Anthony Claret Onwutalobi
University of Helsinki
Department of Computer Science

Using lexical chains for efficient Text summarization

Abstract

The emergence of Internet technology has become a dominant factor in our education, business, and everyday life. In recent years, with the rapid expansion of Internet capability, it has become difficult for end users to efficiently access the enormous amount of information provided for their consumption within a limited time. This problem therefore required an efficient tool to help manage this vast quantity of information. For this reason, any application that has the ability to summarize information automatically and present results to the end user in a compressed, yet complete form; would be a good attempt to the solution of this problem. In this paper, our primary goal is to discuss and present an efficient and effective tool that is able to summarize large documents quickly while preserving its content. We investigate a summarization method which uses not only statistical features but also relative and contextual meaning of documents by using lexical chain which is a method of capturing the "aboutness" of a document. We present a new algorithm to compute lexical chains in a text with robust and economical knowledge resources: the WordNet thesaurus. In this algorithm, summarization proceeds in four steps: the original text is segmented, lexical chains are constructed, strong chains are identified and significant sentences are extracted. We show that our method is efficient and tend to provide quality indicative summaries within a short time. We briefly identify unresolved problems and address future scope and plans of the method.

Keywords: **summarization, lexical, chains, text summarization, cohesion, algorithm**

Introduction

Text summarization is a process of condensing an information source, extract content from it, and present the most important content to the user in a condensed form and in a manner sensitive to the user's or application's need[1]. It can serve many purposes – from analysis of a scientific field to quick indicative notes on the general topic of a text.

Having an indicative summary that quickly and efficiently gives end user an informative overview of whether a text is worth reading remains a challenge. This efficiency is necessary in the Internet search applications were many large documents may need to be summarized at once, and where the response time to the end user is extremely important. In this paper therefore, we investigate a method for the production of such indicative summaries from arbitrary text.

In recent years, several summarization methods have been investigated and reported in the literature. Sparck Jones [2] in his writing summarized this method in two-step process. First step is to extract the important concepts from the source text into some form of intermediate representation. And the second step is to use the intermediate representation to generate a coherent summary of the source document [2].

Within this framework, the relevant question to be asked is what information has to be included in the source representation in order to create a summary. Some scholars have offers different method. Early methods prominent among others were primarily statistical in nature; they focused on title, location of sentence, length of sentence, clue word and word frequency to determine the most important concepts within a document [3]. In a given document, this method collects the frequent words that appear frequently in the document as a topic keywords. The frequent words form the representation that are then abstracted from the source text and make into a frequency table. This method has some limitations as it ignores the semantic content of words and their potential membership in multi-word phrases.

Consequently, another method emerged. This new summarization approach attempts true semantic understanding of the source document by taking an opposite extreme of such statistical approach discussed above. For this method, linguistic approach was used, which tries to understand the contextual meaning of document itself. Apparently the use of such deep semantic analysis offers the best opportunity to create a quality summary because of its expressiveness. But it still has some limitations due to their dependence on the text genre (i.e. domain dependent) which indicates that a domain specific knowledge base must be available and a detailed semantic representation must be created before its operation. This therefore makes it hard to compute.

However, a refined approach was construed, which tries to overcome the limitation of the frequency–based and linguistic method discussed above. Morris and Hirst (henceforth M&H) were the first to introduce these approach which uses the concept of lexical chains [4]. In their work, M&H describe Lexical chains as a system that characterize the lexical cohesion among an arbitrary number of related words, which can be recognized by identifying sets of words that are semantically related (i.e. have sense flow).

By applying this concept, Morris and Hirst aggregated synonym occurrences together as occurrences of the same concept. Using the lexical chains, they realized that they can statistically find the most important concepts of a document by looking at structure in the document rather than deep semantic meaning. After all, the most important component is to have a generic knowledge base that contains nouns, and their associations. With the introduction of this method, H&M realized that using lexical chains in text summarization is efficient, as the contextual relations of words are easily identifiable within the source text, and vast knowledge bases are not necessary for computation.

