International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

# An Apriori Method for Topic Extraction from Text Files

**Anil Kumar K. M, Ajay B, Shashank R, Amogha Subramanya D. A**

***Abstract:*** ***In this data age peta-bytes of data is generated every day. One of the biggest challenge today is to convert this data into useful information, this is known as data mining. Important kinds of data include text-based data, audio-based data, image-based data, video-based data etc. An important challenge in mining useful information from text-based data source (text mining) is topic modeling which is to find out the topic the text is talking about. The solution to this problem finds application, in clustering files based on the topic, pre-processing method in information retrieval, ontology of medical record etc. A lot of research work has gone into this area of topic modeling, and many approaches have been formulated. Some of these approaches take into account the occurrence and frequency of occurrence of words/terms, these models come under the Bag Of Words(BOW) approach. Others take into account the underlying structure in the corpus of text used, Wikipedia category graph is an example of this approach. This paper, provides an unsupervised solution to the above problem by extracting keywords that represent the topic of the text document. In our approach, topic modeling is carried out with a hybrid model which makes use of WordNet and Wikipedia Corpus. Promising experimental results have been obtained for well- known news dataset (BBCNews) from our model. We present the experimental result for our proposed approach along with the results of others in the same domain and show that our approach provides better results.***

***Index Terms*****: *Bag of Phrases, Cosine similarity, Key Phrases, Occurrence Matrix, Keyword Extraction.***

## I. INTRODUCTION

A text file is a kind of computer file that is structured as a sequence of lines of electronic text. A text file is a file that only contains text and has no special formatting, images, etc. Because of their simplicity, text files as well as web pages are commonly used for storage of information. Increasing amount of storage capacity has lead to generation of lot of text based files. So there has risen a need for automating the process of extracting topics from pieces of text. Some of the important areas where it is most required are in medical and law fields. In case of hospitals, hundreds of new patients are admitted daily. Hospital management maintains their records pertaining to their disease or any lab reports. In such situation, certain analysis can be done regarding the disease by going through the reports.

**Revised Manuscript Received on 30 July 2019.**
***** Correspondence Author
**Anil Kumar K M**, Department of CS & E, JSS Science and Technology University, Mysore, India
**Ajay B,** Department of CS & E, JSS Science and Technology University, Mysore, India.
**Shashank R**, Department of CS & E, JSS Science and Technology University, Mysore, India
**Amogha Subramanya D A,** Department of CS & E, JSS Science and Technology University, Mysore, India.



Analysis of reports requires to know what the report is saying. In other words, analysis of a report requires extracting major points or keywords from it. For example, if the extracted keywords are high fever, cough, low appetite, confusion and profuse sweating, then the doctor can immediately conclude that the patient has Pneumonia. Similarly, in a court hearing, if the present case file gives keywords such as conspiracy and robbery, the accused can be sentenced to 5-10 years of jail time. As easy as the process of analyzing a text file gets, extracting keywords is a difficult job to do manually. Hence we present an unsupervised or automated approach to extract keywords. Hence by storing only keywords as meta data in a database rather than the whole report not only saves the storage space but also helps in transforming an unstructured data to a structured data. For almost all the words we search on the internet, there happens to be a Wikipedia page on that word. So it can be recognized that Wikipedia[5] has almost all the words that might essentially be candidate keywords and hence it is easy to get information about those words. So we have made use of Wikipedia corpus to extract keywords. *Topic modelling* is a process of automatically identifying topics(important keywords) present in a text object and to derive hidden patterns exhibited by a text corpus[1]. The statistical model that performs this task is called *Topic model*. This can in turn aid as a tool for automatically grouping different text files based on the topic they talk about. A text can come from any kind of writing. Letters, adverts, user-guides, emails, notes, research articles, medical reports etc., are all different types of texts. All these text files can be differentiated by what they speak about, that is their topic. This rises a need for an automated tool that can predict topics of text files and providing enough information to cluster these files. Our goal is to provide an automated tool that can fairly predict the topic of a given piece of text without any prior knowledge about the file.

