



International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 10 • Number 32 • 2017

Document Summarization and Keyword Generation using Graph Based Ranking Algorithm

M. Senthilraja^a R. Srinivasan^a and S. Iniyana^a

^aComputer Science and Engineering, SRM University, Tamilnadu, India

E-mail: snthl.rj@gmail.com, srinirvs89@gmail.com, iniyana.sv@gmail.com

Abstract: This research work presents a pioneering method for automatic sentence extraction using graph-based ranking algorithms. In the context of text summarization, the task introduces a domain-independent technique that results were favourable comparing to standard benchmark results.

Methods: This technique will automatically do the keyword summarization and sentence extraction by comparing the generated results with the human generated one. Because, the extracts generated by two humans from the same article are surprisingly dissimilar. Even though this observation questions the feasibility of generating perfect summaries by extraction, given that the other elective domain-independent summarization tools are unavailable. It can be established that this is a reasonable, though imperfect, and alternative.

Application: Therefore, the document summarization is an interesting and constructive task which gives support to many other duties. In addition, the rouge tool presents in this research work may be helpful to have an idea of document summarization.

Keywords: Page rank, Text rank.

1. INTRODUCTION

Document summarization is a sub-domain of textual analysis which itself is a technique of accumulating information on how people study and interpret the world. It is a methodological and data-gathering process, for the researchers willing to comprehend the customs of how members of diverse cultures and sub-cultures make logics of their individuality, and of how they exist in the world in that formation. Domains in which textual analysis is useful include musical studies, media studies, sociology, psychology, and perhaps mass communication.

The types of texts by which interpret can include magazines, advertisements, clothes, graffiti, news, articles and so on. It generates an elucidation of things like a book, magazine, and kilt that can treat it as a text. Summarization plays an important role in text analysis. The question that boosts this analysis is how the competence of information utilisation and sharing can be facilitated using automatic summarization, chiefly in a workplace. For that, the results obtained by using these techniques are entailed called summaries. 'Summary' is a term used for the selective output of summarization.

Summarization seeps into our lives through many sources; article contents, full news, weather forecasting, data tables, critic reviews, scientific articles, research papers, abstracts, scripts, and other structured and unstructured data forms are some of the forms that summaries can take.

Automatic text summarization is a class of computational logic rules with an ultimate functionality to take individual text documents, run the automatic tool over it, extract their substance and present it to the end-user in a precise manner. The use of computers exponentially hikes the potential of producing summaries. It not only allows users to browse promptly through a bulk of data and content but also save them the overhead of producing summaries manually that in return saves valuable resource and time. Albeit, using the automatic summarization technique to generate summaries of high quality, at par with the human-generated summaries, is a complicated process due to the inability of computers to interpret natural language like a human does.

The most pragmatic approach that the researchers have found in an attempt to resolve this issue is the extractive text summarization technique. This involves extraction of the most significant content of a document (like nouns, numbers, figures and adjectives) that holds the gist of a report. Keyword extraction is implemented by automatically identifying those terms that best describe the idea of the paper.

Graph-based models, in this report, refer to the collection of machine learning models that extrapolate a primary graph structure. It is important to understand that the graphical models which often appear in Bayesian analysis literature are specific however the graph based models that are conferred here are generic. Graphical models are based on probability factor which inference structures in the form of graphs. In graphical models, nodes, in general, are used to represent variables while the edges are used to represent conditional dependency of the connecting variables. However, probabilistic and non-probabilistic models cannot be directly compared to limitations or advantages.

All these concepts are needed to be enlightened upon and inculcated to form a union for generating an application based on text analysis. Over the course of iteration through this report, all these concepts will merge and produce the desirable output. The expected output of this application is a set of words or phrases which will represent a given natural language text. The components to be ranked are thus sequences of one or more lexical units extracted from given text. These^[9] represent the vertices that are added to the text graph and any relation that can be identified between two such lexical units is likely to be a useful connection that can be added to them. It uses a co-occurrence relation controlled by the distance between word occurrences. Two vertices are connected if their matching lexical units co-occur within a window of maximum words, where a limit can be set anywhere from 2 to 10 words. Co-occurrence^[4] links explain relations between syntactic elements and are similar to the semantic links found productive for the task of word sense disambiguation. They depict cohesion indicators for a given text. The vertices added to the graph can be controlled with syntactic filters, which select only lexical unit of a particular part of speech. For instance, one can consider only nouns and adjectives for addition to the graph, and henceforth draw potential edges based merely on relations that can be established between nouns and adjectives. The Text Rank keyword extraction algorithm is completely unsupervised and proceeds as follows.

