VOL. 12, NO. 19, OCTOBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

QUERY BASED TEXT SUMMARIZATION

Shail Shah, S. Adarshan Naiynar and B. Amutha
Department of Computer Science and Engineering, School of Computing, SRM University, Chennai, India
E-Mail: shailvshah@gmail.com

ABSTRACT

With the increasing demands of solutions to the problems in the field of Artificial Intelligence and Natural Language Processing is one of the most challenging tasks. Query Based Text Summarizer is one of the most explored topics in Natural Language Processing which involves processing and comprehending of text document with an appropriate result based on an input query. There have been many models and structures for a text summarizer which generates effective results; there have been very few approaches towards an extension of this problem. Query based text summarizer is based on sentence-sentence and sentence-word relationship using graphs structure. Several methods and algorithms based on statistics and linguistic techniques have been adopted in the past, however in order to maximise its results, a combination of these techniques must be applied to make it more efficient. This paper aims to solve the righteousness of the output that is being generated.

Keywords: text summarizer, natural language processing, word sense disambiguation, graph based IR.

INTRODUCTION

Query based Text Summarization involves selection of the key phrases and sentences related to the query from the given information and ensuring them to be in a readable form that is understandable by the user. Most of the existing approaches either completely rely on statistical techniques or on linguistic techniques, it is therefore required to build a more comprehensive model in order to obtain the best result. A graph structure is used in our model because it has capability to transform naturally, the meaning and structure of a cohesive text in a document. The entire process is carried out in a series of steps. The query is taken from the user and is tokenized i.e. broken into individual words, punctuations and stop words. Furthermore, this method uses graph comparison of vertices and the overall aim of this method is to improve the righteousness of the information retrieved by the process of matching text sentences against queries in order to obtain relevant sentences or phrases from the document. Once the query is matched with relevant information, relative scoring of sentences including the key words present in query is done using TF-IDF scoring and in order to avoid ambiguity in the sense of the key word in the listed sentences, an algorithm using knowledge repository with dictionary files in them has been implemented. All the ambiguous key word present in the list is selected and those sentences are removed from the list. Now, TextRank algorithm is used to rank the sentences according to their importance. Finally summary of the query is generated. There are 2 types of summaries:

a) Indicative

b) Informative

Indicative summaries as the name suggest roughly indicates to the content from the original document, and doesn't necessarily contain original content from the document.

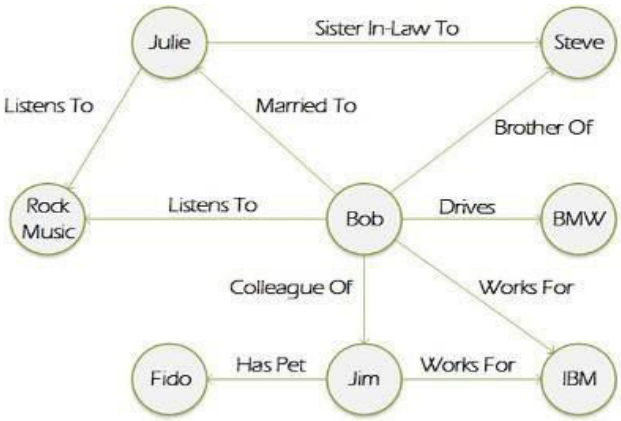

Figure-1. Network of graph.

Whereas, informative summaries have the original content from the document. In informative summaries there are two types:

a) Extractive

b) Abstractive

Extractive summaries have the original content which includes the words/sentences present in the document.

Abstractive summaries carry only the essence of the original text but its words/sentences vary from the document. In this paper an informative extractive summary of the query is generated.

LITERATURE REVIEW

A. Pierpaolo Basile, Marco de Gemmis, Pasquale Lops and Giovanni Semeraro, Solving a Complex Language Game by Using Knowledge-Based Word Associations Discovery, IEEE Transaction on Computational Intelligence and AI in games, Vol. 8, No. 1, March 2016:

This paper aims to develop an artificial player to play, "The Guillotine" game which requires a broad range of knowledge dataset which has a wide range of topics. This player needs human intelligence to solve the problem. The entire paper is about connecting the dots or clues given by the user. The method proposed in this paper is to use a spreading activation algorithm on the clues with respect to the knowledge database. Once implemented, these scores are used to weigh the clues of the game with the words within the knowledge repository, giving a list of possible solutions. Indexing and ranking algorithms are further applied to these candidate solutions to select the word which is most probable. This approach has a good scope in the field of query expansion and information retrieval and other NLP related tasks. Although this may seem to be the best approach, it fails to acknowledge the space complexity and time complexity of this approach.

