ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 01-11 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp01.pdf

Towards Automatic Generation of Smart Summary: Design of a Multioptions Summarizer using Syntactic Chains

Maryam Abu Olaim
maryamabuolaim@yahoo.com

ABSTRACT

The process of document summarization has been studied for more than 60 years in order to achieve an acceptable level at the field of automatic generation of textual summary. But, the demand for generating a smart summary which includes a mixture of helpful forms is still clear. This paper presents an automatic multioptions summarizer based on selecting the best sentences from a given document as a textual summary, converting this summary into a structured form using syntactic chains and then generating three types of summarization: a structured outline, a query by document and a concept map. This summarizer is represented by a novel algorithm which aims at generating a good summary that could reflect the overall picture of a given document by different understandable forms. The implementation of the proposed algorithm has been evaluated with encouraging results.

Key words : Automatic Summarizer, Smart Summary, Syntactic Chains, Text Chunking, Text Summarization.

1. INTRODUCTION

For academic research, literature review is necessary for the classification, presentation and evaluation of what other researchers have written on a specific topic which requires academic skills for outlining what others have done in an area of interest to review the literature [27]. As an effective starting point for the revision process, outlining documents is a helpful tool for gathering notes about what was written in research area. But, it can be challenging for users to extract key phrases to identify the right resources to attach, especially when dealing with large resource collections [1] [32] . So, text outlining can be an effective technique for distilling the main topics by editing a text with the process of condensing a long research paper into a shorter one [22] for taking notes to see the overall points of the text. On other hand , the first stage in the text summarization process [16] is extracting its "aboutness" as a shortest summary taking into account sentence structure [13]. However, in early work on automated summarization it was assumed that the most frequently occurring context bearing words reflect the most important text content and a good indicator for important concepts. But the weak point of this method is that it does not assign any importance to the semantic connections between the words. In the attempt to take the content analysis beyond the word level, analysis methods depend on capturing the dependency relations between the text segments and spot the concepts rather than its word components [16]. Typically, automatic generation of brief summaries focuses on different problems such as sentence extraction, processing structured templates, and creating a headline style summary from a text [11]. Although this area has received a great deal of attention in recent research, the need for a highly efficient tool that produces usable summaries is clear [26]. However, there are many textual summaries schemes such as flat summary with a non-structured summary form and Smart summary where the summarizer automatically provides the best possible type of summary with the optimal length by analyzing the structure of a document. In fact, there are some summarizers already commercially available in the market such as Copernic®, Sinope® and AutoSummarize that generate summary reports for the best sentences and concepts of the text by processing documents, e-mail messages, hyperlinks, web pages, or files [23]. So far, what most summarization systems do is to extract the most important text sentences from the source input [13] or the headline-style summaries [17] which adopt a form of compressed English in unconnected sequences of terms [15] that help to get a quick idea of the content [11]. But, in the current summarizers, there are some issues that still need to be addressed include ways to generate a smart summary from documents that can assist academic researchers in their tasks, by enabling them to easily understand the overall picture of the document and at the same time quickly determine a Query by Document (QBD) to be used by any search engine. Fortunately, by its function, concept maps can also be useful to serve as indicative summaries [45] and considered better than other forms of text summarization [4]. Consequently, this paper presents a proposed algorithm for designing a new summarizer to generate multioptions of descriptive summary forms for a given document such as a QBD and a concept map by exploiting the nature of the concept map that involves both structure and content characteristics of the map [6] which consists of a set of propositions represented by concept-link-concept triples as a meaningful statements about some object [31]. The designed multioptions summarizer is a summarization software system that allows researchers to choose the type of summary they need to be used as an assistance tool in their research works.

The remainder of this paper is as follows; in Section 2, the literature review is described and section 3 discusses the role

of natural language processing and text chunking in building the proposed summarizer. Section 4 explains the methodology used for designing the proposed summarizer and the details of its algorithm. Experiments and evaluation are introduced in section 5. Finally, section 6 underlines conclusion and future work.

2. LITERATURE REVIEW

Hongyan et al. [18] built a novel cut and paste based text summarization system where the input to that system was a single document from any domain by extracting key sentences using existing tools such as current summarizers. Their summarizer edits extracted sentences, using sentence reduction removal inessential phrases and combination to merge resulting phrases together as coherent sentences. Shiyan et al. [7] reported three different summarization approaches: a sentence-based summary that extracts important sentences using various features, another sentence-based summary generated by extracting research objective sentences, and a variable-based summary focusing on research concepts and relationships. Their evaluation results indicated that the majority of users (70%) preferred the variable-based summary, while 55% of the users preferred the research objective summary, and only 25% preferred the sentence-based summary. Qazvinian et al. [10] presented a summarization methodology to find important contributions of scientific articles based on considering the key-phrases that have been repeated in more sentences are more important. Yuen-Hsien et al. [12] described a series of text mining techniques such as text segmentation, summary extraction and feature selection to verify the usefulness of segment extracts as the document surrogates. Alam et al. [23] proposed a design of a Commercial Summarizer that combines document analysis, structural decomposition, XML representation and lexical chain analysis. Their proposed summarizer is compared to three commercially available summarizers: Copernic®, Sinope® and AutoSummarize. Their summarizer could generate a flat summary that is coherent and meaningful . But, they concluded that there still need to generate a good summary from documents that have specific constructs [23]. To generate a good summary, Inderjeet et al. [20] addressed the problem of revising summaries to improve their quality to achieve both indicative and informative functions by performing three types of operations: elimination, aggregation and deleting extraneous information. However, the most researches in this field attempt to generate a good summary by means of understandable and expressive forms at a higher level of abstraction. In this paper, a new algorithm is proposed with a vision of building a structured summary that can be used as an initial base to construct a smart summary with understandable and usable forms.

