INFORMATION EXTRACTION FOR NOTES GENERATION

Jaideep Shinde[1], Viraj Kamble[2], Shamli Mali[3], Ajinkya Mohite[4]

[1]*Computer Enginerring, NBNSSOE, Maharashtra, India*
[2]*Computer Enginerring, NBNSSOE, Maharashtra, India*
[3]*Computer Enginerring, NBNSSOE, Maharashtra, India*
[4]*Computer Enginerring, NBNSSOE, Maharashtra, India*

ABSTRACT

For a specific topic we find many reference books and textbooks to refer and the content is much more than one needs, therefore at the time of revision or quick study we need the summary and short version of the whole content for revision, especially at the time of examinations. Therefore we aim to develop a semi-automated technique to generate notes from English text documents like Reference Books and Text books. The technique discussed is considered to be a pioneering attempt in the field of NLP (Natural Language Processing).This technique has a wide scope in the educational domain. The technique when implemented as an application can be used by both faculty members and students.

Keyword : - *NLP, Segmentation ,Parsing, Ontology, Text Summarization, Text Extraction*

1.INTRODUCTION

Here we discuss the idea of summarizing a text document into more understandable document with less content which covers most of the important points and gives a fair idea about the whole document, like a summary.

Generally, in most of the textbooks, only 20-22% of the words contain the information you need to understand or need while revising the topic. They are known as keywords. And the remaining 80% contains close to no essential information as such, which consists of pronouns, connectives like “of”, “has”, “for", etc. The only purpose of these words is to link the keywords together to form meaningful and sensible sentences. They are useful for first time reading, but for revisions, just the keywords can do the work efficiently, so revising just a small amount of text is much beneficial as reading the whole text again.

[1]There are many techniques involved such as information extractor which combines certain NLP methods like chunking, segmentation, summarization etc, with certain special linguistic features of the text such as the ontology of words, semantic links, noun phrases found sentence centrality etc. The process of the technique comprises of extracting text, creating an ontology, identifying important phrases for bullets and generating brief summary accordingly.[1]

We achieve this by using various methodologies and tools like Parsing, ontology creation, segmentation etc which are discussed further. For a given text document like a reference book we find too much information on a topic which is more than one needs or can handle, to tackle this problem we can make notes and study a topic, here we try to accomplish the generation of notes automatically with abstractive approach. This application can be used to get a

gist of a given document and can be used for quick study like revision and omit the information from a document which is not much useful.

To achieve higher level of accuracy in [2] document classification more informative features of documents are taken into account. For this purpose, for instance, weight is assigned to HTML tags, which affects the efficiency of information retrieval; these are defined using genetic algorithms. Documents classification takes place at the level of separate words, but not like classical works, the relevance of each word here is defined in relation to their informative features, which are the occurrences of a word in the title, emphasis is given on words by means of italic, bold fonts or its underlining and position of a word on the page. A DIG (Document Index Graph) algorithm is based on graph theory and phrases and their weights are taken into account for making suggestions in work [2].

In this approach for text segmentation,[3] we use an efficient linear text segmentation algorithm (called TSHAC).It considers both computational complexity and segmentation accuracy. The process of TSHAC has 4 steps. First, Preprocessing of long text; tokenization, stopwords are removed, and stemming are conducted to construct the vocabulary of the text. Text is then represented as vector after text preprocessing, each of which represents a sentence within the text. A part of sentence similarities are then computed to construct the sentence-similarity matrix. Finally, a the optimal topic boundaries are identified by the proposed algorithm.[3]

The method for extraction of meaningful sentences from the text to summary based on definition of score of relevance for each of the sentences and called "sentence by sentence" is suggested in this section.

The technique to be used is domain-independent which makes it unique from other techniques. From the evaluation measures applied to the technique, we can say that the technique helps in semi- automated generation of notes. The performance totally depends on the ontology provided as input. The more accurate the ontology creation, the more accurate the output will be.