To further the realization of an efficient method, Bazilay and Elhadad among others noted limitations in the way M&H implemented of lexical chains. They opine that the previous implementation cover only some fragment of the possible sense of words while neglecting potentially pertinent contextual information that appears later in a document. These was referred to as “greedy disambiguation” [5] Bazilay and Elhadad therefore presented a less greedy algorithm that constructs all possible interpretations of the source text using lexical chains by calculating the semantic distance between words using WordNet [5].Their algorithm selects the interpretation with the strongest lexical chains and the sentences related to these strong chains are chosen as a summary. Bazilay and Elhadad used WordNet as their knowledge base. WordNet is a lexical database which captures all sense of a word and contains semantic information about the relations between words [6].

The algorithm first segments text, then for each noun in the segment, for each sense of the noun, it attempts to merge these senses into all of the existing chains in every possible way, hence building every possible interpretation of the segment. Next, the algorithm merges chains between segments that contain a word in the same sense in common. The algorithm then selects the chains denoted as "strong" (more than two standard deviations above the mean) and uses these to generate a summary.

In this paper, we investigate the use of lexical chains as a model of the source text for the purpose of producing a summary. In the rest of the paper we first present the overall design of the system and then present the algorithm for lexical chain construction (section 2) and in the next section describe how lexical chains are used to identify significant sentences within the source text and eventually produce a summary (section 3). Then the next section is devoted to experiment results and evaluation (section 4). Finally, we draw conclusions and present future works.

Overall design of the system

To produce a summary with high quality, we show in Figure 1 the flow chart of a design of lexical chain as we rely on a model of the topic progression in the text derived from lexical chains, these chains are created using semantically related words and the concept represented by the strongest chain is the theme of the text.

WordNet thesaurus is used for this purpose. The main concept in the summarization process proceeds in four steps: first, the algorithm segments the original text to be summarized. Second, it constructs lexical chain using the WordNet. Lexical chains require the use of ontology or a database which has predefined chains of semantically similar words .In using WordNet, the synonyms, and hypernyms/ hyponyms of related words tends to be grouped into the same lexical chain. Third, it identifies the strong chains from the group of word that are semantically related (i.e. have a sense flow). Finally significant sentences are extracted from the strong chain.

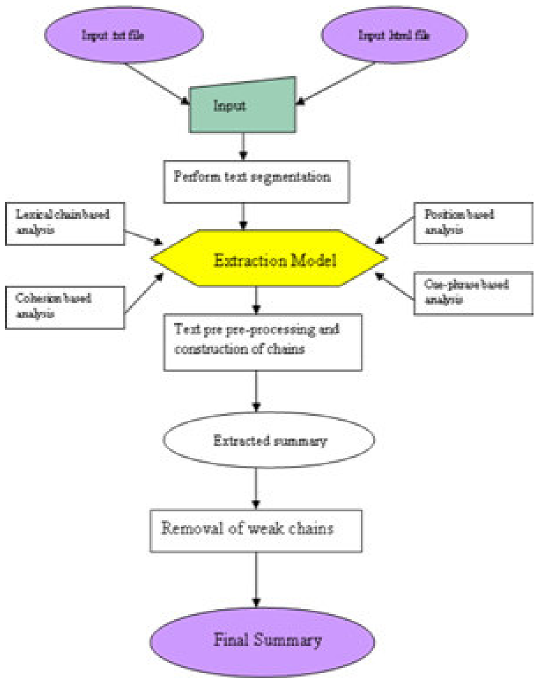

Figure 1

Algorithm for Computing Lexical chain

In recent times, three algorithms for the calculation of lexical chains have been presented in Hirst & Stonge [4], Bazilay and Elhadad [5] and Stairmand [6]. The three algorithms use the WordNet lexical chain database for determining relatedness of the words. In applying WordNet, the senses in the database are usually represented by synonym sets ('synsets') – which are the sets of all the words sharing a common sense.