The main features of the text files considered in this paper are:

1. The text data comprises of plentyof topics. It is better if the model can handle diversity of the topics.
2. The text speaks about at most one topic (mostly).

The text files we considered are those from news dataset[20]. The reason being that news is a kind of data that satisfies the above two requirements and in cases where feature 2 fails, we predict all the eligible candidate topics as keywords.

Our approach mainly focuses on obtaining semantic similarity on a set of important words known as bag of words (BOW) and further reducing the BOW based on the similarity measure. Finally the reduced set of words indicate the words that are likely to be the topic of the text.

The rest of this paper is organized as follows. Section 2 discusses the related works on the subject. In section 3, we explain our methodologies used to solve the problem. We present of experimental setup in Section 4 and results are discussed in section 5. Finally, conclude and future work is discussed in section 6.

## II. RELATED WORKS

Many researchers have been working around topic modelling problem from quite a long time. This section discusses the literature survey.

### A. *WordNet*

WordNet[2] is made up a database of synsets or synonym set which are set of words carrying unique meaning. It was developed by Fellbaum in 1998. Concepts represented by these synsets are defined in a glossary for each synset. Explicit semantic relations are defined in WordNet which connect these synsets. For example, the word pair "cat" and "mouse" have higher semantic similarity than "cat" and "car".

*1) Keyword extraction using lexical chains:* This is an attempt made by Ercan and Cicekli in 2007 [3]. The semantic content of a portion of text is represented by a set of semantically related words of text known as lexical chain. In creation of these lexical chains, the WordNet's synsets, hyponym, meronym trees are used in order to find relations between two word senses. The lexical chain is made up of nodes that represent a word sense of a word, and links represent the relation between two word senses, which can be synonym/reiteration, hyponym/hypernym, or meronym. The figure Fig (a) represents a simple lexical chain that shows relations among the selected word senses. In Fig (a), the solid lines represent hyponym relations, and the dotted lines represent synonymic relations. .

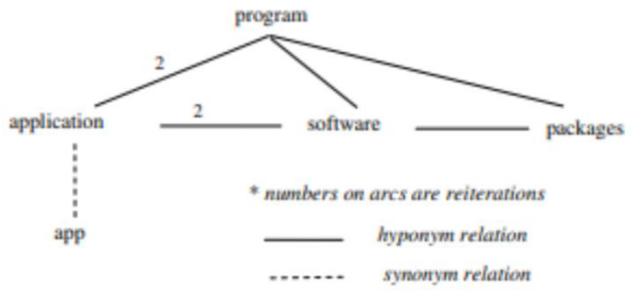


**Fig (a). A Simple Lexical Chain**

### B. *Wiki Relate*!

*WikiRelate!* was created by Ponzetto and Strube [4]. It makes use of the category structure of Wikipedia. It works as follows: for any two words $w_1$ and $w_2$, semantic relatedness $S(w_1, w_2)$ is obtained as follows; pages corresponding to the words $w_1$ and $w_2$ are retrieved. Next, running through the category tree, the categories to which pages belong to are extracted. The extracted pages and the corresponding paths connecting the taxonomy form the basis for computing the semantic relatedness $S(w_1, w_2)$ between the words $w_1$ and $w_2$.

*1) Page retrieval and disambiguation:* Page retrieval of a page $P_i$proceeds as follows: First, the page titled $W_i$is queried. Following this, all the redirects are followed. Since Wikipedia returns a page called disambiguation page they have to be resolved. The disambiguation page consists of links to candidate targets for a given query. The disambiguation of the page $P_i$ for $W_i$, begins first by obtaining all the hyperlinks in page $P_j$ for $W_j$ without disambiguating. For the cases where both the queries are ambiguous, the word $W_j$ and all the links of the page $P_j$ are considered for disambiguation purpose. On occurrence of any disambiguation term in one of the links in $P_i$, the target page is returned; otherwise, the first link in the disambiguation page is returned. This strategy though less accurate, than when all links in the disambiguation page are followed, is a more practical solution since disambiguation pages contain large number of links.