3. THE ROLE OF SYNTACTIC CHAINS IN BUILDING A STRUCTURED SUMMARY

In natural language processing (NLP), some coherence chunking approaches could significantly improve the quality of text summarization based on dividing sentences in chunks[4] using grammar patterns [5] for avoiding sentence ambiguity and part-of speech (POS) for recognizing the boundaries of specific phrases[1], identifying their types and distinguishing non-overlapping text chunks syntactically [3]. Accordingly, the proposed algorithm in this paper acts as a way to advance the existing text summarization based on deep syntactic analysis by developing a new text "Chunker" for specific purposes such as concept mapping and query generation. Firstly, The meaning of the complete sentence is assigned by the meaning of the words and the pattern in which these words are arranged. Accordingly, the semantic relationship between the concepts is denoted by the words organized in patterns, while a syntactic representation of a sentence is a data structure that represents how a set of surface words are arranged in a sequence to form a pattern [25]. Identifying semantic relationships in text involves looking for particular linguistic patterns in the text that point to the existence of a particular relationship using pattern-matching to recognize the segments of the text or the parts of sentence that match with each pattern [21]. The recognition of the patterns demands structured data, but text to some extent is unstructured. This text mining problem can be considered as enforcing structure on text to make it obedient to the analytic techniques of data mining. Natural Language Processing (NLP) could be a powerful tool for text mining and consequently for text summarization, because it can differentiate how words are used such as by sentence parsing and part-of speech (POS) tagging, and so might add significant power to statistical text analysis [19]. In addition, punctuation marks provide obviously unambiguous word boundaries by segmenting long expressions into their component words. So, this research introduces a new text summarization algorithm based on NLP tools in order to signal semantic relationships between the concepts. In this algorithm, a syntactic-oriented approach is used for feature selection criteria in both concept detection (e.g. nouns or proper nouns are selected as concepts) and relation detection by using the shallow parsing for chunking purpose and to output the chunked text. In addition, in an attempt to classify the contents of the text, a rule-based method is adapted to detect all concepts, relations, and segmentation points with some proposed patterns for the purpose of matching with specific syntactic parsed (tagged and chunked) fragments. Initially, the text is converted into a structured form as a table that consists of three classes; concept, relation, and segmentation point. After that, this table is used for finding all the potential forms of propositions as a syntactic-chain. For this purpose, two specific patterns are defined for matching with the contents of the created table. Finally, for

ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 01-11 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp01.pdf

generating a structured form of a summary, a set of rules are suggested to construct a headline-style list of summary in order to shape the overall outline. For example, to achieve the goal of extraction all potential propositions, two main rules are defined to be applied through the process of generating an outline from as a structured summary; which are:

Rule 1: The first {concept –relation-concept} pattern found in a segment must be selected as the first unit of a syntactic-chain.

Rule 2: All the subsequent {relation-concept} patterns in the same segment must be added to the first unit of that chain.

4. METHODOLOGY

As a preprocessing phase for the automatic generation algorithm of a smart summary, the input document has to be summarized by a proper text summarization tool that can express extracting the most important sentences throughout the text based on its key concepts into an unstructured flat summary. First phase of this algorithm is outlining the flat summary into a headline-style form. For this phase, some current available summarizers such as Copernic Summarizer are exploited to produce a comprehensive summary, because, this summarization technologies could significantly create concise document summaries using sophisticated linguistic and statistical algorithms by identifying the key concepts and extracts the most related sentences, resulting in a summary that is a shorter, condensed version of the original document [29]. In addition, to introduce the author's view or main points, the focus is on position-based mode of text summarization that depends on the sentences occur at the beginning as the most important ones, therefore, the whole summary of the original document is obtained, afterwards, only the first sentences with no more than 20% of the summarization are used for analysis, editing, and transforming them into an outline form using NLP techniques to automatically extract information from unstructured text through a detailed syntactic analysis of the previous summarized text. Chunking as a shallow parser is used to tag each word with its POS and exposes the noun phrase and verb phrase chunks for the whole text. By adapting a new method for syntactic analysis, the next phase is extracting structured data as a regular organization of entities and relationships from unstructured text based solely on chunking as a natural language processing for parsing the text and detecting sentence fragments to build a system that aims at extracting propositions. Therefore, a software system is built with beginning by chunking the text as its first input for the process of building two arrays: type-classifier array and candidate concept-relation array. To produce a new summary of the original document, this system takes the classification arrays of the produced structured data as a new input to generate a list of headlines which form the final outline as its first output, followed by creating the simplest form of a concept map as the second output, and finally, generating a QBD from the created concept map using all its concepts as a candidate key phrases with a combination of using the most frequently words in the original document to select the key phrases which are used to form a QBD as a final output of the developed system.

Initially, the algorithm of the proposed multioptions summarizer is explained with the following steps:

Step 1. Summarizing a document into its most important sentences.

Step 2. Chunking the Summarized Text.

Step 3. Segmentation.

Step 4. Normalization.

Step 5. Generating an outline form of summary as follows:

1. Converting the text into structured data.
2. Identifying candidate concepts and relations.
3. Building a structured form of sentence fragments.

Step 6. Concept mapping by converting the generated outline as a whole into a graphical form of a concept map.

Step 7. Query generation by extracting the key-phrases from the generated outline. Figure 1, shows all phases of this algorithm that simplify the following details:

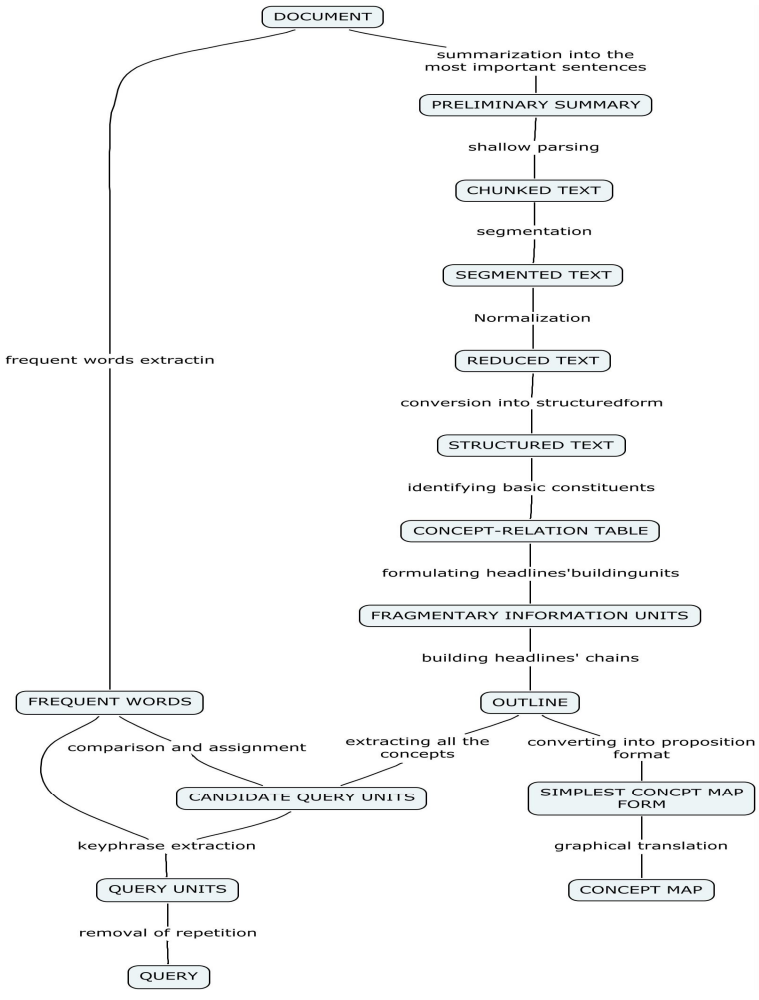

Figure 1: The all phases of the multioptions summarizer's algorithm

ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 01-11 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp01.pdf