First, the text is tokenized and annotated with part of speech tags. This is a pre-processing step required to enable the application of syntactic filters. In this paper mentioned every word as persons for summation to the set of graph vertices, with multi-word keywords being ultimately reconstructed in the post-processing phase. This is to avoid excessive growth of the size of graph by adding all probable combinations of series consisting of more than one lexical unit (n-grams). Then, all lexical units that overtake the syntactic filter are added to the graph and an edge is added to link the lexical units that co-occur within a window of words. After the undirected unweight graph is constructed, the score associated with each vertex is set to 1, taken as initial value, and the ranking algorithm is run on the graph for several iterations until the graph finally converges⁷: say for 20-30 iterations, at a threshold value of 0.0001. Once a final score is found for each vertex in the graph, vertices are sorted in the inverse order of their score, and the top few vertices in that ranking are retained for post-

processing. While it may be set to any fixed value, it usually ranges from 5 to 20 keywords. In post-processing part, all units selected as potential keywords by the Text Rank algorithm are marked and sequences of adjacent potential keywords are collapsed into a multi-word keyword. For instance, in the text “number system” for plotting ambiguity functions, if both “number” and “system” are selected as potential keywords by Text Rank, since they are adjacent, they are collapsed into one single keyword “number system”.

Text Rank is applied over a problem by first building a graph related with the text, where the vertices of graph represent the units to be ranked. In the scenario of sentence extraction, the aim is to rank entire sentences. For that reason a vertex is added to the graph for each sentence in the text ⁷. The co-occurrence relation which is used otherwise for keyword extraction cannot be implemented here as the text units under scrutiny are significantly larger than one or few words, and “co-occurrence” is just not a meaningful relation for such big contextual units. This “similarity” is measured as a function of their content overlap and such a relation between two sentences can be seen as a process of “recommendation”. Moreover, to avoid promoting long sentences, normalization factor can be used, and divide the content overlap of two sentences with the length of each sentence.

Other sentence similarity procedures, say, cosine similarity, string kernels, longest common subsequence are also possible, and then currently evaluating their impact on the performance of summarization. The resulting graph is densely connected and a weight associated with each edge, indicating the strength of those edges as connections established between various sentences in the text. The text is hence represented as a weighted graph, and accordingly by using the weighted graph-based ranking formula explained below. After the ranking algorithm is executed over the graph, sentences are sorted in a reverse order of their score. Then, the top ranked sentences are selected for inclusion in the summary ⁷.