*2) Category tree search:* For computing measures based on path and information content, the category tree paths are necessary. The categories $C_i$ and $C_j$ to which the pages $P_i$ and $P_j$ belong to are extracted. These links in pages denote the primitive concepts the word represents. A depth-limited search with a maximum depth of 4 is performed for a least common subsume for each category in the category list. It is noted that the results are better when the search is limiting this is the consequence of the strongly connected nature of the Wikipedia Category Graph (WCG).

*3) Computation of Relatedness measures:* The taxonomy measures such as information content and path measures are used. These are computed using the set of paths found between the category pairs. [8]

### C. *Explicit Semantic Analysis (ESA)*

Explicit Semantic Analysis (ESA) was created by Gabrilovich and Markovitch [5]. Their work associates semantic interpretations to both words and text fragments. They propose an association based method which uses text features and links within articles in Wikipedia to aid their model. The main assumption is the availability of vectors of basic concepts, $C_1$, $C_2$ ... ,$C_n$. A text fragment $t$ is represented using weights $w_1$,$w_2$ ...,$w_n$ here $w_i$ denotes the associative strength between the fragment $t$ and the concept $C_i$. The weight associated vector are the semantic interpretation vector of the fragment $t$.

Index documents used in this work is the Wikipedia articles. The concept $C_i$ is represented as word vectors. The word vectors are obtained by assigning a weight value using tf-idf scheme [6].

Next an inverted index containing a mapping from word to list of concepts they appear in is built. Those indicies that show insignificant association between words and concepts are pruned. They make use of cosine metric to compute semantic relatedness of a pair of words. This cosine metric makes use of the vectors associated to each word, from the inverted indices.

### D. *Latent Semantic Analysis (LSA)*

LSA [7] proposed a statistical technique of modelling semantic relationship between words which is influenced by the co-occurrence information of words from a large unlabeled corpus of text. Originally LSA (also known as Latent Semantic Indexing or LSI) was proposed for Information Retrieval (IR), i.e., for a given query searching and selecting the most related document from a large database of documents.

LSA stands out unique among others as it does not use human-organised knowledge, instead it learns from a mathematical model obtained by applying SVD (singular value decomposition) over a w-d (word-by-document) co-occurrence matrix. LSA is essentially identifies the important dimensions in the data. Identifying these dimensions is a dimensionality reduction scheme. These prominent dimensions correspond to the "latent concepts" hence the name latent semantic analysis. [5] Firstly, a huge collection of domain relevant text is obtained which is further divided into "documents". For many cases, paragraphs act as individual separate documents since it is assumed that the information within them are coherent and related. Next, a matrix of documents versus terms is created, this is called a co-occurrence matrix. The cells in this matrix denote the term frequency of the term in the corresponding document, i.e. *M (i , j) = number of times the term i occurs in document j.* A term is a token or word that occurs multiple times in a document. Term does not use stemming or other schemes to combine different forms of a word. The co-occurrence matrix of size *m* x *n* can be thought of as vectors of m and n dimensions for each term and document respectively. The individual components of vectors are usually weighted to minimize the effect of stop words that occur throughout the corpus. A common weighting method is the tf-idf scheme that multiplies the term frequency with the information gain also called inverse document frequency. Up on applying SVD with a specified parameter k which denotes the number of dimensions, three matrices are produced. These matrices containing k-dimensional vectors are usually denoted as U. $\sum .V^T$ here, U and $V^T$ are two different vector spaces and $\sum$ is the singular values that can be used to transform the vectors from one space to another. U contains vectors for documents in the corpus, and $V^T$ contains those for terms. The usage of U and $V^T$ depends on the application.