2. METHODOLOGY

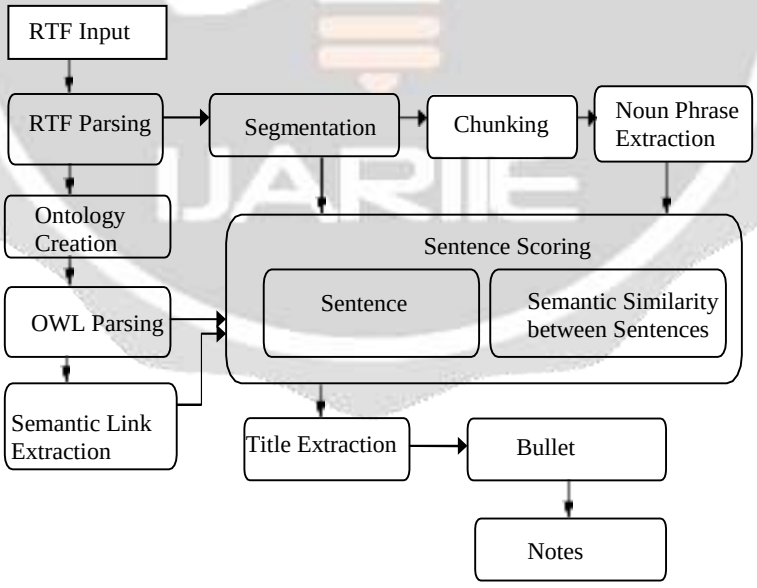

Fig -1: Architecture

- **RTF PARSING :**

  It accepts the input as a RTF document parses the Bold, Italics and Underlined text and stores it. Finally, the module converts the input to a TXT document.

- **SEGMENTATION :**
  The Segmentation module takes the input as a TXT document, divides the text into segments using the Text Tiling Algorithm and outputs the segments.
  A better alternative to the text tilling algorithm is the linear text segmentation using Agglomerstive Clustering.

- **CHUNKING :**
  Chunking module represents the chunker tool known as MontyLingua, which converts the segments into chunks and outputs the chunks. It can also be done using NLTK in python.

- **NOUN PHRASE EXTRACTION :**
  The Noun Phrase Extraction module extracts noun phrases from the chunked text.

- **ONTOLOGY CREATION :**
  Ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. The reasons about the properties is given by this domain, and also used to define the domain. The tool extracts the ontology and outputs it as an OWL file.

- **OWL PARSING :**
  In the OWL parsing module, the OWL file is taken as input and parsed to create an ontology tree and the tree is maintained as an Adjacency Linked list.

- **SEMANTIC LINK EXTRACTION :**
  In the Link Extraction module, the adjacency linked list representation of the ontology is used to find the semantic links and the physical links between sentences. The output is maintained as Relational Matrix.

- **SENTENCE CREATION :**
  The Sentence Scoring module is the critical and most important component which takes multiple inputs such as the segments, noun phrases, the adjacency linked list representation of ontology and the relational matrix of semantic links. It performs scoring and assigns weightages to the segments and sentences of each segment, using which the important phrases are identified and given as output.

- **NOTES FILE :**
  This will be the final output of the application.

3. CONCLUSION

The proposed technique is domain-independent, so we can extract notes from any document without any auxiliary knowledge base which makes it unique from other techniques. From all the evaluation measures applied to the technique, we conclude that the technique generates notes using a semi automated notes generation system. The performance totally dependent on the ontology provided as input. The more accurate the ontology, the more accurate the output or the generated notes will be.

4. REFERENCES

[1]. K.Gokul Prasad, Harish Mathivanan,Madan Jayaprakasam,T.V.Geetha, "Document Summarization and Information Extraction for Generation of Presentation Slides ",***2009 International Conference on Advances in Recent Technologies in Communication and Computing***

[2]. Alguliev, R.M, Aliguliyev, R.M, "Effective summarization method of text documents", ***Web Intelligence The 2005 EEE/WIC/ACM International Conference, pp.264 – 271, Sept. 2005***

[3]. Ji-Wei Wu, Judy C.R. Tseng, Wen-Nung Tsai" An Efficient Linear Text Segmentation Algorithm Using Hierarchical Agglomerative Clustering",**Feb.2011**