A. Summarizing a document into its most important sentences

To condense a document into its main points as a preparation step for automatic generation of a concept map, the input document has to be summarized by a proper text summarization tool to extracts its most important sentences based on frequency keyword approach which first find a set of index terms for the document, and then choose the sentences which contain most keywords.

To introduce the author's view or main points, the most focus is on position-based mode of text summarization that depends on the sentences occur at the beginning as the most important ones, therefore, the whole summary of the original document is taken, afterwards, just first sentences with no more than 20% of the summarization is captured for further analysis and editing.

B. Chunking the summarized text

This step based solely on shallow parsing as a natural language processing for a syntactic analysis of sentences in the summarized text. Chunking is used to perform noun phrase and verb phrase recognition which identify these grammatical elements by assigning a tag for each word in a sentence to indicate whether this word is inside or outside a chunk. for the input of this process, the "NLProcessor by Infogistics" [30] is used to produce linguistic information which directly marking text with XML tags.

C. Segmentation

Splitting the chunked text up into a series of separated segments to construct, if possible, one outline from each segment. This step makes use of punctuation marks as segment separators for decomposing the text into multiple parts.

D. Normalization

Normalization is the process of removing multiple words from the text in order to reduce preliminary the size of a text by defining terms that can be excluded without losing essential textual information. For normalization, unnecessary elements are deleted from the text.

E. Automatic outline generation

As a new technique for summarizing large documents into a list of descriptive and comprehensive headlines by making an analysis of the sentences' syntactic structure, Identifies the key components of headlines, and then reuses these components to create an outline as shown in Figure2. Accordingly, this algorithm automates the process of outlining by three steps as follows:

1. Converting the Text into Structured Data: A prior step to convert unstructured data of natural language sentences into a structured data is building a matrix with three classifier arrays (index, term, type) by taking words directly from the chunked text and at the same order they are found in the text, assigning for each term an index number that point to, classifying each term with a type of either entity, or link category. For entity detection, the definition of a noun phrase is used which is either a single noun or a group of words containing a noun that function together as the subject or object of a verb or a preposition. In link recognition, strings which connects syntactically related words, phrases or clauses together within the text are detected. Therefore, verbs, verb phrases, prepositions, preposition phrases, to infinitives and conjunctions are considered as terms with a category of link.

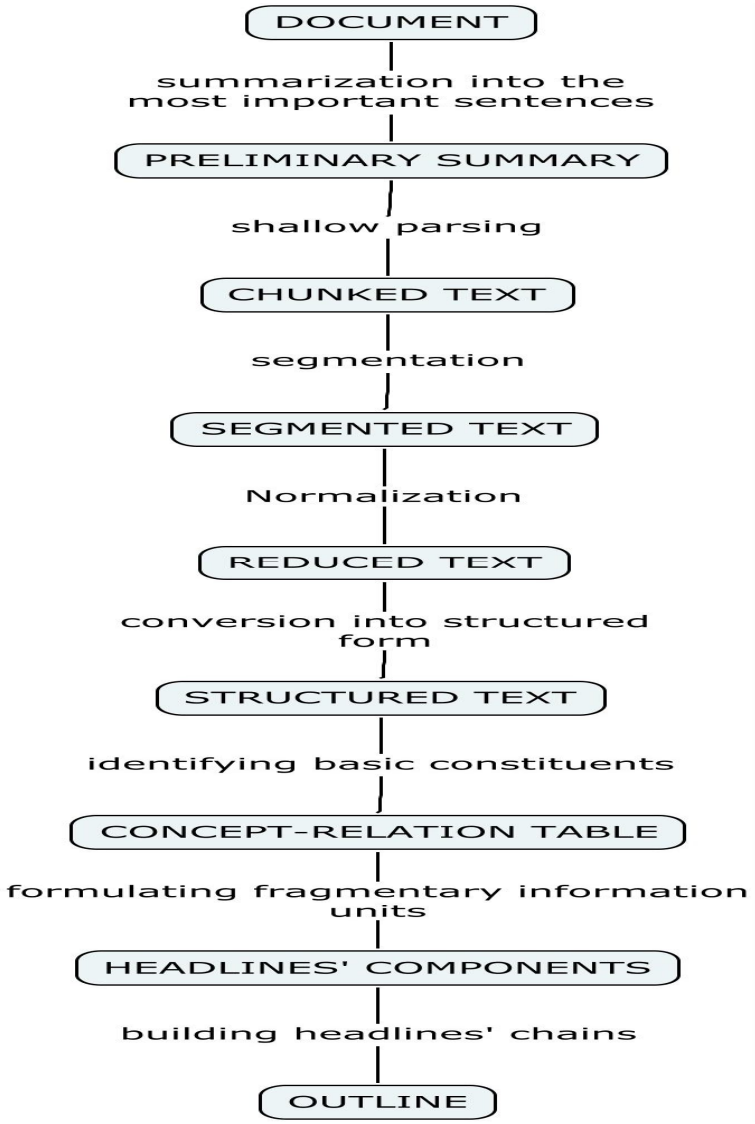

Figure 2: Outlining phase of the multioptions summarizer's algorithm.

ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 01-11 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp01.pdf

2. Identifying Candidate Concepts and Relations: Scanning a structured form of separated terms in type classifier matrix and then operating on them in sequence to build a new classifier for re-chunking the text into cascaded chunks of candidate concepts and candidate relations. From their definitions, concepts are sequences of words that occur together and are used frequently to represent independent entities or ideas, and relations are the connection between these entities. Therefore, in building candidate concepts, all consecutive terms in the matrix with type entity that are mentioned near one another in the text are considered as one candidate concept, and for building candidate relationships, all consecutive terms in the matrix with type link that are mentioned near one another in the text are considered as one candidate relationship for a pair of entities .

3. Formulating Headlines' Building Units: In this process, a set of rules are suggested and applied on each segment separately to convert it into vectors of regular units that consist from fragments shorter than a sentence. Within a particular segment, a forward searching is followed to scan its content from left to right for detecting specific patterns of sentence fragments, and backward deletion as reverse orientation to remove all discarded concepts or relations from this segment, or to delete the whole segment if needed. To build headlines' units vector, the search is for two different patterns in the same segment is done: First-unit pattern, and next-units pattern. First-unit pattern is {C1 R1 C2} which represented by one vector of a pair of candidate concepts and a candidate relationship intervenes between them. To detect the first vector existence, the first candidate concept C1 in a segment is found to begin with, and determine any preceded candidate relation, if found, as discarded relations. Then, finding R1 and C2, where R1 is the next chunk that represents a candidate relation, and C2 is the first succeeding candidate concept in the segment. If the three conditions are satisfied then, remove any discarded relations and consider the triple of (C1, R1, C2) vector as the first unit of a headline , and if not satisfied then, apply backward deletion to delete the whole segment from the text. But If the first vector is found and the current segment didn't finish then complete the process by detecting another pattern for next fragments existence. Next-units pattern is {Rn, Cn+1} which represented by a vector that involved candidate relation and its following candidate concept. Begin with finding, if possible, a new candidate relation Rn then check if it is followed by a candidate concept Cn+1. If the two conditions are satisfied then determine that the vector (Rn, Cn+1) is the next unit of the headline, then reapply this step until reaching the end of the segment. If Rn is found and Cn+1 isn't exist then treat Rn as a discarded relation. Once the forward searching finds the beginning of a new segment the previous process should be all over again to identify new vectors for a set of headline units.

F. Outline extraction

This process aims at producing a set of headlines in order to form an outline form of summary that describes a document by summarizing what it is about. As shown in Figure 3, every headline form is built by simply concatenating the consecutive pieces of fragmentary information units from the output vectors of each segment separately, and aggregating them by their successive order in the text to act as a continues chain of concepts and relations, where a chain is a list of words that form one complete headline, and must satisfy the following four rules:

1. The order of the words is that of their occurrence in the text.
2. Every chain must begin and end with concepts.
3. There must be one relation between every pair of concepts.
4. There is, at least, one (C1– R1– C2) triple in a chain:

C1 – R1 – C2…Rn – Cn+1.

Figure 3: A headline chain with n relations intervene n+1 concepts.

For example, If the analysis of one segment has the following outputs of structured data:

1. The first fragment is: C1 – R1 – C2.
2. The next fragments are: R2 – C3– R3 – C4.

Then the produced headline which represents that segment would be built from structured chain as in Figure 4. Finally, the output set of the separated chains are aggregated in order and formalized in a numbered list to obtain a structured outline of the original document.

C1 – R1 – C2 – R2 – C3 – R3 – C4.

Figure 4: A headline chain with three relations and four concepts

G. Concept mapping

This phase is specialized to generate the simplest form of concept maps by converting the entire outline into a set of propositions as shown in Figure 7. By its functions, current concept mapping software make it possible to import this generated structured list of propositions and translate it into a visual map. Hence, "CmapTools" software (learning software developed by the Institute for Human and Machine Cognition) is exploited to produce a graphical representation of the concept map [2][24]. But, this software needs writing a set of propositions manually and saved them in a separated file to be used as an input text then it translate these propositions into graphical form. Fortunately, this phase of the algorithm could automatically generate a fitting set of propositions from flat data structures to be used as an input text for this software. Therefore, the resulted outline chains are adapted structurally as follows:

1. For each chain, make one repetition of all its concepts except the first and the latest ones, as shown in Figure 5.

ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 01-11 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp01.pdf

2. Build a structured list of propositions by decomposing each modified chain separately into multiple parts using the beginning of the repeated concepts as splitting points, as shown in Figure 6.
3. Merge all the separated headlines in one concept map by connecting the root of every headline with a common relation to the title of the original document through adding n initial triples of { C0 – R0 – C1 } to the structured list of propositions, where n is the number of the resulted propositions, C0 is a concept that includes the title of the document, R0 is a proper relation, and C1 is the beginning concept in a headline.

C1 – R1 – C2– C2 – R2 – C3 – C3 – R3 – C4.

Figure 5: The adapted headline chain after an addition of the repeated concepts C2 and C3.

C1 – R1 – C2
C2 – R2 – C3
C3 – R3 – C4.

Figure 6: The output form of automatic propositions generation

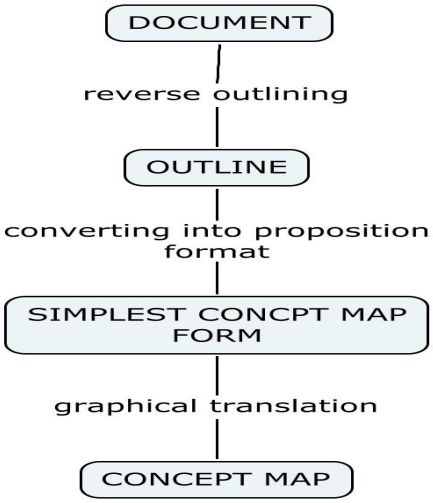

Figure 7: Generating Concept Map Phase

H. Query Generation

Based on the combination of the most frequently words in a document and all the concepts found in the generated outline, the QBD generation is performed as follows:

1. From the generated outline, extract all the concepts that contain two words or more and consider them as candidate query units.
2. Rank the most frequently words by the number of their occurrence in the original document and put them in one list. For this process, current available summarizers such as "tools4noobs" [28] are used to pick only the first thirty frequently words from its outputs.
3. Beginning with the first word, map each word in the previous list to the set of candidate query units and consider the units that contain this word as chosen ones.
4. Detect and delete redundant concepts through mapping process.
5. Aggregate all the resulted chosen query units to form a QBD. Figure 8, summarize this phase.