For example as Bazilay and Elhadad [5] puts it, two senses of a word "computer" may be represented as: {calculator, reckoner, figurer estimator, computer} (i.e., a person who computes) and {computer, data processor, electronic computer, information processing system} [5]. In WordNet Database, there are over 118,000 different word forms [5]. Most words of the same category are linked through semantic relations like synonymy and hyponymy.

In the following lettering, our Algorithm will implement the method of Bazilay and Elhadad, as well as an extended scoring system that we propose in our introduction. However, we will first describe the segmentation process we pointed out in the paper as the first step in constructing lexical chain. The reason for segmenting the document is to allow for comparison. See the algorithm in details here

1. Segment the source document
2. For each noun in the source document, from all possible lexical chains
   a. Look up all relation information including synonyms, hyponyms, hypernyms, this information is stored in array indexed on the index position of the word from WordNet for constant tome retrieval.
3. For each noun in the source document, use the information collected by the previous step to insert the word in each "meta chain". A "Meta chain" is so named, because it represents all possible chains whose beginning word has a given sense number. Meta-chains are stored by sense number. The sense numbers are now zero based due to our re-indexing of WordNet. Again, the implementation details are important as they allow us to retrieve the meta-chain in constant time.

Figure 2

From the figure 2 above, during the segmentation of the source document, the initial phase of the implementation constructs an array of "meta chain." Each Meta -chain contains a score and a data structure which encapsulate the Meta-chain. The score is computed as each word is inserted into the chain. While the implementation creates a flat representation of the source text, all interpretation of the source text is implicit within the structure. [7] After the segmentation process the last three steps are mainly concentrating on constructing and scoring lexical chain.

Generally, a procedure for constructing lexical chains usually follows three steps:

1. Select a set of candidate words.
2. For each candidate word, find an appropriate chain relying on a relatedness criterion among members of the chains;
3. If it is found, insert the word in the chain and update it accordingly. [5]

This procedure has been used by many researchers, the typical example where the procedure has been represented is in Hirst and St-Onge paper (henceforth, H&S). In the preprocessing step, mostly all words that appear as a noun entry in WordNet are chosen. Relatedness of words is usually determined in terms of the distance between their occurrences and the shape of the path connecting them in the WordNet thesaurus [5]. H&S defined three kinds of relations to establish this: extra-strong (between a word and its repetition), strong (between two words connected by a WordNet relation) and medium-strong when the link between the synsets of the words is longer than one (only paths satisfying certain restrictions are accepted as valid connections) [8]. In this relation, the maximum distance between related words depends on the kind of relation: for extra-strong relations, there is not limit in distance, for strong relations, it is limited to a window of seven sentences; and for medium-strong relations, it is within three sentences back. In table below, we demonstrate on how to find a chain in which to insert a given candidate word. In doing this, the extra-strong relations are preferred to strong relations and both of them are preferred to medium-strong relations. If a chain is found, then the candidate word is inserted with the appropriate sense, and the senses of the other words in the receiving chain are updated, so that every word connected to the new word in the chain relates to its selected senses only. If no chain is found, then a new chain is created and the candidate word is inserted with all its possible senses in WordNet.

We will point out the limitation in the greedy disambiguation implemented in this algorithm from the given example

Consider the sentences:

Mr. *Barack is the* ***person*** *that invented an anesthetic* ***machine*** *which uses* ***micro-computers*** *to control the rate at which an anesthetic is pumped into the blood. Such* ***machines*** *are nothing new, but his* ***device*** *uses* ***two micro-computers*** *to achieve much closer monitoring of the* ***pump*** *feeding the anesthetic into the patient.*