B. R. V. V. Murali Krishna, S. Y. Pavan Kumar, Ch. Satyananda Reddy, A Hybrid Method for Query based Automatic Summarization System, International Journal of Computer Applications (0975 8887) Volume 68 No.6, April 2013:

This paper is based on the hybrid techniques used to calculate the relationship between sentences in a text document and the various queries asked based on the given document. Many linguistic and statistical approaches previously used before have been mentioned to find the relationship between the query and the sentences in the document, then these sentences are discarded based on the scores of the relation using a scoring algorithm. Finally redundant data is filtered using iterative clustering algorithm. The method proposed in this paper leverages the merits of individual algorithms and methods in order to obtain an efficient and optimized relationship between the sentences. Although it misses a key element, which defines the semantic relatedness of the context of the query.

C. Query-Based Summarization Based on Document Graphs - Ahmed A.Mohamed, Sanguthever Rajasekaran:

This paper describes a directed graph model of a document, it comprises of concepts and intents in the document having different types of relationships like "is a" or "related to" which is further used with the help of POS tagging. Basically, this paper aims to form these graphs for each sentence in the document and the query. Finally they run similarity check between these graphs recursively to get the best possible solution. Based on a lot of assumptions and particular dataset these results can give effective results due to which generic documents can be evaluated easily, but not all types of documents. Although this approach suggest, a more methodical way to solve the problem it clearly ignores many other aspects of solving the problem.

METHODOLOGY

The entire process of generating an output is divided into 3 distinguished task, these are:

a) Information Retrieval

b) Word Sense Disambiguation

c) TextRank

All these tasks are performed using graphs. Graphs are used over trees and other data structure due to its flexibility and accessibility from any point at any time. Due to all these factors, it makes graph more efficient. The given tasks are explained in detail below:

Information retrieval

Information retrieval is basically how the computer can effectively obtain or retrieve specific information from a document. The model proposed is to use graph-comparison recursively to get a better similarity score (Rada Mihalcea and Paul Tarau 2004) which further enhances the relevancy of the sentence with the query. Since graphs can capture structure and relations between nodes, and thus project a wide range of relations between the data, we have chosen graph theory. Our approach is to represent sentences and queries as nodes. The most standard way to represent such sentences is to associate it in n-dimensional vector from each sentence to the query, where n is number of indexing words or phrases (node). Finally the framework for this approach is a bipartite graph where query is the set of vertices with multiple incoming directed edges while sentences are the set which has single outgoing edge to a particular vertex. These edges are indexed in order to have a pointer and relation between each sentence-query pair. Edges correspond to the link that exists between sentences and queries that have weights of their score from TextRank. Furthermore, graph edge settings reflect the TF-IDF (Term Frequency-Inverse Document Frequency) paradigm.

Let $G = (V, E)$ be a citation graph, where V be the vertices representing sentences and E be the directed edges representing words or clues. Let,

$h_k \; and \; a_k$

Be the hub score and the authority score of vertex k respectively (Kleinberg, 1999). The hub and authority scores of vertex k is computed using,

$$h_k = \sum_{i(k,i)\in E} \; and \; a_k = \sum_{i(i,k)\in E}$$

These are represented in the form of matrices and computed. The similarity between two vertices j and k from the first graph and second one respectively is computed using the similarity scores between their related. Therefore, similarity can be found using:

$$M_{k+1} = BM_kA^T + B^TM_kA$$

The convergence property of the above equation is essential for the calculation of similarity between. Furthermore, normalizing the similarity matrix S can be used to solve the convergence problem at each iteration step.

$$\mathrm{M_{k+1}} = \frac{\mathrm{BM_kA^T + B^TM_kA}}{\|\mathrm{BM_kA^T + B^TM_kA}\|}$$

In order to satisfy similarity measure, the given conditions must be satisfied:

1) $(\mathrm{k,j}), \mathrm{M(k,j)} \geq 0$
2) $(\mathrm{k,j}), \mathrm{M(k,j)} = \mathrm{M(j,k)}$
3) $(\mathrm{k,j}), \mathrm{M(k,k)} = \mathrm{M(j,j)} \geq \mathrm{M(k,j)}$

The algorithm mentioned above compares graph vertices from two graphs, as discussed earlier the iterative algorithm converges to a similarity matrix MAB between nodes of graph A and graph B. The entire algorithm is given below:

$$M_0 \leftarrow 0, k \leftarrow 0$$

$$A \leftarrow A + \sum_{n=2}^{\infty} f_2(n)\, g_2\left(\frac{A^n}{\|A^n\|}\right)$$

$$B \leftarrow B + \sum_{n=2}^{\infty} f_1(n)\, g_1\left(\frac{B^n}{\|B^n\|}\right)$$