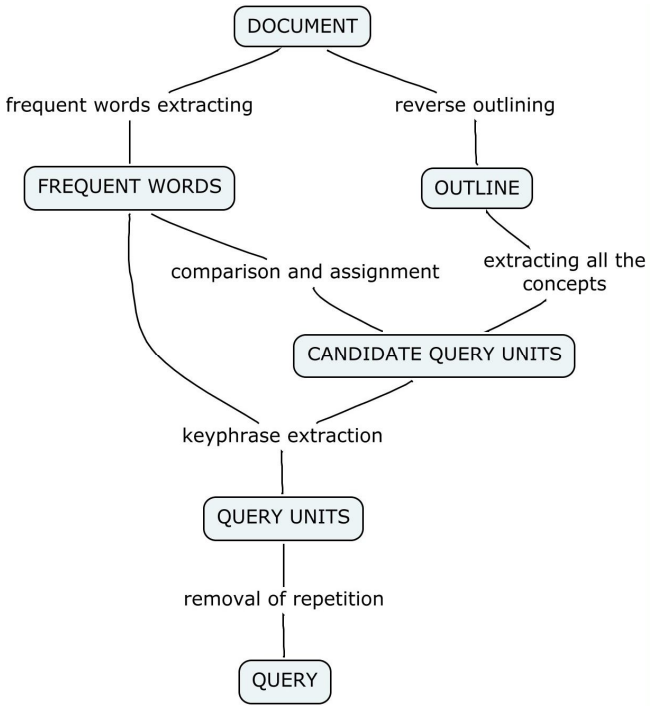

Figure 8: Query generation phase

4. EXPERIMENTS AND EVALUATION

Evaluating a summary or the generated concept map is a difficult task because there does not exist an ideal summary or standard concept map for a given document. Besides, the agreement between human is quite low, both for generating and evaluating concept maps or summaries. The absence of a standard human or automatic evaluation metric makes it very hard to compare deferent systems and establish a baseline [14] [8]. To avoid this problem, a new framework is proposed for concept map and QBD evaluation with some measures that are specially used to explore the extent of the proposed algorithm ability to produce a reasonable concept map and QBD for large document automatically. For this

ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 01-11 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp01.pdf

purpose, the proposed algorithm is transformed into an executable software system and using its automatic outputs for further manually tests. These tests were applied on a collection of 20 documents of academic research papers and a collection of 200 documents for information retrieval purpose. For more illustration, the research paper: "A Ranking Approach to Keyphrase Extraction" [9] was used as an example of the system's input to demonstrate the actual outputs of the algorithm's implementation as three forms of summaries shown by figures 9, 10 and 11.

1-paper addresses issue of extracting keyphrases from document.

2-problem was formalized as classification and learning methods for classification were utilized paper points that is essential to cast keyphrase extraction problem as ranking and employ learning to rank method to perform task.

3-state-of-art method of learning to rank in keyphrase extraction.

4-experiments conducted on datasets show that ranking svm outperforms baseline methods of classification.

5-learning to rank techniques in keyphrase extraction.

6-paper address automatic extraction of keyphrases from document.

7-keyphrases of document mean words and phrases and represent content of document

8-keyphrases are useful for various applications such as document summarization document retrieval document categorization and clustering in digital library keyphrases of scientific paper help users to get rough sense of paper.

9-keyphrase extraction consists of steps.

10-candidate phrase identification and keyphrase selection.

11-problem was formalized as classification in classifier was trained and used to categorize phrases as keyphrases or non-keyphrases.

12-documents as keyphrases assigned by authors or

Figure 9: An example of automatic outlining outputs

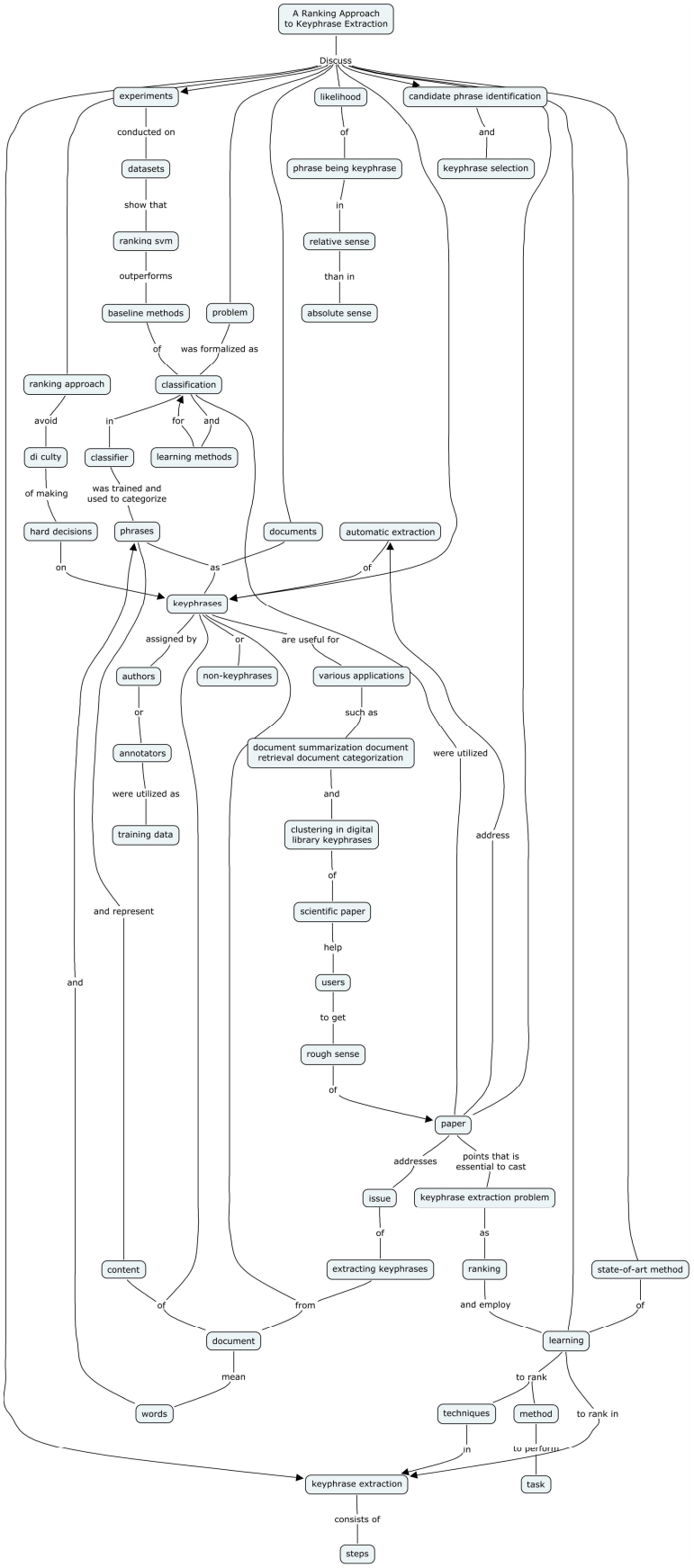

Figure 10: An example of automatic concept mapping outputs

ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 01-11 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp01.pdf