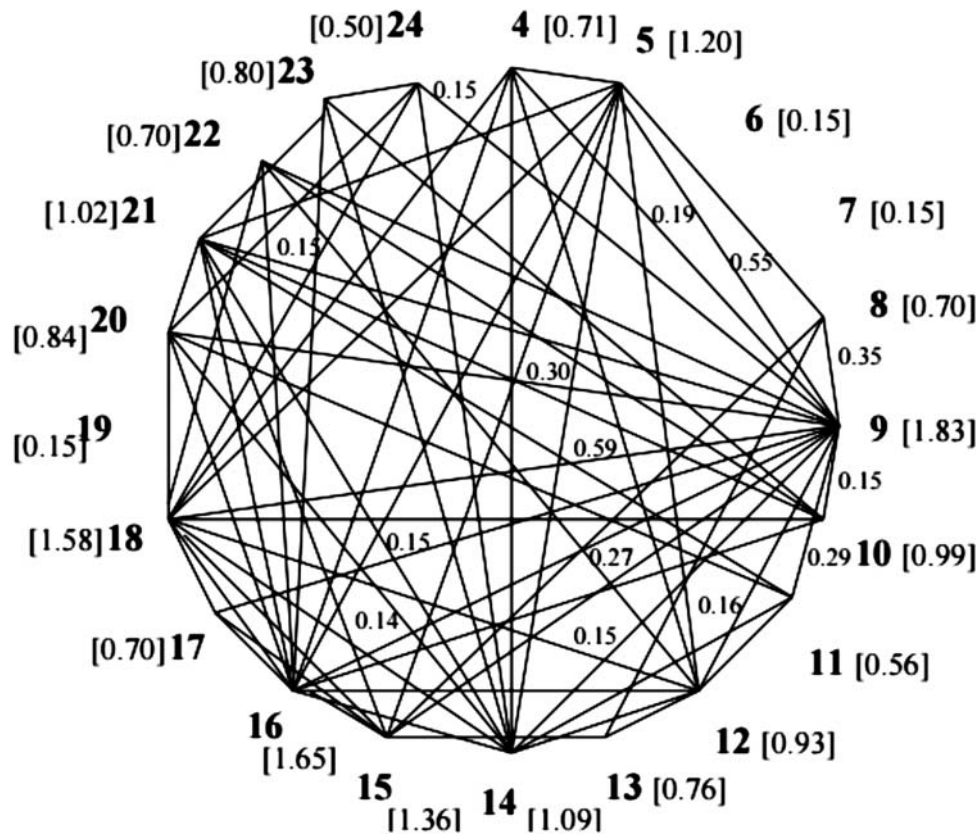


Figure 1: Graph showing text rank of sentences

2. PROPOSED WORK

Modules are the independent units that have a standardized functionality and simple implementation. These modules can be inculcated for making more complex structures or tools.

2.1. Tag extract

It applies filters based on POS (Part-of- speech) tagging. The tagging in the tool usually takes into account nouns, adjectives and proper nouns by tag name NN, JJ and NNP respectively.

2.1.1. Part of Speech (POS) Tagging

A POS Tagger is a software by which it reads text and identify each word such as noun, verb, adjective, etc.,

Table 1
List of part-of-speech tags

<i>No.</i>	<i>Tag</i>	<i>Description</i>
1.	CC	Coordinating Description
2.	CD	Cardinal Number
3.	DT	Determiner
4.	EX	Existential There
5.	FW	Foreign Word
6.	JJ	Preposition or subordinating conjunction
7.	JJR	Adjective , comparative
8.	JJS	Adjective , superlative
9.	LS	List item marker
10.	MD	Modal
11.	NN	Noun , singular or mass
12.	NNP	Proper Noun, singular
13.	RB	Adverb
14.	VB	Verb
15.	VBD	Verb, Past Tense
16.	IN	Preposition or subordinating conjunction
17.	SYM	Symbol
18.	RP	Particle
19.	UH	Interjection
20.	RBS	Adverb, superlative

2.1.2. Normalize

It normalizes the text and keywords and removes the special characters including full-stop, comma from the words.

2.1.3. Unique_everseen

It takes a stream of all keywords tokenized from the given document and removes the redundancy. It keeps single instance of every keyword and returns the unique set of keywords as output.

2.1.4 L-distance (Levenshtein distance)

L-distance is calculated between the vertices to obtain a weight for every edge drawn between any pair of vertices. L-distance is also used directly to adjust sequences, and the demonstration proves how this works efficiently. The minimum numbers of editing operations are Insertion, Deletion and Substitution.

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} leva,b(i-1,j) + 1 \\ leva,b(i,j-1) + 1 \\ leva,b(i-1,j-1) + 1(ai \neq bj) \end{cases} & \text{otherwise} \end{cases}$$

Upper and Lower limits are

1. Difference between the sizes of the two strings
2. The value is zero, only if the strings are equal

The L-distance between two strings of length n can be approximately – $(\log n)^{O(1/\epsilon)}$ where $\epsilon > 0$ is a free parameter, in time $O(n^{1+\epsilon})$

The time complexity of the program would be $O(mn)$ and a more optimization will produce $O(\min(m, n))$ where m is first string and n is the second string.

Example, the Levenhstien distance between “rank” and “graph” is 3.

graph → rank (substitution of “p” and “n”)

graph → rapk (substitution of “h” and “k”)

graph → graph (insertion of “g” at the first place)

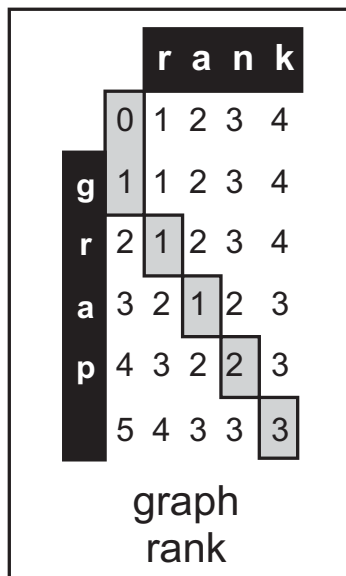


Figure 2: Adjacency Matrix for Levenshtein distance calculation

2.1.5. Build graph

Initialization and growth of graph takes place in this module. It uses pythonnetworkx library to create text and sentence based graphs and store them. Creation of graph and nodes and their combinations are made using networkx library.

