Unsupervised Key-phrase Extraction using Noun Phrases

Shailendra Singh Kathait
Co-founder & Head
Artificial Intelligence &
Machine Learning Lab
Valiance Solutions,
Noida, Uttar Pradesh, 201301

Shubhrita Tiwari
Data Scientist
Artificial Intelligence &
Machine Learning Lab
Valiance Solutions,
Noida, Uttar Pradesh, 201301

Anubha Varshney
Data Scientist
Artificial Intelligence &
Machine Learning Lab
Valiance Solutions,
Noida, Uttar Pradesh, 201301

Ajit Sharma
Indian Institute of
Technology (IIT),Kharagpur
Kharagpur, 721302, India

ABSTRACT
The growing abundance of text articles in internet requires automated tagging using key phrases. The automated key phrase generation of resources helps in the information retrieval. To generate the key phrases for texts from all possible domains, the need is an automated approach that would extract the key ideas directly from the text itself. In this paper, we have suggested a methodology that uses the noun words and phrases, their occurrence and co-occurrences to generate the keywords. The method, employing both the statistical and linguistic features, has been successful in extracting the keywords and phrases to tag a text that best summarizes its content.

Keywords
Unsupervised Key phrase Extraction, Automated Tagging, Key phrase extraction using Nouns, Natural Language Processing, Information Retrieval.

1. INTRODUCTION
The key-phrase enables concise understanding of a text, enabling one to grasp the central idea, without the hassle of reading through the entire text document. At present times, where there exists a vast amount of information in the form of text on internet, the generation of keywords or phrase has assumed much wider application and importance. With the growing abundance of resource materials on the internet, the need of information retrieval calls for automatic tagging of a text or document to extract relevant information for a particular query of a user. Without any doubt, the task of manually tagging or summarizing such texts will be herculean; and this calls for automation in this field to reduce the time and effort and of course to meet the unprecedented volume of information to be exchanged today. For instance, text tagging is important to provide results for a query by a search engine as well as in the case of text classification, text summarization and comparison.

Any key-phrase extraction model aims at generating words and phrases that would together summarize the entire text. The algorithms for key-phrase extraction can be broadly classified into two types [1] [2] –

1. Supervised key-phrase extraction
2. Unsupervised key-phrase extraction.

Supervised method [1] employs choosing the best keywords from a prepared set of keywords, which is likely to contain topics from all genres and fields of interest. This method requires labeled documents with their tags or keywords. A model is developed to learn the ways in which the tags and keywords can be associated with a text and how they can be generated from a text.

While this can produce interpretable rules as to what characterizes a key-phrase, but the greatest challenge in this method is the availability of training data, tags and keywords for large number of texts whose topics encompass all genres of interest like scientific journals, news, business, education, entertainment, sports etc. Also, this method would not be able to incorporate the actual context of the text or in other words it would lack specificity.

The unsupervised methods [2] eliminate the need for training data. Unlike supervised methods, this uses the structure of the text itself and generates keywords and phrases from the text

itself using its properties. This approach holds undeniable importance as this can be applied across all languages and domains.

Another way, in which the key-phrase extraction methods can be classified, is extractive and abstractive methods of key-phrase extraction [3]. While in the extractive methods, the extraction is purely based from the text itself, the abstractive methods generate the keywords or summaries by the contextual understanding of the text incorporating the knowledge of Natural Language.

The Abstractive method has the capability to generate keywords similar to what a human might generate. But abstractive methods have still not been able to generate impressive results. It is still a challenge as it requires deep understanding of the natural language, to unlock the lexical, semantic and contextual meaning of the text.

Without doubt, the application of the Natural Language Processing to the key-phrase or contextual summarization

would be the best way to summarize a text, given the fact that it takes into account the grammar and language syntax to understand the text and the summary or key-phrase is generated accordingly.

