NATURAL LANGUAGE PROCESSING AND SUMMARIZATION

Aseem Upadhyay
Grad number: 7
CS5760: Topic Assignment 2
Michigan Technological University

ABSTRACT

In this paper, Natural Language Processing is discussed along with its applications, particularly text summarization. Summarization can be done using supervised as well as unsupervised methods. Using one of the unsupervised methods, the TextRank[1] algorithm, which is a graph-based ranking model for text processing, summarization can be obtained. In the graph, the vertices represent the sentences and the edge weights determine the importance of the sentence in the text. A new algorithm consisting of concepts of an intersection function and a sentences dictionary is proposed which when combined with the TextRank algorithm may provide a better and more efficient summary for the text document.

INTRODUCTION

Natural Language Processing

The primary focus of NLP is the interaction between the human beings and the computer systems. It is a medium for computers and humans to analyze and understand what the other means. NLP can be easily described as an intersection between computer science, artificial intelligence and computational linguistics. The history of machine translation [5][7] (which served as a base for NLP) dates to the early 20th century. In 1950, Alan Turing came up with, what is now called as "Turing Test", which could determine the difference between a computer and a human being based on conversational content. Noam Chomsky also revolutionized linguistics by coming up with a universal grammar [8][9]. The Georgetown experiment in the 1950s, was responsible for the translation of more than sixty Russian sentences into English. After the advancements in machine learning algorithms during the 1980s, there was a significant increase in the computational power of a machine. Decision tree models and statistical models were used to attach real weights to the input data which makes the text processing easier. Recent research is focused on the learning algorithms i.e. training the machine to learn the language and then respond accordingly. These learning algorithms could be either unsupervised (like clustering, k-means etc.) or supervised (decision trees, neural networks etc.).

The two major components of NLP are natural language understanding(NLU) and natural language generation(NLG). Natural language understanding comprises of multiple tasks such as mapping the natural language input to another representation or analyzing the different aspects of the language such as part-of-speech. NLG is relatively simpler than the understanding part. It can be described as a process to produce meaningful sentences or phrases in the natural language using some internal representations. Text planning, which means retrieving the information from the base is one of the initial steps in NLG. Sentence planning, which includes the choosing of optimal and meaning words to create a specific tone for the sentence, is done after text planning.

In the end, Text Realization is done, through which the entire sentence structure is created based on the previous two steps.

There are several difficulties in NLU. Ambiguity is the primary cause of it. There can be three basic levels of ambiguity. Lexical ambiguity refers to the ambiguous meaning of a certain word because of the multiple ways in which it is used. For instance, if a word “fly” is present in the text, the ambiguity arises as to if it is a noun or a verb. Syntax level ambiguity is a level above lexical, and refers to the ambiguity that arises when a sentence is parsed. For instance if there is a sentence, “He killed the bird with hands”, it is clear to the human beings to understand the meaning of this sentence. However, a machine may interpret this sentence as him killing the bird which had hands. This is syntax level ambiguity and is harder to deal with than the lexical ambiguity. The third level of ambiguity can be the referential ambiguity which usually arises due to the use of pronouns. For instance, “Alex told Clark that he had lost”, gives possibilities of the “he” pronoun referring to either Alex or Clark. For a machine, it is very hard to resolve such ambiguities as well.