ranking svm ,ranking approach ,keyphrase extraction problem ,keyphrase extraction ,keyphrase selection ,phrase being keyphrase ,extracting keyphrases ,clustering in digital library keyphrases ,automatic extraction ,document summarization document retrieval document categorization ,learning methods ,candidate phrase identification ,baseline methods ,classification approach ,state-of-art method ,scientific paper ,key

Figure 11: An example of automatic query generation outputs

4.1 Concept Mapping Evaluation

For this evaluation, analytically assessement of the generated concept maps is performed using five proposed tests to examine their quality considering both the layout factors of the maps and their content; Subjectivity, Focusing, Accuracy, clearness, and organization.

A. Concept Map Subjectivity:

Test 1: Counting the number of sentences in the abstract of the research paper that have been covered by the generated concept map. This test could be computed by the following proposed formula: Subjectivity degree = the number of abstract's sentences that have been covered by the generated concept map / the total number of the abstract's sentences. Figure maps for 20 research papers.

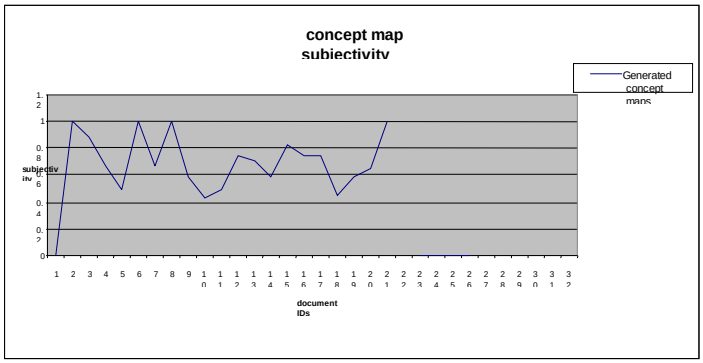

Figure 12: The degree of subjectivity in the generated concept maps

B. Focus Concept Map:

Test 2: Counting the number of the distinct ideas in the generated concept map. Figure 13, shows the focusing degree of the generated concept maps.

This test could be computed by the following proposed formula:

Focus degree = 1 – (the number of the repeated ideas / the total number of the ideas that have been covered by the generated concept map).

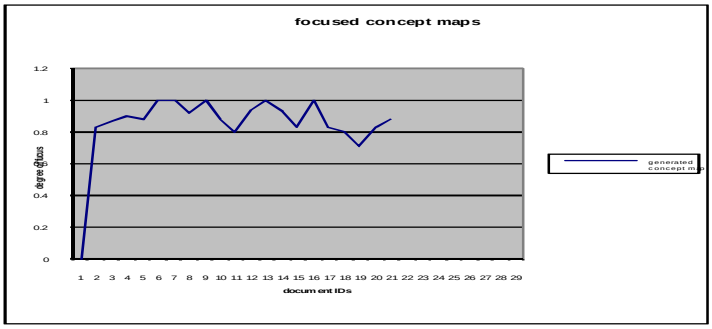

Figure 13: The focusing degree of the generated concept maps

C. Concept Map Accuracy:

Test 3: Estimating, syntactically, the rate of the correct concepts and relations in the generated concept map. The results seen in Figure 14, could be computed by the following roposed formula: Accuracy degree = 1 – {(the number of the syntactically incorrect concept labels + the number of the syntactically incorrect link labels) / the total number of the labels that have been used in generating the concept map}.

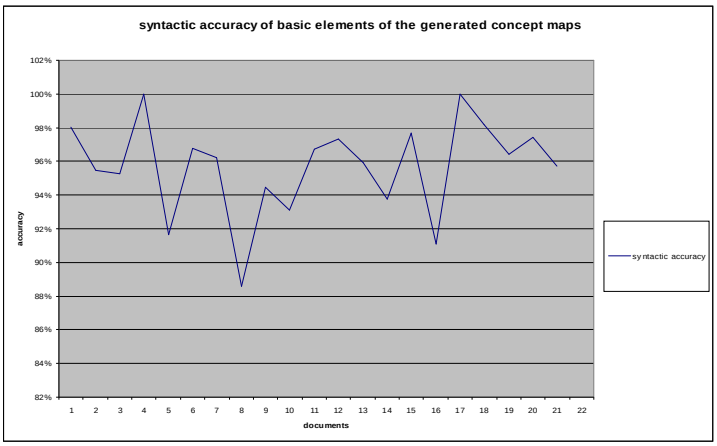

Figure 14: The accuracy of the generated concept maps

ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 01-11 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp01.pdf

D. Concept Map Clearness:

Test 4: Estimating the rate of the clear ideas in the generated concept map. This test could be computed by the following proposed formula:

Clarity degree = 1 – (the number of the ambiguous ideas appear in the generated concept map) / the total number of the main ideas in the generated concept map).

Figure 15, shows the clearness degree of the generated concept maps.

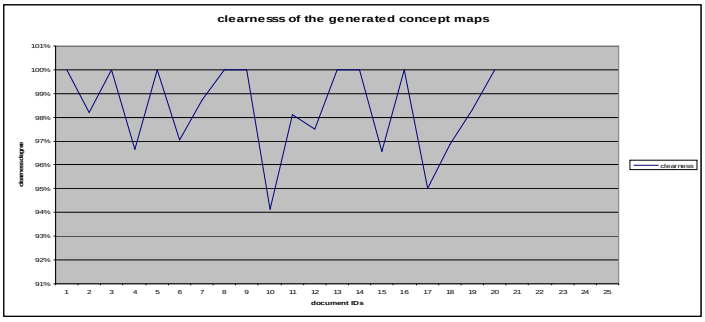

Figure 15: The clearness of the generated concept maps.