2. RELATED WORKS

The Kea System [4] uses sequences of consecutive words, usually not more than three as the candidate keywords. It excludes proper names and phrases beginning or ending with stop words. It then uses two features to select key-phrases, TF-IDF (term Frequency – Inverse Document Frequency) and the distance of the word from the beginning of the document. The features are then weighted by Naive Bayes techniques. Since the proper names and phrases are excluded, accuracy of extraction is reduced.

The Extractor [5] by Turney extracts relevant key-phrases from a list of candidate key-phrases which consists of all sequences of a small number of words, up to five with no stop words or punctuations in between. Selection of the key-phrases is based on scoring the candidate key-phrases on a number of features such as frequency, of the stemmed word in the phrase, the length of the phrase, position of the phrase in the document. The entire content of the text is not covered in this case.

Krulwich & Burkey [6] extracted "semantically significant phrases" from the structural features of the text. Phrases are chosen based upon various heuristics with the purpose of extracting phrases to determine a user's interest. The proper nouns are excluded and the entire information describing the text cannot be extracted using this method.

The unsupervised learning method used by us takes into account all the possible text data that best describes the content and sentiments of the text.

3. KEY-PHRASE EXTRACTION USING NOUN PHRASES

Document key-phrases have enabled fast and accurate searching for a given document from a large text collection. They exhibit potential in improving many natural language processing (NLP) and information retrieval (IR) tasks, such as text summarization, text categorization, opinion mining, and document indexing [7].

Algorithms for unsupervised way of key-phrase extraction basically involve 2 steps:

1. Candidate words or lexical units are extracted from the textual content of the target document by applying stop-word and parts-of-speech filters. Only noun and adjectives that are likely to be key-phrases are retained in this step.

2. Next, candidate words are scored based on some criterion. For example, in the TF-IDF scoring scheme, a candidate word score is the product of its frequency in the document and its inverse document frequency in the collection.

3. Finally, consecutive words, phrases or n-grams are scored by using the sum of scores of individual words that comprise the phrase. The top-scoring phrases are output as predictions (the key-phrases for the document) [8].

Any text, belonging to any domain of interest, follows the basic structure of grammar and language. Hence, any text is a collection or string of words where each word belongs to a particular part of speech and their structural position obeys the basic principles of grammar of that particular language. Obviously, all words are not equally important to convey the central idea of the text. Some words are present just to make the sentence grammatically, linguistically valid.

Our method is based on the premise that most of the meaning and concept of the text is conveyed by the noun words and noun phrases present in the text [9]. For example, in the following text, the words in bold conveys much of the information regarding the central idea:

*The **Obama administration** is dismantling a **dormant national registry program** for visitors from countries with **active terrorist groups** — a **program** that **President-elect Donald Trump** has suggested he is considering resurrecting. The **registry**, created after the **attacks** of September 11, 2001, has not been in use since 2011, so the **move** is largely symbolic and appeared to be aimed at distancing the **departing administration** from any effort by the **new president** to revive the **program**, known as the **National Security Entry-Exit Registration System**, or NSEERS.*

The other important aspect of sentence structure employed in the approach to extract key-phrases/words is that two candidate words, (here noun and adjective words and noun phrases) which occur together in a sentence give meaning and weight to each other [10]. This can be seen in the above paragraph that the noun "Obama" and "administration" gives meaning to each other. Similar case is with {('dormant', 'Adjective'), ('national', 'Adjective'), ('registry', 'Noun'), ('program', 'Noun')}.

Using these two principles, we designed an algorithm which involves extraction of the candidate keywords or phrases, finding the interconnection of those words with other words occurring in the sentence and finally devising a scoring method to rank all the candidate key-phrases/words. The best candidates based on the score are selected as the keyword for the text. Thus, this method employs both linguistic features and the statistical features to generate the keywords or phrases for a text. After the generation of the candidate keywords using the facets of natural language, the statistical metrics like how many times the word/phrase has occurred, and number of times it has co-occurred with other words help us score the candidates.