### *E. Rapid Automatic Keyword Extraction (Rake)*

RAKE was developed by Rose, Engel and Cramer [9] as an alternative for corpus-based keyword extraction methods. RAKE operates on the proposition that keywords and key phrases do not overlap with stop words, and that the more non-stop words are found adjacent to each other, the higher the probability that it is a key phrase [9] whereas the traditional methods considered word frequencies. It means that words with longer length have high relevance scores than words that are short. So, a word machine learning will have a higher relevance score than learning, and a word minimal supporting sets will have a higher relevance score than supporting sets. Therefore, the RAKE method is highly dependent on a reliable and comprehensive list of stopwords, because the list deter- mines where the text is divided up and where the boundaries of candidate keywords are, which decides the candidates length [10].

Scores are computed for each individual word and the score of a sequence of words is the sum of the scores of individual words contained in that sequence. Degree score is the sum of lengths of all sequences where the word appears. A graph of word co-occurrences is made, where a word is considered to co-occur with another if they are found in the same phrase. So if a word minimal appears in a bigram minimal sets and a trigram minimal supporting sets, the degree score for word minimal is (2-1)+(3-1)=3;the -1 on each side accounts for the number of words minimal occurs with, not the number of words in the sequence. Thus, degree score favours words that occur often in longer candidate keywords [9]. Frequency score is the number of times the word occurs in the whole text. According to the paper by Rose et al. [9], each individual word score is calculated by dividing degree by frequency, that is degree/frequency.

### *F. Maui (Multi-Purpose Automatic Topic Indexing)*

Maui is dependant on the Weka machine learning toolkit [11], developed by the Machine Learning Group at the University of Waikato. Maui was developed by OlenaMedelyan [13] as a replacement or improvement of Kea, the well-known Keyword Extraction Algorithm [14] which is also developed by the Machine Learning Group. Kea is an algorithm for extracting keywords or key phrases from a document. It can be considered a supervised machine learning method, as it trains a model based on manually labelled documents, and predicts keywords from unseen documents[10] Maui is an open-source Java implementation built on Kea, but hassignificant additions and improvements. Kea is only useful for keyword extraction, which is one kind of topic indexing, while Maui can perform more kinds of topic indexing, including term assignment from a controlled external vocabulary, and mapping keywords to terms in Wikipedia[10]. Also, Maui contains the complete machine learning library of Weka, whereas Kea only contains a few of Wekas classes. Finally, Maui includes the Jena software library, for incorporating external controlled vocabularies, and Wikipedia Miner [10].

## III. METHODOLOGY

### *A. Procedure*

We use unsupervised learning approaches for extracting the relevant topic from the Text files using their information. Machine Learning is a sub-field of computer science that evolved from the study of computational learning theory and pattern recognition in artificial intelligence. [12] Unsupervised Learning is a branch of machine learning which is used to infer a function that describes the hidden structure of unlabelled data.

*1) Pre-processing the data set:* The input text files are pre-processed to remove unwanted words(stop words) and characters(special symbols). Articles from BBC News contain time-stamps and other unnecessary junk. A script is generated to remove these junk from the test files.

*2) Feature Extraction:* Feature extraction involves the following:

*a) Parts Of Speech tagging:* We have used Stanford POS Tagger[21] to extract noun-noun, noun- adjective, noun-verb sets apart from nouns , verbs and adjectives. For example, consider the following sentence.

"My cat also likes drinking milk"

The tagged version of the above title is:

```
        "My                 PRP$
Cat         NN
Also        RB
Likes               VBZ
Drinking    NN
Milk        NN"
```

where,
PRP$- Possessive pronoun
RB Adverb
NN- Noun, singular or mass
VBZ Verb, 3rd person singular present

*b) Importance of Noun-Noun/Adjective-Noun pairs (Bigrams) in determining the Context:* Context is one of the major impactful factors in determining the bag of phrases(BOP) of given text. However, we can define classification of a paper as assigning keywords to a paper from a given set of keywords. (Classes) For example, consider the following sentence. "Having a controlled vocabulary of keywords fixed, we can define classification of a paper as assignment of keywords." These words – 'vocabulary' and 'controlled' give a different meaning when used in different contexts. But when its used together, they give a holistic view of what the author says. In another way, the word 'controlled' is an adjective qualifying the word 'vocabulary'. Previous research work has proved that adjectivesare goodindicators of evaluative and subjective sentences. We also have Noun-Noun pairs which have different behaviours when used in different context. For example: A black hole is a place in space where gravity pulls so much that even light cannot get out. In this sentence, the noun-noun pair 'black hole' gives us an information prompting us to consider that the words used are in a context related to 'astrophysics'. However, the words – 'black' and 'hole' have completely different meaning in different contexts. We consider these pairs (Bigrams)in building our Bag Of Phrases.