E. Concept Map Well-Organization:

Test 5: Finding the well connection rate for the propositions in the generated concept map. This test could be computed by the following proposed formula:

Organization degree = 1 – (the number of errors appear in the connection forms of the generated concept map) / the total number of the connections in the generated concept map).

Figure 16, illustrated the organization degree of the generated concept maps

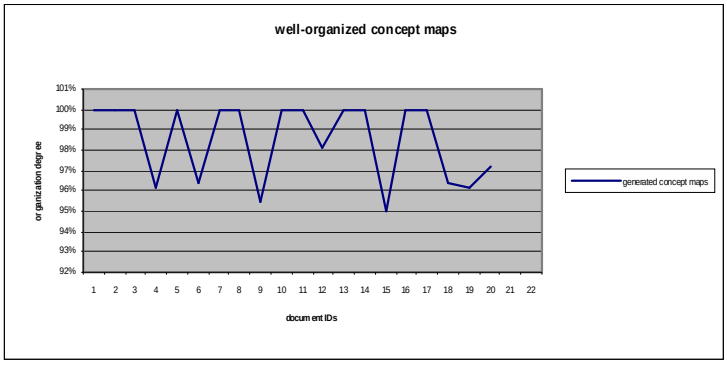

Figure 16: The organization degree of the generated concept maps

4.2 Query Generation Evaluation

For evaluating automatically-generated query by document, the results of key-phrase extraction process are compared between the proposed multioptions summarizer's algorithm and the well known Yahoo Extractor from the perspective of Syntactic Accuracy, and document retrieval. Figure 17, demonstrates a comparison with Yahoo Extractor results:

A. Query Generation Syntactic Accuracy:

Test 6: as away to generate a well-structured query, this test aimed at ensuring the syntactically correctness of the extracted key-phrases.

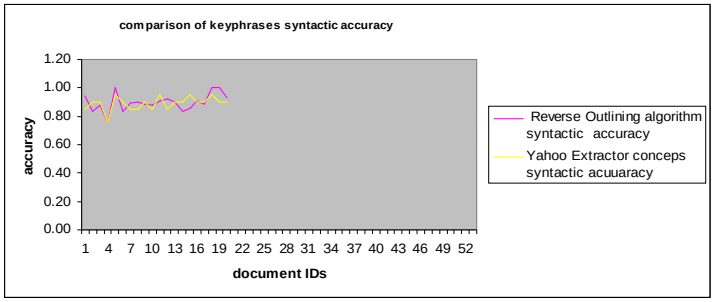

Figure 17: Comparison for the generated queries syntactic accuracy.

B. Query Generation Test for Retrieving Document:

Test 7: In this test every document in the collection is searched by its generated query and its retrieval priority number is checked to be considered as an indicator of the ability of the generated query to represent the original document.

Figure 18, shows a comparison of these results with the Yahoo Extractor results.

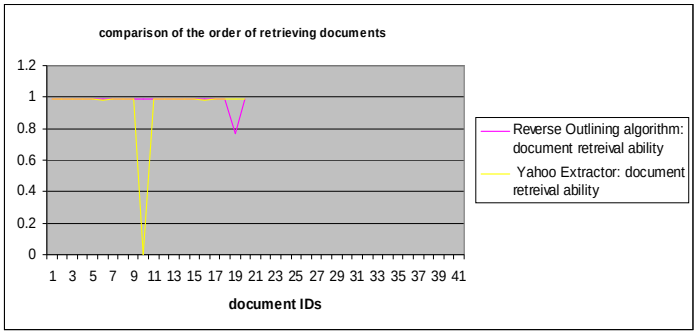

Figure 18: Comparison for the capability of the generated queries to retrieve its original document.

ISSN 2278-3083

International Journal of Science and Applied Information Technology, Vol.7. No.1, Pages : 01-11 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJSAIT/static/pdf/Issue/icsic2017sp01.pdf

C. Evaluation of the information *retrieval effectiveness*

To evaluate the generated queries effectiveness for information retrieval, the experiments was performed with searching by both Yahoo and Google search engines for each QBD in a collection that consists of the generated queries via the developed outlining system.

Test 8: taking the search engine results and checking each one for relevancy (by manually distinguishing them into relevant and irrelevant documents, and then compute the precision for every query to examine the extent to which the retrieved documents are relevant using the following formula:

Precision = ({relevant documents} ∩ {retrieved documents}) / {retrieved documents}.

Figure 19, shows the precision results for the collection of 200 documents retrieved for the generated queries. Obviously, these results provide preliminary support for the suggestion that the proposed multioptions summarizer's algorithm is valid to assist in the process of retrieving the relevant documents from different search engines.

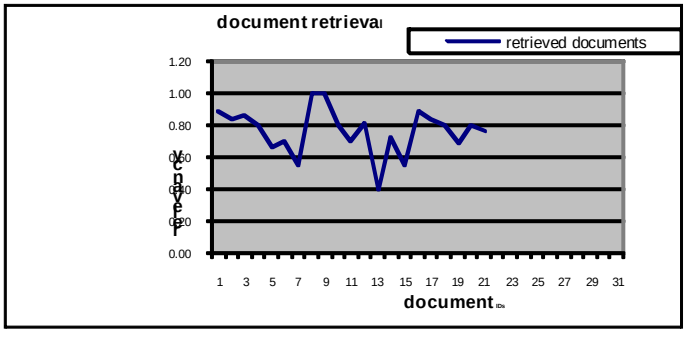

Figure 19: The information retrieval by the generated query.

