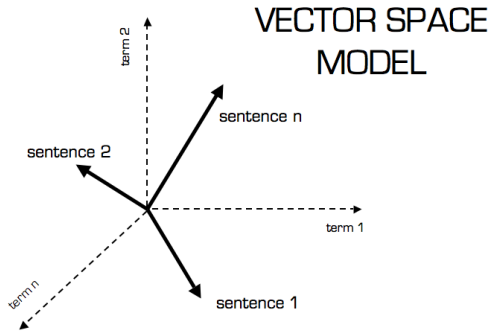

Fig. 4: Vector Space Model (VSM)

This similarity is captured in the BagOfWords representation of our sample vectors where both documents are more likely to share similar words relating to the commercial side of the technology topic that may specifically mention words relating to sports enterprises. Using the vector space model where each word feature count is a dimension and each document is a vector in this space, the words pertaining to the sports will geometrically be closer to each other in this space.

4.3.2 Common Terms Amongst Clusters

Once we have clustered the documents using the Vector Space Model of similarity and comparing the proportions of categories found amongst each cluster, we are specifically more interested in what these categories of each cluster have in common and how they contrast with the other clusters. One interesting observation was that the word said was not included in the stop words and being that it is a pretty frequent word it was the number one for several of the lists, so we decided to remove it. Looking at Cluster 1 and 2 we confirm from our latter results the similarity between the Politics and Business categories with the word 'mr' being found amongst the most frequent terms for both categories of both clusters exclusively from the other 3 topics. Particularly note in Cluster 2 such the high occurrence of the word for the Politics category. Unsurprisingly for Clusters 1, 2, and 3, we find words like ''song', ''music', and 'film' while the Sports category for example sees somewhat apparent segregation for general and specific topics. Clusters 1 and 2 mention proper locations such as 'wales' and 'liverpool' while the third mentions more general words such as 'game' and 'year' which are also found at the top of the lists for the other categories as these are fairly general terms that concern matters not pertaining to super specifics of sports and are more likely to be more topically relevant to other categories. We can imagine Sports articles in Cluster 3 are more likely to be similar on a sentence level structure to the other categories as these ones contain more words that could be associated with other topics.

5. RESULTS

5.1 Text Classification

Logistic regression was initially used on the dataset to perform text classification. We used K-fold cross-validation; we split our training data into 5 separate groups and compute metrics separately for each subset. After performing the task of classification, we were able to test the accuracies and other performance metrics, namely, precision, recall, and F1 score. Logistic regression performed incredibly well even though it is a parametric classifier. We got an impressive accuracy of 98.2926% when we tested this algorithm on the testing dataset. This accuracy represents the average of the five splits of data. The table below shows the performance of the logistic regression in the other performance metrics.

Table 2: Precision, Recall, and F1 for all categories

Categories	Precision	Recall	F1
Business	0.96675	0.96835	0.96732
Politics	0.97200	0.96448	0.96815
Sports	0.98844	0.99808	0.99322
Tech	0.97330	0.97271	0.97286
Entmt.	0.97558	0.97401	0.97473

We can conclude from the results that Logistic Regression was able to classify new articles quite successfully.

Furthermore, we then went on to test the other algorithms and their accuracies and found out that Support Vector Machine performed just better than Logistic Regression which was followed by Naive Bayes and Random Forest Classifiers had the least accuracy as shown in the table below.

Table 3: Accuracies of different text classification models

Model Name	Accuracy
SVM	98.697%
Logistic Regression	98.293%
Naive Bayes	97.078%
Random Forest Classifier	83.057%

5.2 Text Summarization

We had 4 evaluators read all the text documents, then read the summarizations produced by TextRank and then rate the importance of the information displayed. As can be seen in the table below, we found that summarizations of short stories performed the worst.

Table 4: Average evaluation scores for text summarization

Category	Evaluation score
Politics	72
Finance	78
Short Stories	32
Entertainment	66

We could conclude from using TextRank that it is indeed a great tool for text summarization, and it has a lot of advantages as mentioned above in 2.4. We can also conclude that the role of the number of sentences to be displayed in the summary is critical. There are various passages especially stories with which TextRank does not perform as well simply because it is an extractive summarizer and not abstractive and it is not possible for a lot of passages to be summarized using this technique as extractive summarization hampers the flow of the passage of text. Our results were confirmed through our findings that the summarization performed significantly worse with the short story text documents and evenly with the other categories.

5.3 Clustering

From clustering these documents out of the BBC news dataset, we could represent all as vectors in Euclidean space and used scikit-learn's K-means clustering library to formulate clusters that partition the samples. Next from viewing a single cluster, we could then look at the corresponding labels and make inferences regarding the proportions of categories among the clusters.