$$Repeat\ until\ convergence\ achieved\ for\ k\ even,$$

$$\left|\begin{array}{l} M_{AA_k} \leftarrow \frac{AM_{AA_k}A^T + A^TM_{AA_k}A}{\left\|AM_{AA_k}A^T + A^TM_{AA_k}A\right\|_F} \\ M_{BB_k} \leftarrow \frac{BM_{BB_k}B^T + B^TM_{BB_k}B}{\left\|BM_{BB_k}B^T + B^TM_{BB_k}B\right\|_F} \\ M_{AB_k} \leftarrow \frac{BM_{AB_k}A^T + B^TM_{AB_k}A}{\left\|BM_{AB_k}A^T + B^TM_{AB_k}A\right\|_F} \end{array}\right.$$

$$k \leftarrow 1$$

$$M_{AB} \leftarrow \cdot \frac{M_{AB} \cdot * M_{AB}}{diag(M_{AA}) \cdot * diag(M_{BB})^T}$$

$$Output\ M_k\ (k\ is\ even)\ as\ similarity\ matrix$$

Once the required sentences are acquired they are put together in a separate list.

Word sense disambiguation

This task is carried out once the information is retrieved from the document and is moved to a probable solution list. In this section, the Word Sense Disambiguation is handled. WSD is used to identify or distinguish same word with different which can be used in different context. For example, a plant can be a green plant or it can be a factory. In order to remove such ambiguity in the answers to the queries a sequential step must be carried out. For this to work the system must have a knowledge repository. A knowledge repository is the collection of files and documents stored in the database that is easily accessible. In our knowledge repository the following sources are used:

a) Dictionary

b) Compound forms

c) Movies, Books, Songs

d) Proverbs

Since information stored will be in textual format, bag-of-words (BOW) model is chosen to represent the textual information in NLP. Every sentence in this model is individually represented as a group of words, disregarding its order or context, but retaining its multiplicity. In this experimental model, each source is a database with such sentences i.e., text pieces which are equivalent to basic units, represented as BOW's. For weighing the terms we have used TF-IDF, which is a standard method that we have opted to compute the occurrences of a word in a document and frequency of words in the entire corpus. A text fragment is called a Cognitive Unit (CU), as it helps the machine to understand. The main advantage of this format is that it can used as a vector and can be developed in graph structure. Once these BOW's are created, using the spreading activation algorithm (Pierpaolo Basile, Marco de Gemmis, Pasquale Lops and Giovanni Semeraro 2016) appropriate linkages and weights are assigned to it. Now, the process starts from each and every Cognitive Unit which has triggered the search process over the entire graph network. Likewise, every node N_i in the network has an associated activation level AL_i attached to it and is a real number within the range $(0.0 \ldots 1.0)$. This number in AL_i represents the stimulus level of the node N_i. Eventually the nodes are ranked based on the activation level values in descending order, and the equivalent nodes (words) are included in the list called the candidate solutions, which will further compare it to the context of the document using the reverse path algorithm, where it will validate maximum number of words that are present in the context with the document. For example, the word "third estate" is a common entity for the words French revolution, city workers, farmers, social privileges. Although the term maybe used in all these context, this algorithm looks for the relating words in the document and finally throws the context in which it is used.

TextRank

We have opted to choose TextRank model to further sequence the order of output sentences due to its model which is followed by "recommendations", which basically counts the score of the vertices that link with the keywords of the query (in other words context) and keeps a count of each incoming vertex to that particular keyword. This is mainly done once we obtain the candidate list of sentences from Section III.A and filter

few sentences further from Section III.B. Since we have list of probable sentences that ultimately form the output and we also have all values of weighted graph, now we run can run a ranking algorithm on these weighted graph and order them in descending order. To explain how these weighted graph works, Let's assume $G = (V, E)$ be a directed graph where V are the vertices and E are the edges, also E is a subset of $V \times V$ and strength of each linkage word W_{ij} that defines the relation between the two vertices in terms of a numeric score. This score of a vertex V_i is defined as follows:

$$B = \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum_{V_k \in Out(V_j)} W_{jk}} WS(V_j)$$
$$WS(V_i) = (1 - d) + d * B$$

Where d is a damping factor that is set arbitrarily between 0 and 1 initially, which has the task of getting into the model, basically a model that accounts the probability whether or not to jump into another vertex in the graph. However, in this model the graphs are in natural language, and include many different types of links between the vertices, that it is useful to use this into the model to further enhance the connection between the vertices V_i and V_j as a weight W_{ij} which is added to the respective edge which connects the two vertices.