5. CONCLUSION

To provide a smart summary of a given document with possible reference to the researcher's interests, the proposed multioptions summarizer's algorithm in this paper could provide three summarization options that are designed as a structured outline which is extracted from a document as a list of short notes or in a visual representation by a descriptive concept map that facilitate the understanding of its main topics. In addition, to the third option as a concept-based query from a given document that could act as a helpful tool to search for similar documents. However, since the present algorithm based mainly on a deep analysis of the given text's syntactic structure, it could work better if its input was a well-written text but it is still a current limitation of this study to deal with ill-formed text. Finally, although this algorithm is proved to be acceptable to create multiple forms of summarization for a given document automatically, it is currently applicable only for English language, but fortunately, it has gain some significant advantages that can facilitate future works for converting it to other languages, such as:

1. The ability to derive the dependency information by its own.
2. The ability to build a well structured-text.

REFERENCES

1. T. Reichherzer and D. Leake. **Towards Automatic Support for Augmenting Concept Maps with Document,** in A. J. Cañas, J. D. Novak (Eds.), Concept Maps: Theory, Methodology, Technology,Proc. of the 2nd Int. Conf. on Concept Mapping. San Jose, Costa Rica: Universidad de Costa Rica, 2006.
2. A. J. Cañas, G. Hill, R. Carff, N. Suri, J. Lott and T. Eskridge. **CmapTools: A Knowledge Modeling and Sharing Environment**, in A. J. Cañas, J. D. Novak & F. M. González (Eds.), ConceptMaps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping, Vol. I, pp. 125-133, Pamplona, Spain,2004.
3. C. Yllias, and K. Maheedhar. **Summarization Techniques at DUC 2004**, in Paul Over and J. Yen, editors, Proceedings of the Fourth Document Understanding Conference (DUC '04), Boston, Massachusetts, USA, 2004.
4. R. Richardson and E. A. Fox. **Evaluating Concept Maps as a Cross-Language Knowledge Discovery Tool for NDLTD,** in Proceedings of Electronic Theses and Dissertations Conference, Sydney, 2005.
5. S. Abney. **Parsing by Chunks**, in Robert Berwick, Steven Abney, and Carol Tenny, editors, Principle-Based Parsing, Kluwer Academic Publishers, 1991.
6. V. Alejandro, and D. Leake. **Jump-Starting Concept Map Construction with Knowledge Extracted from**

Documents, in Proceedings of the 2nd International Conference on Concept Mapping, 2006.

7. O. Shiyan, K. Christopher and H. Goh. **Automatic Multi-Document Summarization for Digital Libraries**, Asia-Pacific Conference on Library & Information Education & Practice, 2006.
8. J. Villalon and A. Calvo. **Concept Map Mining: A definition and a framework for its evaluation**, in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology,wi-iat, Vol. 3, pp.357-360, 2008.
9. X . Jiang, Y. Hu and H. Li. **A Ranking Approach to Keyphrase Extraction**, in Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGGIR09, 2009. .
10. V. Qazvinian, D. Radev and A. ¨Ozg¨ur. **Citation Summarization Through Keyphrase Extraction**, in Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 895–903, 2010.
11. D. Maji, and A. **Mondal. Improved Algorithms for Keyword Extraction and Headline Generation from Unstructured Text**, M.Sc. Thesis. Indian Institute of Technology, Kanpur, 2004.
12. T. Yuen-Hsien, L. Chi-Jen and Yu-I. Lin. **Text Mining Techniques for Patent Analysis,** Trends in Computer Aided Innovation IFIP International Federation for Information Processing, Volume 250/2007, 89-96, 2007.
13. E. Lloret. **Topic Detection and Segmentation in Automatic Text Summarization**, December 13, this Work is Protected under the Creative Commons License"Attribution, Www. Dlsi. Ua. Es, 2009.
14. D. Das and A.F. Martins. **A Survey on Automatic Text Summarization**. In Literature Survey for the Language and Statistics II course at CMU, Vol 4, pp. 102-195, 2007.
15. D. Zajic. **Multiple Alternative Sentence Compressions as a Tool for Automatic Summarization Tasks,** Ph.D. dissertation, University of Maryland at College Park College Park, MD, USA , 2007.
16. B. Maria, A. Roxana and M. Francine. **Multidocument Question Answering Text Summarization Using Topic Signatures**, in Proceedings of the Dutch-Belgian Information Retrieval Workshop (DIR'5), Journal on Digital Information Management, 2005.
17. W. Ruichao, Nicola, Stokes, P. William, N. Eamonn, C. Joe and D. John. **Comparing Topiary-Style Approaches to Headline Generation**, in Proceedings of the 27th European Conference on Information Retrieval (ECIR-05), 2000.
18. J. Hongyan, and M. Kathleen. **Cut and Paste Based Text Summarization**, in Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2000), pages 178–185, Seattle, Washington, 2000.
19. M. Sharp. **Text Mining**, Rutegers University, 2001.
20. M. Inderjeet, G. Barbara, and B. Erie. **Improving Summaries by Revising them**, in Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics(A CL '99),pages 558-565, University of Maryland, Maryland, 1999.
21. S. Fuller, R. Debra, P. Bugni, and G. **Martin. A Knowledgebase System to Enhance Scientific Discovery,** Telemakus. Biomedical Digital Libraries, 2004.
22. Simmers, M. (2009). "How to Use Reverse Outlining to Analyze Material". ArticlesBase: Free Online Articles Directory.
23. H. Alam, A. Kumar, M. Nakamura, F. Rahman, Y. Tarnikova, C. Wilcox. **Structured and unstructured document summarization: Design of a commercial summarizer using lexical chains,** in 7th International Conference on Document Analysis and Recognition. Volume 2., Edinburgh, Scotland, UK (2003) 1147–1150, 2003.
24. http://cmap.ihmc.us
25. Y. Yin, B. Nilesh, D. Wisam, I. Panagiotis, K. Nick, and P. Dimitris. **Query by Document.** in *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, WSDM'09, Barcelona, Spain, 2009.
26. L. Zhongguo, and S. Maosong. **Punctuation as Implicit Annotations for Chinese Word Segmentation**, in *Journal Computational Linguistics*, Volume 35 Issue 4, 2009.
27. H. Silber, and K. McCoy. **Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization,** in *Proceedings of Computational Linguistics*, pp 487-496, 2002.
28. http://www.tools4noobs.com
29. **Copernic Summarization Technologies White Paper**, february, 2003. Available commercially online on : http://www.copernic.com/data/pdf/summarization-white paper-eng.pdf
30. http://www.infogistics.com/textanalysis.html.
31. R. Richardson, B. Goertzel and E. A. Fox. **Automatic Creation and Translation of Concept Maps for Computer Science-Related Theses and Dissertations**, in *A. J. Cañas, J. D. Novak, Concept Maps: Theory, Methodology, Technology Proc. of the 2nd Int. Conference on Concept Mapping*, 2006.
32. A. Cañas, C. W. John, C. Mary, P. Feltovich, R. Hoffman, J. Feltovich, J. Novak. **a Summary of Literature Pertaining to the Use of Concept Mapping Techniques and Technologies for Education and Performance Support**, no. technical report submitted to the chief of noval education and training, Pensacola, FL: IHMC, 2003.