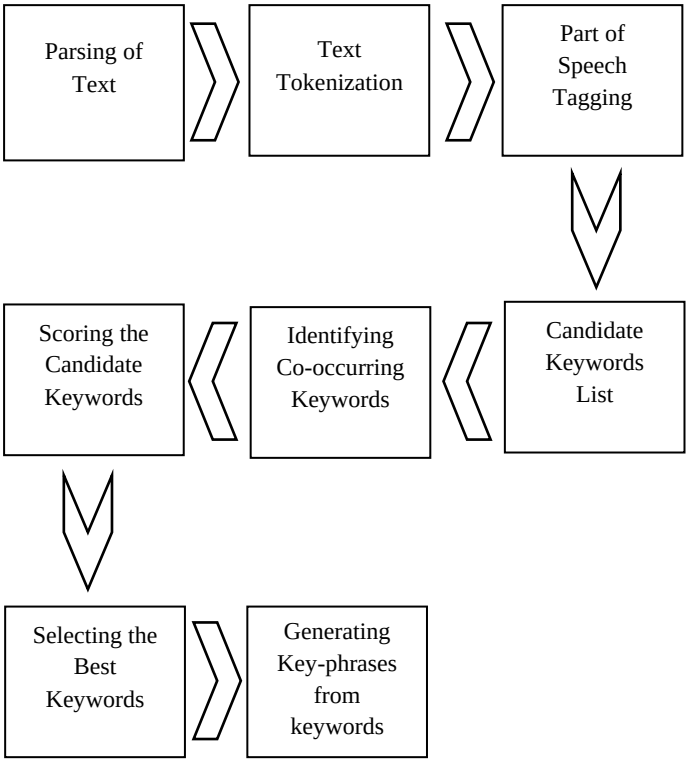

Figure (i): Schematic Representation of the Proposed Approach

4. METHODOLOGY

Most of the text available on the internet is simply a string of characters. Such texts are useless to apply the tools of Natural Language on. Hence, the primary step involves cleaning the input text so that processing can be done in later phases. To achieve this, we have used the Natural Language Toolkit [11] which is a popular platform for building python programs to work with human language data. The 'nltk' provides most of the tools that is required for text cleaning and processing.

- **Parsing of text:** The first step involves parsing of the text. This involves identifying words and sentences in the text, which is identified by the spaces, punctuations and the other non-alphanumeric characters. At first step, the entire text is split into sentences by noting the location of the punctuation marks viz. ".", "?", "!"
- **Tokenization:** Text, whole of the text or each of the sentence is converted into tokenized words. This converts each sentence into a list of words. It is to be noted that punctuations like ",","," etc. form individual tokens. Converting the text into a list of tokens is important because this helps applying the tools of natural language to the text.
- **Part of Speech Tagging:** After the text, has been parsed and tokenized, the part of speech of each tokenized word is identified using the tools of Natural Language Toolkit [10]. This identifies the words as nouns, adjectives, verbs, determiners etc. One important aspect of the Part of Speech Tagger of the Natural Language Toolkit [9] is that it not only identifies word as noun or adjectives but also identifies whether it is a noun, common, singular or mass as NN, noun, proper, singular as NNP and noun, common, plural as NNS. Hence, based upon the context, the POS tagger of nltk identifies the part of speech of the words correctly.
- **Listing candidate keywords:** Next, a set of all the candidate keywords are created. For this, visit all the words in the text and then choose the ones that are noun or adjectives preceding the noun. While doing this if the selected word is found to be a very common word in the language, then this is not taken into account. Thus, the set finally consists of the candidate keywords that have the potential of becoming a keyword defining the text in a way or the other.
- **Identifying all the co-occurring words:** After a set of candidate keyword is created, the sour task is to identify the interconnection of that word with the other words in the text. For this, a window is selected on N words, usually N ranges between 2 and 10. For a particular candidate word, if another candidate word is found within a window of N words, then the two words are defined to co-occur and thus interconnected. This is repeated for all the potential candidate keywords, which gives all pairs of co-occurring words. The diagram below illustrates the co-occurrence relation between the candidate keywords when N=4.