Steps in NLP

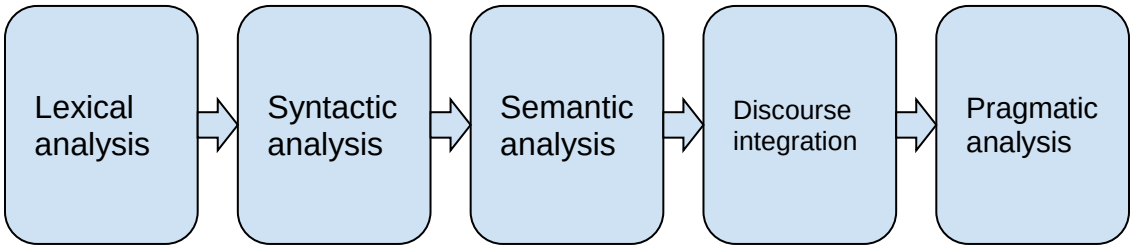


Fig1.: Steps in NLP

The first step in natural language processing, lexical analysis, refers to the identification and analysis of the structure of words. Through this, the entire input text is divided into paragraphs, sentences and then words. Syntactic Analysis is also referred to as parsing and it involves the analysis of words in the sentence for grammar. The sentences are broken down into their component parts of speech and arranged to show the relationship among the words in the sentence. Semantic analysis refers to the extraction of the meaning of the entire text. It can also be described as the process of relating all the syntactic structures that are identified in the syntactic analysis. The correct meaning of a particular sentence depends on the meaning of the sentences it follows and the one that it precedes. The identification of this meaning is known as discourse integration. In the last step, the analysis of what is meant and what is written is derived based on the real-world knowledge. The correct interpretation of the text is determined in this step.

Applications of NLP

Natural Language Processing has been constantly changing and enhancing the way in which humans and computers interact and thus is an essential part of majority of HCI applications. Some of the major applications include the following.

Machine translation [5] is one of the most important applications of NLP. Since there is a huge amount of information available on the internet these days, and this amount increases exponentially with every passing year, there is a need for making the information available to everyone irrespective of their language. This calls for a requirement for translation of data into different languages while maintaining the meaning of information. Since there is abundant information available on the internet, it requires a need for proper summarization of all the documents as well. This is also one of the applications of NLP. Along with summaries, the sentiment analysis can also be done to analyze the emotional meaning behind the information such as status updates on social networks. Another important application of NLP are the spam filters. Unwanted emails are analyzed and key words or phrases are searched for to determine if the email is spam or not. One of the major tasks that search engines use NLP for is question answering. Search engines put everything available to the queries in front of us, however it is still difficult for them to answer natural language queries by the users, which often leads to searching multiple keywords to get the desired results.

Apart from just text processing, NLP is now being used in speech processing as well. Speech recognition, text-to-speech generation or vice versa are also now important parts of NLP. The machine should be able to recognize the words that are said by the user, irrespective of their accent and that can be generated as text by it. Also, to generate speech data after analyzing text is also there. These are not as developed as the text processing tasks, but rapid advancements are taking place in terms of their research.

AUTOMATIC SUMMARIZATION

Automatic summarization is the process of creating a summary of a document by reducing the text document with the help of a computer algorithm [6]. The challenge here is that the summary should not miss out on any of the important elements or lose the actual meaning of the original document. Syntax and semantics both are considered when it comes to summarization. Summarization can be classified as a part of natural language processing and machine learning. Due to the problem of information overload i.e. availability of excess information, which hides the desired part of the information, the need for summarization is also increasing. Search engines like Google use summarization technologies for videos, images and textual data.

There are primarily two approaches to summarization [6]: extraction and abstraction. In the extraction based summarization, the computer just extracts important words, phrases and sentences from the original text without modifying them at all. In this, the algorithm can extract the sentences as it is from the original text and then use them in the final summary. The extraction based summarization takes less effort than the abstraction based. For abstraction based, natural language generation is used and an attempt is made to abstract the information from the original text and then represent it in natural language as the summary to the user. Majority of the summarization systems are extractive because it is very hard to design good abstraction based systems which convey the summary clearly.

TextRank Model [1]

Keyphrase extraction means to extract the list of keywords in a document. It is very essential to determine the keywords or key phrases to generate an understandable and accurate summary. One of the keyphrase extraction algorithm is TextRank, which was designed by Rada Mihalcea and Paul Tarau from the University of North Texas. It is an unsupervised approach for keyphrase extraction, which means that there is no need for training data. TextRank is a graph-based ranking algorithm for NLP. A graph based ranking algorithm is a way of deciding the importance of a vertex within a graph, by considering the global information recursively computed from the entire graph, rather than relying on local vertex-specific information. It is based on Google's PageRank algorithm **[3]** which was designed in 1999. Just like PageRank selects the important web pages, TextRank selects the important phrases and words that would be essential for generating a summary. Another inspiration for the designers of TextRank was Kleinberg's HITS algorithm **[2]**, which was very successfully used in citation analysis, social networks and the analysis of the World Wide Web. An important aspect of TextRank is that it does not require an in-depth linguistic knowledge domain, which makes it highly portable to other genres and languages.