2.1.6. Keyword Extraction

It uses the nltk package to tokenize and normalize the text to segregate all the keywords. Moreover, it identifies the adjacent keywords then integrates them to generate another list of key phrases.

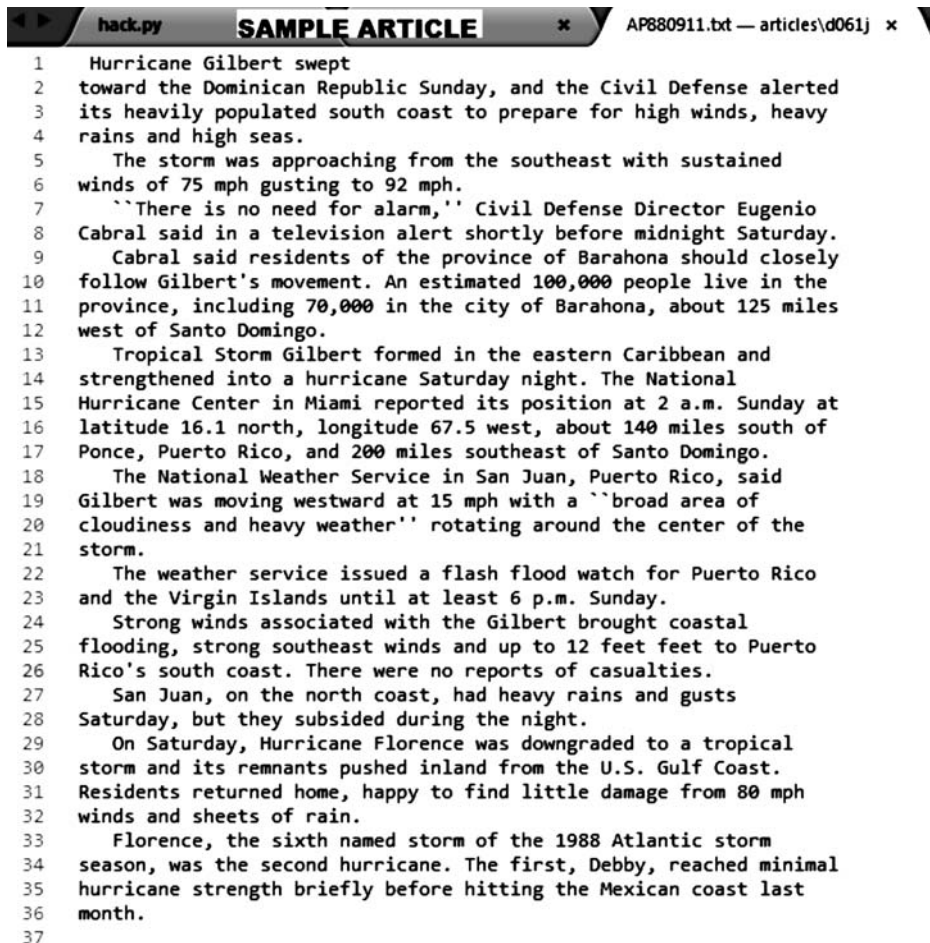
2.1.7. Sentence Extraction

It generates a set of separated sentences by breaking down the paragraphs at stop points.

2.1.8. Write files

Finally, the results are expected to be stored in some directory using OS library. The summary, set of keywords and extracted sentences, and graph are stored in separate files with same title as that of the document being analysed.

3. EXPERIMENTAL RESULTS



```
hack.py SAMPLE ARTICLE * AP880911.txt — articles\d061j x
1 Hurricane Gilbert swept
2 toward the Dominican Republic Sunday, and the Civil Defense alerted
3 its heavily populated south coast to prepare for high winds, heavy
4 rains and high seas.
5 The storm was approaching from the southeast with sustained
6 winds of 75 mph gusting to 92 mph.
7 ``There is no need for alarm,'' Civil Defense Director Eugenio
8 Cabral said in a television alert shortly before midnight Saturday.
9 Cabral said residents of the province of Barahona should closely
10 follow Gilbert's movement. An estimated 100,000 people live in the
11 province, including 70,000 in the city of Barahona, about 125 miles
12 west of Santo Domingo.
13 Tropical Storm Gilbert formed in the eastern Caribbean and
14 strengthened into a hurricane Saturday night. The National
15 Hurricane Center in Miami reported its position at 2 a.m. Sunday at
16 latitude 16.1 north, longitude 67.5 west, about 140 miles south of
17 Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
18 The National Weather Service in San Juan, Puerto Rico, said
19 Gilbert was moving westward at 15 mph with a ``broad area of
20 cloudiness and heavy weather'' rotating around the center of the
21 storm.
22 The weather service issued a flash flood watch for Puerto Rico
23 and the Virgin Islands until at least 6 p.m. Sunday.
24 Strong winds associated with the Gilbert brought coastal
25 flooding, strong southeast winds and up to 12 feet of rain to Puerto
26 Rico's south coast. There were no reports of casualties.
27 San Juan, on the north coast, had heavy rains and gusts
28 Saturday, but they subsided during the night.
29 On Saturday, Hurricane Florence was downgraded to a tropical
30 storm and its remnants pushed inland from the U.S. Gulf Coast.
31 Residents returned home, happy to find little damage from 80 mph
32 winds and sheets of rain.
33 Florence, the sixth named storm of the 1988 Atlantic storm
34 season, was the second hurricane. The first, Debby, reached minimal
35 hurricane strength briefly before hitting the Mexican coast last
36 month.
37
```