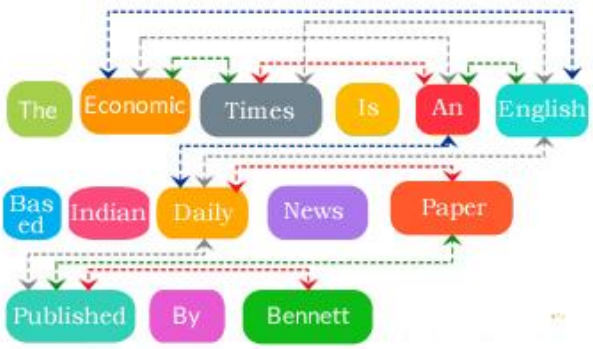

Figure (ii): **Co-occurrence relation between keywords**

- **Scoring of the potential candidate keywords:** The scoring criteria for keywords extraction is very crucial as it should be able to surface out the most of the necessary keywords. The scoring of the keywords depends upon the pattern of occurrence of the word in the text. In our approach, we used a blend of frequency of the occurrence of the word and also frequency with which a word has co-occurred with other words.
- Let the score of a word, w_i from its frequency of occurrence be $f_{1,i}$ and that from the co-occurrence of the word be $f_{2,i}$. Thus, the final score for a candidate word W_i will be:

$$W_i = \alpha f_{1,i} + \beta f_{2,i}$$

$f_{1,i}$ is the number of times the word has occurred in the text. But we propose the scoring of $f_{2,i}$ in the following manner: The co-occurrence distance between w_i and w_j be $d_{i,j}$, the function $g(d_{i,j})$ returns the interconnection/co-occurrence score depending upon the distance, S_i is the set of all candidate words connected to word, w_i.

Hence,

$$f_{2,i} = \sum_{j \in S_i} g(d_{i,j})$$

Now, we define $g(d_{i,j})$. The co-occurrence score should be such that it is higher when two words are very closer and lower when they are far apart. For instance, when N = 4(N being the maximum window of occurrence). The final score of each candidate keyword now involves defining 2 more terms: α and β. We see that the keyword 'will' is highly connected, that is with high co-occurrence score, even though it has low frequency, it should turn out as an important keyword. Hence, the final score should give higher weight-age to the co-occurrence score than that of frequency score so as to lower the undesirable effect of some words with very high frequency. In our implementation, we found the best results for $\alpha = 0.2$ and $\beta = 2$.

From Keywords to Key-phrases: With the final scoring, one can choose *n* number of keywords that best describe the text. It is observed that the keywords generated now are all unigrams, but obviously, phrases convey central idea better the unigrams. These generated keywords can be used to extract phrases from the text. To extract the phrases, firstly all the possible phrases are extracted from the text. One can see that, most phrases especially noun phrases, the one of our interest starts after "a", "an", "the" and ",". Similarly, most of these would end with",", ";" "to", "has", "from", "with", "and", "that". Hence the words sequence between this start and end list will form a phrase. All such phrases are extracted from the entire document. Next, phrases which have at least two words among the top-words list are chosen as the key-phrases defining the central idea of the text.

5. IMPLEMENTATION

The above proposed method was implemented in python 2.7 and used the Natural Language Toolkit 3.0 and Word-net thesaurus to process the text. The program was run on a variety of text from different domains example the scientific journals, research papers, online news, literature books and more. The method proposed was successfully able to list the keywords that would be able to classify a text. The efficacy of the model was evaluated by generating the keywords on the journal articles and papers which already have the keywords penned by the author. Four out of five keywords generated from the abstract were found to be similar to that given by the author. Hence, this method with such success can be applied well to tag papers and text documents automatically across all domains.