Graph based ranking algorithms **[4]** implement the basic idea of recommendation or voting. When one vertex links to another one, it basically votes for it over the other. The more number of votes a vertex gets, the higher would be its importance in the text. TextRank creates a graph in which the vertices are the sentences and the edge weights between sentences determine how similar the sentences are to each other (in terms of natural language).

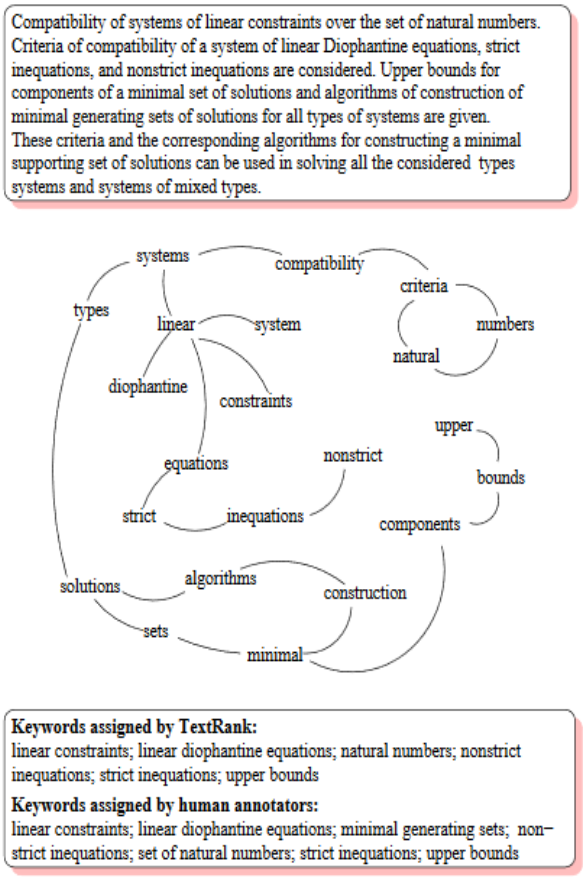


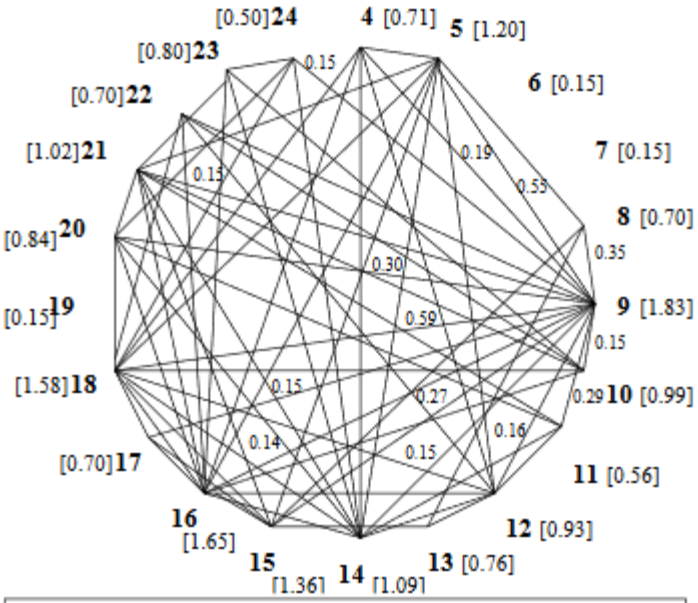
Fig 2: Sample graph for keyphrase extraction [1]

To go from a string of text to a list of scored sentences based upon how much they represent the important information from the overall text, the following steps take place: 1)Tokenize the text into sentences 2)Tokenize each sentence into a collection of words 3) Convert the sentences into graphs 4) Score the sentences.