*c) Deducing Bag Of Phrases (BOP):* We calculate the frequency of occurrence of each word from the pre-processed text file and decide on a threshold based on the file size. The word that qualify the required threshold are added to bag of phrases(BOP).

*3) Stemming:* In information retrieval and linguistic morphology, stemming is defined as the process of reducing inflected (or sometimes derived) words to their word stem, base or root for. The stem need not have to be similar to the root of the word. It is not necessary for the stem to be present in a valid root, it is sufficient that related words map to the same stem. Methods for stemming have been studied in computer science from the 1960s. A stemmer for English, for example, should identify the string "dogs" (and possibly "doglike", "doggy" etc.) as based on the root "dog", and "branches", "branching", "branched" as based on "branch". A stemming algorithm reduces the words "fishing", "fished", and "fisher" to the root word, "fish". On the other hand, "argue", "argued", "argues", "arguing", and "argus" reduce to the stem "argu" (illustrating the case where the stem is not itself a word or root) but "argument" and "arguments" reduce to the stem "argument".

We have made use of Porter's Stemming algorithm[22]. In the latter shown example, the word "argu" makes no sense for Wikipedia as it expects "argue". So we have extracted Did You Mean? words fromwww.dictionary.com website using Jsoup.

*4) Reduction of BOP using WordNet:* .WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. WordNet can thus be seen as a combination of dictionary and thesaurus. Nouns, adjectives, verbs and adverbs are grouped into sets of synsets, each expressing a distinct concept. Synsets are connected to each other by means of lexical relations and conceptual-semantic. WordNet resembles a thesaurus, where it groups all the words together based on their meanings (semantic similarity). The Words in BOP are matched against each other to calculate the semantic similarity between them . If it is found to be greater than a threshold set by us, the words are merged into a dictionary and their frequencies are added up.

*5) Reduction of BOP using Wikipedia:* Wikipedia is a free online encyclopedia, created and edited by volunteers around the world and hosted by the Wikimedia Foundation. We have use Wiki API[16] to extract information about the non reducible words in the WordNet phase. We extract the catagories for each of the words in the BOP from the last phase. Categories are meant to group pages in wikipedia together on similar subjects. They are implemented by a MediaWiki feature that adds any page with a text like [[Category:ABC]] in its wiki-markup to the automated listing that is the category with name ABC. Categories help readers to find, and navigate around, a subject area, to see pages sorted by title, and to thus find article relationships. We then extract the Wikipedia pages of these categories and then for each page, we calculate the frequency of the words(and its synonyms, extracted using JSoup) in BOP (reduced) and create an occurrence matrix. Occurrence matrix is a matrix whose rows consists of all words of BOP and the columns consist of the pages extracted from wikipedia for all these words. Each entry in the Occurrence Matrix Fij corresponds to the frequency of the word Wi in in the page Pj. We then calculate the cosine similarity between each row and then reduce the number of rows and hence the BOP using a threshold.

*6) Extraction of Topic From the input file:* The topic of the text files that contain only one word can be found out promisingly by the previous phase. But a problem occurs when the topic is in the form of a sentence. So we designed an algorithm based on scoring method that recognizes a sentence the is close to the topic using the text file. Here we have used the text file as a whole without considering the paragraphs as different.

***B. Tools:***

*1) WS4J(WordNet Similarity for Java) Package:* WS4J provides a pure Java API for several published algorithms to measure the semantic similarity/relatedness between words.