Figure 3: Sample Article for summarization

The Graph based ranking algorithm is evaluated in the context of a document summarization task.

No of news article is taken as an input: 67

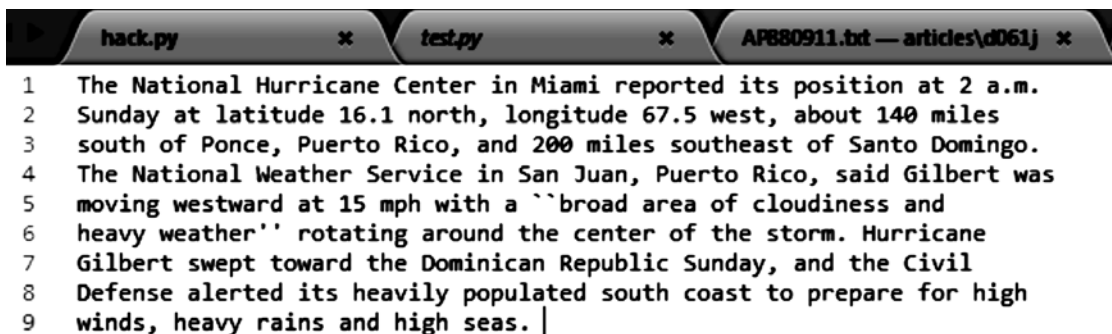
No of words generated for summary: 100(per article)

The ROUGE evaluation toolkit can be used for evaluating Ngram statistics, found to be highly connected with human evaluations (Lin and Hovy, 2003a).



```
hack.py x test.py x
1 Caribbean
2 Director Eugenio
3 Hurricane Florence
4 Tropical
5 Islands
6 estimated
7 National Hurricane
8 tropical
9 midnight Saturday
10 Atlantic
11 latitude
12 Dominican Republic
13 movement
14 province
15 strength briefly
16 Barahona
17 southeast
18 hurricane Saturday
19 sustained
20 hurricane strength
21 television
22 Defense Director
23 Hurricane Gilbert
24 Defense
25 position
26 cloudiness
27 flooding
28
```

Figure 4: Keywords generated



```
hack.py x test.py x AF880911.txt - articles\d061j x
1 The National Hurricane Center in Miami reported its position at 2 a.m.
2 Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles
3 south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
4 The National Weather Service in San Juan, Puerto Rico, said Gilbert was
5 moving westward at 15 mph with a ``broad area of cloudiness and
6 heavy weather'' rotating around the center of the storm. Hurricane
7 Gilbert swept toward the Dominican Republic Sunday, and the Civil
8 Defense alerted its heavily populated south coast to prepare for high
9 winds, heavy rains and high seas. |
```