6. OBSERVATIONS

The proposed algorithm was tested for accuracy on 10000 different types of texts content of varying categories from which different key-phrases were extracted and checked for accuracy that is whether it is describing content of the texts completely or not. The results obtained were better than those obtained using statistical methods.

Table-1 Observation Table

Type of Text	Number of relevant key-phrases	Number of irrelevant key-phrases	Extraction Accuracy
Political	3	1	0.75
Sports	4	2	0.50
Entertainment	5	1	0.83
Education	3	0	1.0
Economy	2	0	1.0

The algorithm was tested on 10000 different texts of different categories, the key-phrases extracted were manually checked for relevance and the accuracy was tabulated in table. The overall accuracy was obtained as 80%, much better as compared to that obtained through statistical methods.

7. CONCLUSION

The algorithm provided promising results, as can be seen from the observation table (Table-1) but still there lies a scope of improvement, especially in some categories of text (like Sports) for which poor results were obtained.

For these types of texts, supervised learning approach can be used in-order to get accurate results. A novel features based approach based on citation network information used in conjunction with traditional features for key-phrase extraction is implemented that gives remarkable improvements in performance over strong baselines [12].

8. LIMITATIONS AND FUTURE WORK

The aspect that limits the efficiency of the model is the way in which the co-occurrence and interconnection has been defined. The above method assumes two candidate keywords located within a distance of N $\in$ [2,10] as co-occurring and increasing weights of each other. But a better method would be to understand the actual dependency of words using principles of grammar and natural language. This would surely improve the quality of key-phrase extraction. The further work can be extended to include the lexical and semantic structure of the natural language at a deeper level that would enable producing the key-phrase identical to what an educated man may produce.

9. REFERENCES

[1] Ana Mestrovic, Beliga, Sanda Martincic-Ipsic, Slobodan, "An overview of graph-based keyword extraction methods and approaches", Journal of Information and Organizational Sciences 39, (2015).

[2] Aditi Sharan, Siddiqi, Sifatullah, "Keyword and key-phrase extraction techniques: A literature review", International Journal of Computer Applications" 109, (2015).

[3] Alexander Gelbukh, Erendira Rendon, Garcia-Hernandez, Rafael Cruz, Rene Arnulfo, Romyna Montiel, Yulia Ledeneva, "Text Summarization by Sentence Extraction Using Unsupervised Learning" , Mexican International Conference on Artificial Intelligence, Springer.

[4] Carl Gutwin, Craig G. Nevill-Manning, Eibe Frank, Gordon W. Paynter , Ian H., Witten, "KEA: Practical automatic key-phrase extraction" , Proceedings of the fourth ACM conference on Digital libraries.

[5] Peter D, Turney, "Learning algorithms for key-phrase Extraction", Information retrieval 2, no. 4 (2000).

[6] Bruce, Chad Burkey, Krulwich, "Learning user information interests through extraction of semantically significant phrases", Proceedings of the AAAI spring symposium on machine learning in information access.

[7] Kazi Saidul Hasan, Vincent Ng, "Automatic Keyphrase Extraction: A Survey of the State of the Art", University of Texas at Dallas Richardson.

[8] Cornelia Caragea, Sujatha Das Gollapalli, "Extracting Keyphrases from Research Papers Using Citation Networks", University of North Texas

[9] Abed Alhakim, Biswanath Dutta, Fausto Giunchiglia , Freihat, "Compound Noun Polysemy and Sense Enumeration in WordNet".

[10] George A, Miller, "Word-Net: a lexical database for English", Communications of the ACM 38, no. 11 (1995).

[11] Bird, Steven, Edward Loper, Ewan Klein, "Natural language processing with Python", O'Reilly Media, Inc.", 2009.

[12] Andreea Godea, Cornelia Caragea, Florin Bulgarov, Sujatha Das Gollapalli, "Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach", Singapore.