When TextRank is implemented, firstly the original text is separated into sentences. After, that every sentence is broken down into the different words that it has. A sparse matrix of words is constructed and the number of times a word appears in a sentence is noted. Every word is then normalized with tf-idf. Tf-idf refers to Term Frequency-Inverse Document Frequency, which is a numerical statistic that is intended to reflect upon how important a particular word is to a document in a collection. Tf-idf can be considered as a weighting factor in information retrieval and text mining **[10]**. After normalization, a similarity matrix is constructed which determines how similar two sentences are to each other.

3: BC−HurricaineGilbert, 09−11 339
4: BC−Hurricaine Gilbert, 0348
5: Hurricaine Gilbert heads toward Dominican Coast
6: By Ruddy Gonzalez
7: Associated Press Writer
8: Santo Domingo, Dominican Republic (AP)
9: Hurricaine Gilbert Swept towrd the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains, and high seas.
10: The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.
11: "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly after midnight Saturday.
12: Cabral said residents of the province of Barahona should closely follow Gilbert's movement.
13: An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.
14. Tropical storm Gilbert formed in the eastern Carribean and strenghtened into a hurricaine Saturday night.
15: The National Hurricaine Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
16: The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westard at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.
17. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.
18: Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds, and up to 12 feet to Puerto Rico's south coast.
19: There were no reports on casualties.
20: San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.
21: On Saturday, Hurricane Florence was downgraded to a tropical storm, and its remnants pushed inland from the U.S. Gulf Coast.
22: Residents returned home, happy to find little damage from 90 mph winds and sheets of rain.
23: Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.
24: The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.

Fig 3(a) Scores given to different sentences [1]



TextRank extractive summary
Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil De-fense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. Strong winds associated with Gilbert brought coastal flooding, strong southeast winds and up to 12 feet to Puerto Rico's south coast.

Manual abstract I
Hurricane Gilbert is moving toward the Dominican Republic, where the residents of the south coast, especially the Barahona Province, have been alerted to prepare for heavy rains, and high wind and seas. Tropical storm Gilbert formed in the eastern Carribean and became a hurricane on Saturday night. By 2 a.m. Sunday it was about 200 miles southeast of Santo Domingo and moving westward at 15 mph with winds of 75 mph. Flooding is expected in Puerto Rico and in the Virgin Islands. The second hurricane of the season, Florence, is now over the southern United States and down-graded to a tropical storm.

Manual abstract II
Tropical storm Gilbert in the eastern Carribean strenghtened into a hurricane Saturday night. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday to be about 140 miles south of Puerto Rico and 200 miles southeast of Santo Domingo. It is moving westward at 15 mph with a broad area of cloudiness and heavy weather with sustained winds of 75 mph gusting to 92 mph. The Dominican Republic's Civil Defense alerted that country's heavily populated south coast and the National Weather Service in San Juan, Puerto Rico issued a flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

Fig 3(b): Sample graph build for sentence extraction. Manual summaries and TextRank extractive summaries are also shown [1]

The formula used in the TextRank algorithm **[1]** is:

Let G(V,E) be the directed graph with the set of vertices V and set of edges E, where E is a subset of V*V.
For a given Vi, let In(Vi) be the set of vertices that points to it (predecessors) and let Out(Vi) be the set of vertices that vertex Vi points to (successors).

The score of vertex Vi is:

S(Vi) = (1 - d) + d (Summation of j where j ranges in the set of In(Vi)) 1/|Out(Vj)|*S(Vj)*

where d is the damping factor ranging from 0 to 1.

PROBLEM STATEMENT

TextRank is an extremely useful and good algorithm. The best part about it is its simplicity in terms of functioning. The basic concepts of graph theory are combined with the PageRank to create a simple algorithm to obtain summarization. However, the simplicity of the algorithm works against it when a less precise summary is obtained. There are instances where the algorithm fails to identify the key words and ends up losing some information in the final summary. To reduce this problem, certain improvements can be made by combining the TextRank algorithm with other algorithms. This might make it possible for the combined algorithm to identify key words more easily and facilitate the process of keyphrase extraction.