Figure 5: Summary Generated

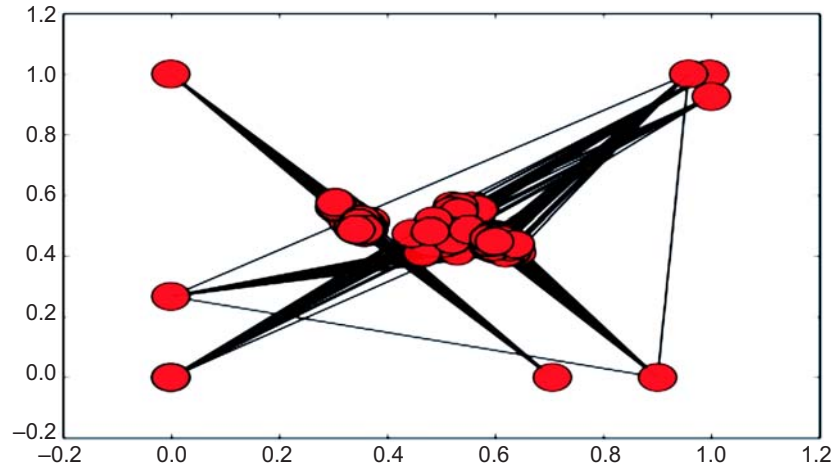


Figure 6: Graph generated of sentences

It evaluate the summaries through graphs and it builds on texts, this tool identifies connections between various entities of a text.

4. DISCUSSION

The Graph-based Rank approach to sentence extraction succeeds on establishing the most important sentences in a text based on information exclusively drawn from the text itself. Unlike other supervised systems, which attempt to understand what makes a good summary by training on collections of summaries built for other articles. Text Rank is fully unsupervised, and relies only on the given text to derive an extractive summary.

Amidst all algorithms, the HITS_A and PageRank algorithms provide the best performance, at par with the best performing system from DUC 2002. Notice that Text Rank goes beyond the sentence “connectivity” in a text. Another important advantage of Text Rank is that it gives a ranking over all sentences in a text which means that it can be easily acquire to extracting very short summaries, or longer more explicative summaries, consisting of more than 100 words.

PageRank measures a ranking of the nodes in the graph G based on the structure of the incoming links. It was initially designed as an algorithm to rank web pages. A directed graph can also be used because the PageRank function will convert it into undirected by taking edges and weights. Then, Sorting them in reverse order of rank to get most important sentences and return top 10 sentences or a 100 word summary. Files are read from different directories and keywords and summaries are generated in a new separate directory with the same file name as the input file name. Finally, write output in two separate files containing keyword and summaries for each article.

5. CONCLUSION

Most of the current research is based on extractive single-document summarization and multi-document summarization. Current summarization systems are widely used to summarize news, documents, email threads and other online articles. The scheme and tool works well because it does not merely depend on the local context of a text unit, but rather it takes into account information recursively drawn from the entire text which is iterated over and over till convergence is achieved. Through the graphs and summaries it builds on texts, this tool identifies connections between various entities of a text then implements the concept of voting and recommendation. A unit of text recommends other related text units, and overall strength/importance of the recommendation is recursively computed based on the recommendation of the units making the recommendation. In the process of identifying key sentences in a text, a sentence can recommend another sentence that underlines

parallel concepts as being useful for the overall deciphering of the text. Highly recommended sentences are likely to be more informative and knowledgeable for the given text. Therefore, they will be given a higher score. An important aspect of Text Rank algorithm is utilized in this tool. It is that this algorithm does not require deep linguistic knowledge or domain or language specific annotated corpora. This makes it highly portable to other domains, languages or genres. By a large, the high density input which is imported as input to this tool is ran over by an analyser that breaks down the content and produces the optimistic results saving time, effort and resources. Resultant of this is a concise set of information which is renowned for its application in multiple domains.

REFERENCES

- [1] Bird, Steven, Edward Loper, Ewan Klein, "Natural Language Processing with Python" O'Reilly Media Inc.
- [2] "<http://duc.nist.gov>", "<http://www-lpir.nist.gov/papers/duc/>", DUC.2002.The Document Understanding Conference
- [3] Frank E., Paynter G.W., Witten I.H., Gutwin C., Nevill- Manning C.G., "Domain-specific Key phrase Extraction", Proc.16th International Joint Conference on Artificial Intelligence, 1999.
- [4] J. Hobbs., "A model for natural language semantics Part I: The model Technical report", Yale University, 1974.
- [5] C.Y Lin., E.Hovy, "The automated acquisition of topic signatures for text summarization", In Proc. of the 18th conference on Computational linguistics - Volume 1, 2000.
- [6] Mihalcea R., Tarau P., "Text Rank: Bringing Order into Text", In Proc. OfConference on Empirical Methods on Natural Language Processing, 2004.
- [7] R.Mihalcea, "Graph-based ranking algorithms for sentence extraction, applied to text summarization", In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 2004.
- [8] ROUGE Toolkit, <http://www.berouge.com/>
- [9] S.Brin., L.Page, "The anatomy of a large-scale hyper textual Web search engine", Computer Networks and ISDN Systems, 1998.
- [10] http://en.wikipedia.org/wiki/Automatic_summarization