Finally the proposed system is broken into two modules:

A. Document scanning

B. IR and Proposed System.

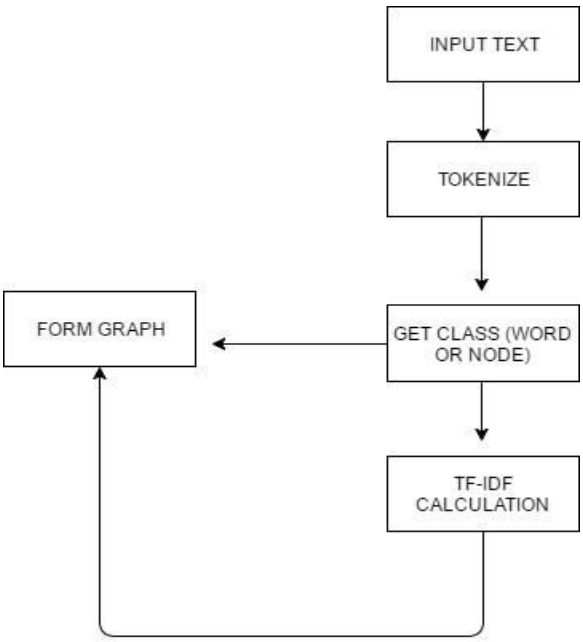

Figure-2. Document scanning.

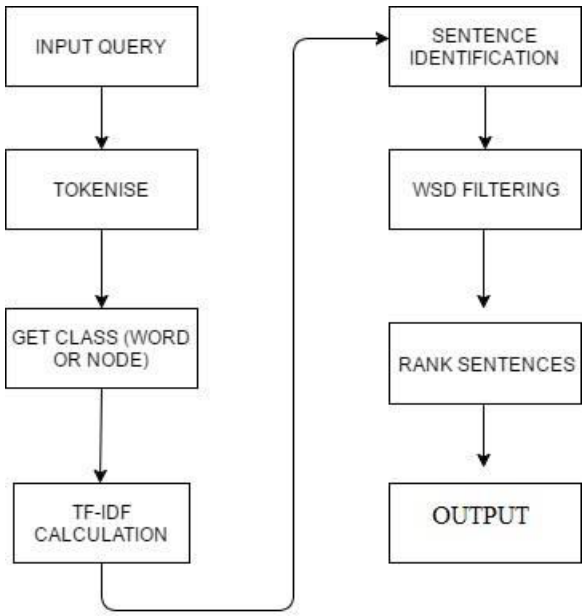

Figure-3. IR and proposed system.

ADVANTAGES

The advantages of this approach include:

a) Better accuracy in terms of output that is generated.

b) More efficient in terms of traversals as the information retrieval is much faster due to the use of graph structure.

c) Eliminates the word sense ambiguity.

d) TextRank is advantageous because of it recursive nature.

CHALLENGES

The challenges of using the proposed methodology are:

a) Accuracy is not guaranteed i.e. it might be less or it might be more depending upon the clarity of the document.

b) The process is slow as a result of capturing the sense of data and storing it.

c) It is a highly complex system with a requirement of large database.

d) This approach cannot handle complex queries.

FUTURE RESEARCH INTEREST

This whole field of Natural Language Processing and its application in Artificial Intelligence is an area full of research. Therefore the further research interests of this paper is using machine learning to make the program learn from mistakes, alternate algorithm for ranking text in more

efficient ways, algorithms for efficient storing and querying words or sentences by training and classifying them.

CONCLUSIONS

The main application of this paper is to reduce the human effort put in order to summarize the text. The use of graph structure will provide a sense of realism to the data captured as in reality there is no hierarchy in data just connections. The algorithms used are also effective in dealing with large scale datasets. Plenty of research was done before coming up with this paper. We give the abstractive summary based on the query of the document.

REFERENCES

[1] Pierpaolo Basile, Marco de Gemmis, Pasquale Lops and Giovanni Semeraro. 2016. Solving a Complex Language Game by Using Knowledge-Based Word Associations Discovery. IEEE Transaction on Computational Intelligence and AI in games. 8(1).

[2] R. V. V. Murali Krishna, S. Y. Pavan Kumar, Ch. Satyananda Reddy. 2013. A Hybrid Method for Query based Automatic Summarization System. International Journal of Computer Applications (0975 8887). 68(6).

[3] Rada Mihalcea and Paul Tarau, TextRank: Bringing Order into Texts.

[4] Saeedeh Gholamrezazadeh, Mohsen Amini Salehi, Bahareh Gholamzadeh. A Comprehensive Survey on Text Summarization Systems, Conference: Computer Science and its Applications, 2009. CSA ’09.

[5] Quoc-Dinh Truong, Taoufiq Dkaki, Josiane Mothe, Pierre-Jean Charrel. Information retrieval model based on graph comparison, 9es Journes internationalesd Analyse statistique des Donnes Textuelles.

[6] Antonio Jurez-Gonzlez, Alberto Tllez-Valero, Claudia Delicia-Carral. Manuel Montes-y-Gmez and Luis Villaseor-Pineda, Using Machine Learning and Text Mining in Question Answering, INAOE Publications, National Institute of Astrophysics, Optics and Electronics - Mexico.

[7] Query-Based Summarization Based on Document Graphs - Ahmed A.Mohamed Sanguthever Rajasekaran.