*2) Wiki Java Package::* Wiki Java is a Java Package that makes it easy to access and parse data from Wikipedia. Wiki- Java wraps the MediaWiki API.[16]

*3) JSOUP::* Jsoup is a Java library used to work with real-world HTML. It provides a very expedient API for extracting and also manipulating data, using the best of CSS, DOM, and jquery-like methods. Jsoup is developed to deal with all varieties of HTML

found in the wild. jsoup will create a sensible parse tree from pristine and validating, to invalid tag-soup.

## IV. EXPERIMENT

*1) Data collection:* The data collected consists of various topics from various domains. The data collected includes articles on various topics, the BBC News dataset and other general text of strings. Overall the number of text files considered is 700.

*2) Processing of Data:* Pre-processing the data included the following task in order:

a. Stop words removal
b.POS tagging and extraction of bigrams
c. Porter stemming

The pre-processed words is then subjected to the bag of phrases.

*3) Topic modeling using WordNet:* The resulting bag of words is processed as follows:

a. We use the WordNet reduction algorithm .The similarity is calculated for each pair of words in the bag of words.
b. After scoring the words were reduced based on similarity
c. The reduced words are put to a hash map containing the <word, frequency> pairs and sent to Wikipedia-model for further reduction

*4) Topic modeling using Wikipedia:* The hash map of <word, frequency> obtained from the WordNet modeling is further reduced using this model. The steps carried out include the following.

a. For every word in the hash map, we extract the "did you mean?" words from Wikipedia. This is done because, the words obtained after stemming might not have a proper meaning. For example: the word argue is reduced to argu through stemming which has no meaning, hence we extract the "did you mean?" words from the Wikipedia.
b. Next for every word we extract the page content, and enter down the frequencies in a matrix called occurrence matrix.
c.Let W1, W2 W5 denote the words P1, P2 P10 denote the pages Each entry, Wi, Pi =frequency of word Wi in page Pi.
d.From the above occurrence matrix, we calculate the similarity among the words using the equation (1).

1.For every pair of words in the occurrence matrix Wi , Wj we treat their row entry to be vectors Vi,Vj.
2.The cosine similarity between two vectors is defined as the cosine of angle between the vectors. Hence,

$$\text{Cosine sim}(Vi,Vj) = (Vi.Vj)/(|Vi| * |Vj|) \qquad \text{Equation (1)}$$

## V. RESULTS

We applied the above described methods on BBC News dataset and we compared our results with that of RAKE-Rapid Automatic Keyword Extraction[9] and the results are shown in the Table 1. The parameter we considered to compare our result is F1-score.

**Table 1. Comparison of results**

| Model | Avg. Precision | Avg. Recall | Avg. F1-score |
|---|---|---|---|
| Baseline (BOW) | 0.2406 | 0.238 | 0.2392 |
| RAKE | 0.3951 | 0.2507 | 0.3056 |
| Our Model | 0.6163 | 0.668 | 0.686 |

*F-score* is dependent on two measures *precision* and *re-call*, which in turn depends on the following measures of a *Confusion matrix*.

*Confusion matrix:* A confusion matrix is a table that is used to describe how a classification model has performed on a set of test data for which the true values are known. In our experiment, F1 score is 0.686.

From the results it is evident that our model performs way better than RAKE model.

## VI. CONCLUSION AND FUTURE WORK

In conclusion, we are successfully able to devise unsupervised algorithm that can aid in solving the problem of topic modeling with acceptable performance. Our model outperforms both the baseline and RAKE model, when tested on BBCNews dataset. Using the WordNet model as a pre- processing model, we could reduce the whole files into set or bag of phrases, this drastically improved the performance of our model. The unsupervised nature of our model saved a lot of training time which we would have to incur had we used a supervised approach. Our model, takes quite amount of time to give out results as it dynamically access Wikipedia corpus, each time the model is used. Future work can include an additional storage buffer that could store the whole corpus as a dump and access this, hence making the model faster. Also, new ways of word representation such as word-vectors can be included with our model that would drastically improve its performance..