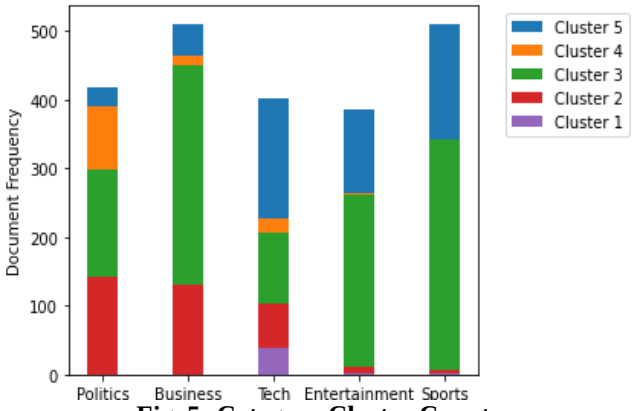

Fig. 5: Category Cluster Counts

One particularly noteworthy observation is how the political articles are distributed amongst the clusters. Notice the orange Cluster 4 is mostly just composed of this one category, indicating that the politics article has words that use exclusively from all the other categories. Another observation is the similarity between the business and politics documents which almost exclusively comprise the red Cluster 2. Note also that the largest cluster comprising the most documents is the green Cluster 3 which is most likely to be made of the more general documents that do not have high frequencies of unique words.

Table 5: Cluster 1

Category	Term	Frequency
Tech	game	214
	technology	141
	mobile	104
Entmt.	song	81
	best	60
	music	58
Sports	new	34
	wales	23
	game	19

Table 6: Cluster 2

Category	Term	Frequency
Politics	mr	669
	party	264
Business	year	276
	mr	191
Tech	use	172
	people	131
Entmt.	music	103
	urban	52
Sports	liverpool	28
	gerrard	26

Table 7: Cluster 3

Category	Term	Frequency
Politics	mr	531

Politics	govern	271
Business	year	614
	company	411
Tech	use	139
	software	138
Entmt.	film	620
	year	413
Sports	game	422
	year	415

6. DISCUSSION AND FUTURE WORK

6.1 Text Classification

Text classification holds very high importance as an Artificial Intelligence research topic. However, there are still various questions that need to be answered or even addressed regarding this field of text classification.

- Which feature is both high performing and scalable?
- How much of a difference does different training corpus make?
- How to reduce dimensionality over large training corpora.

There is also the problem of polysemy, which means that a word can have two or more different meanings. We need to find a way around this problem of polysemy to increase the accuracy of text classification algorithms. We hope to find a way to counter this problem by predicting the meaning of the word using the other words in the sentence and storing that along with the word.

6.2 Text Summarization

In our future work, we plan on expanding our research by increasing the number of documents tested as well as increasing the categories of the documents. We would like to include a percentage parameter that outputs close to a particular percentage of text and then compare the results of different texts with different lengths of summaries. We also hope to come up with better techniques to evaluate the efficiency of the summary as currently there was a big difference between the evaluation scores of the different evaluators. In future, we wish to automate the evaluation process so that the cognitive biases are not present.

7. REFERENCES

[1] Brank J., Grobelnik M., Milic-Frayling N., Mladenic D., “Interaction of Feature Selection Methods and Linear Classification Models”, Proc. of the 19th International Conference on Machine Learning, Australia, 2002.

[2] Han X., Zu G., Ohyama W., Wakabayashi T., Kimura F., Accuracy Improvement of Automatic Text Classification Based on Feature Transformation and Multi-classifier Combination, LNCS, Volume 3309, Jan 2004, pp. 463-468

[3] Balcerzak, Bartłomiej & Jaworski, Wojciech & Wierzbicki, Adam. (2014). Application of TextRank Algorithm for Credibility Assessment. 451-454. 10.1109/WI-IAT.2014.70.

[4] Ikonomakis, Emmanouil & Kotsiantis, Sotiris & Tampakas, V.. (2005). Text Classification Using Machine Learning Techniques. WSEAS transactions on computers. 4. 966-974.

[5] Netzer, O., Lemaire, A., & Herzenstein, M. (2019). When Words Sweat: Identifying Signals for Loan Default in the Text of Loan Applications. *Journal of Marketing Research*, *56*(6), 960–980.

[6] Huang, S., Cai, N., Pacheco, P. P., Narrandes, S., Wang, Y., & Xu, W. (2018). Applications of Support Vector Machine (SVM) Learning in Cancer Genomics. *Cancer genomics & proteomics*, *15*(1), 41–51. https://doi.org/10.21873/cgp.20063

[7] Mihalcea, Rada. ”Language independent extractive summarization.” Proceedings of the ACL 2005 on Interactive poster and demonstration sessions. Association for Computational Linguistics, 2005.

[8] Huang, Hongzhao, et al. ”Tweet Ranking Based on Heterogeneous

[9] Networks.” COLING. 2012.