PROPOSED IMPROVEMENT

There are two proposed improvements for the algorithm. It is believed if either or both are implemented along with the TextRank algorithm, it would make it more efficient to identify the key words and thus facilitate the keyphrase extraction. The two concepts that can be added with TextRank are 1) Intersection function and 2) Sentences Dictionary. A new individual algorithm using both the mentioned concepts is proposed. It would make the TextRank algorithm more efficient if a way to combine all the advantages of it and the new algorithm are combined.

Intersection function

This function receives two sentences, and returns a score for the intersection between them. The sentences are split into words/tokens and the number of common tokens are counted among the two sentences. This would make two sentences having similar meanings to have more common tokens among them than the others and thus help in identifying the key sentences in the original text.

Sentences dictionary

The sentence dictionary could become the “heart” of the algorithm if it is combined with TextRank. It would receive the text as input, and calculate a score for every sentence. The calculations consist of two steps. In the first step, we split the text into sentences and then store the intersection value between every two sentences in a matrix (which is a two-dimensional array). So, for instance, the values [0][2] will hold the intersection score between sentence no. 1 and sentence no. 3. The text is converted into a fully connected weighted graph. Each sentence is a node in the graph and the two-dimensional array holds the weight of each edge.

In the second step, an individual score can be calculated for each sentence and stored in a key-value dictionary, where the sentence itself is the key and the value is the total score. All the intersections with the other sentences in the text is summed up together (not including itself).

The score for each node would be calculated in the graph, which can be simply done by summing all the edges that are connected to that node.

Building the summary

The final step in the algorithm is generating the final summary. This can be done by splitting the text into paragraphs. This is where the actual role of the sentences dictionary comes in. The best sentence is chosen from each paragraph per the sentences dictionary. The idea here is that every paragraph in the text would represent some logical subset of the final graph. Hence, we pick the most valuable node from every subset.

DISCUSSION

There are two main reasons why the algorithm works. The first reason is that a paragraph is a logical atomic unit of a text i.e. there is a very specific reason why the author or creator decided to split the text in the way that he did. Every paragraph logically contains information about a particular idea or subtopic. Thus, taking the paragraph as a base to divide into sentences is a good idea. The second reason why the above methodology would work is because of the intersection function. If two sentences have a high intersection, there are high chances that both of them convey the same information in the text. Hence, if one sentence has a good intersection with many other sentences, it means that the author/creator is just repeating the same information in the text, which makes the other sentences as redundant. Hence, while creating the summary, redundancy would be avoided to a large extent. This would lead for an overall better keyphrase extraction, and increase the efficiency of the summarization.

CONCLUSION

In conclusion, the paper gives a new algorithm which combines the concepts of intersection function and sentence dictionary. These concepts can be easily combined with the original TextRank algorithm to increase the efficiency of the summarization. According to some of the previous results by Mihalcea and Tarau[1], the TextRank algorithm was almost 77% correct, when it was compared with a manual human summary of similar texts. It is believed that if the new concepts are added along with the TextRank, it would increase the accuracy and efficiency of the entire algorithm and provide better results.

REFERENCES

[1]R.Mihalcea and P. Tarau, "*TextRank: Bringing Order into Texts"*

[2] J.M. Kleinberg. 1999. "*Authoritative sources in a hyperlinked environment."* Journal of the ACM, 46 (5) : 604 -632

[3] S. Brin and L. Page. 1998. "*The anatomy of a large scale hyper-textual Web search engine."* Computer Networks and ISDN Systems, 30(1-7).

[4] R. Mihalcea. 2004. "*Graph-based ranking algorithms for sentence extraction, applied to text summarization."* In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004) (companion volume), Barcelona, Spain.

[5] Hutchins, J. (2005). "*The history of machine translation in a nutshell*"

[6] Mani, Inderjeet (2001). "*Automatic Summarization"*. ISBN 1-58811-060-5

[7] "SEM1A5 - Part 1 - A brief history of NLP"

[8] Chomsky, Noam (1965). "*Aspects of the Theory of Syntax*." MIT Press.

[9] Chomsky, Noam (1995). The Minimalist Program. MIT Press.

[10] Rajaraman, A.; Ullman, J. D. (2011). Data Mining. "*Mining of Massive Datasets"* pp. 1–17. ISBN 9781139058452.