## REFERENCES

1. *Beginners Guide to Topic Modeling in Python*:https://www.analyticsvidhya.com/blog/2016/08/begin ners-guide-to-topic-modeling-in-python
2. Christiane Fellbaum (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.
3. Ercan, Gonenc&Cicekli, Ilyas. (2007). Using lexical chains for key-word extraction. Information Processing & Management. 43. 1705-1714. 10.1016/j.ipm.2007.01.015.
4. M. Strube, S.P. Ponzetto, Wikirelate! Computing semantic relatedness us- ing Wikipedia, Proceedings of the 21st National Conference on Artificial Intelligence, AAAI06, vol. 2, AAAI Press, 2006, pp. 14191424.
5. E. Gabrilovich, S. Markovitch, Computing semantic relatedness using Wikipedia-based explicit semantic analysis, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence (San Francisco, CA, USA, 2007), IJCAI'07, Morgan Kaufmann Publishers Inc., 2007, pp. 16061611.
6. G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., New York, NY, USA, 1986.
7. S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, R.A. Harsh- man, Indexing by latent semantic analysis, Journal of the American Society of Information Science 41 (6) (1990) 391407.
8. HadjTaieb, M.A., Ben Aouicha, M., Ben Hamadou, A., 2013. Computing semantic relatedness using Wikipedia features. Knowl.-Based Syst. 50, 260278.
9. S. Rose, D. Engel, N. Cramer and W. Cowley, *Automatic keyword extraction from individual documents*, in Text Mining: Applications and Theory, West Sussex, Wiley, 2010, pp. 1-20.
10. Alice Leung, *Evaluating Automatic Keyword Extraction for Internet Reviews*, University Of Lorraine
11. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA Data Mining Software: An Update," SIGKDD Explorations, vol. 11, no. 1, 2009.
12. https://en.wikipedia.org/wiki/Machine learning
13. O. Medelyan, "Human-competitive automatic topic-indexing," 2009.

14. E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin and C. G. Nevill-Manning, "Domain-Specific Keyphrase Extraction," in Proceedings of the 16th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 1999.
15. H. Kopka and P. W. Daly, *A Guide to LATEX*, 3rd ed. Addison-Wesley, 1999.
16. API: Wiki-java master https://github.com/MER-C/wiki-java
17. *Scoring Based Unsupervised Approach to Classify Research Papers* Anil Kumar K.M, Gagan S G, Rajasimha N, Anil B, Rajath Kumar U, Department of CS & E, Sri Jayachamarajendra College of Engineering, Mysore-570006
18. Thomas K Landauer, *Latent Semantic Analysis* A text book, University of Colorado, Boulder, Colorado, USA
19. *Latent Dirichlet Allocation*, David M. Blei Computer Science Division University of California Berkeley, CA 94720, USA , Andrew Y. Ng Computer Science Department, Stanford University, Stanford, CA 94305, USA, Michael I. Jordan, Computer Science Division and Department of Statistics, University of California, Berkeley, CA 94720, USA
20. http://mlg.ucd.ie/datasets/bbc.html
21. https://nlp.stanford.edu/software/tagger.html
22. http://snowball.tartarus.org/algorithms/porter/stemmer.html


## AUTHORS PROFILE

**Anil Kumar K M** is working as an Associate Professor, in department of computer science and engineering. His area of Interest are Text mining, Sentiment Analysis, Data mining, Opinion mining, Web Mining, Data Analytics and Computer Network.

**Ajay B** is currently in final year BE in the department of computer science and engineering. His area of Interest are Deep Learning, Reinforcement Learning, Web Mining and Distributed Computing.

**Amogha Subramanya D A** is currently in final year BE in the department of computer science and engineering.His area of interest are Natural Language Processing, Machine Learning and Deep Learning.

**Shashank R** is currently in final year BE in the department of computer science and engineering. His area of interest are Natural Language Processing, Text Mining, Computer Vision and Deep Learning