Base on H& S's algorithm, the chain for the word "Mr." will be created first [lex "Mr.", sense {mister, Mr.}]. "Mr." belongs only to one synset, so it is disambiguated from the beginning. The word "person" is related to this chain in the sense "a human being" by a medium-strong relation, so the chain now contains two entries:

[Lex "Mr.", sense {mister, Mr.}]

[Lex "person", sense {person, individual, someone, man, mortal, human, soul}]

When you use the algorithm to process the word "*machine*", it relates it to this chain, because "machine" in the first WordNet sense ("*an efficient person*") is a holonym of "person" in the chosen sense. In other words, "*machine*" and "*person*" are related by a strong chain relation. In this case, "machine" is disambiguated in the wrong way, even though after this first occurrence of "machine", there is strong evidence supporting the selection of its more common sense: "micro-computer", "device" and "pump" all point to its correct sense in this context – “any mechanical or electrical device that performs or assist in the performance”.

We notice that this example indicates that disambiguation cannot be a greedy decision. So to choose the right sense of the word, we must consider the ‘*whole picture*’ of chain distribution in the text.

Barzilay and Elhadad propose to develop a chaining model according to all possible alternatives of word senses and then choose the best one among them.

Let us illustrate this method on the above example. First, a node for the word
“Mr.” is created [lex "Mr.", sense {mister, Mr.}]. The next candidate word is “person”. It has two senses: “human being” (person – 1) and “grammatical category of pronouns and verb forms” (person – 2). The choice of sense for “person” splits the chain world into two different interpretations as shown in Figure 3.

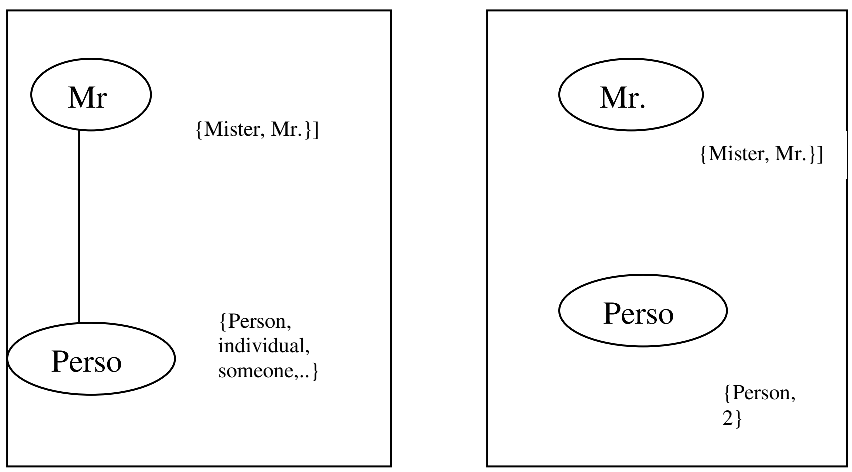

Figure 3: step 1, Interpretation 1 and 2

From this example, we created a list of interpretations that are exclusive of each other. In this list, words can influence each other in selection of their respective sense. So if there are words that are related with which other, they tend to group into a chain. The next candidate word “anesthetic” is not related to any word in the first component, so we create a new component for it with a single interpretation. The word “machine” has 5 senses machine1 to machine5. In its
first sense, “an efficient person”, it is related to the senses “person” and “Mr.” It therefore influences the selection of their senses, thus “machine” has to be inserted in the first component. After its insertion the picture of the first component becomes the one shown in Figures 4 to 7.

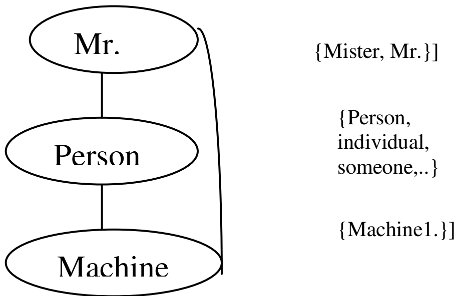

Figure 4: step 2, Interpretation 1

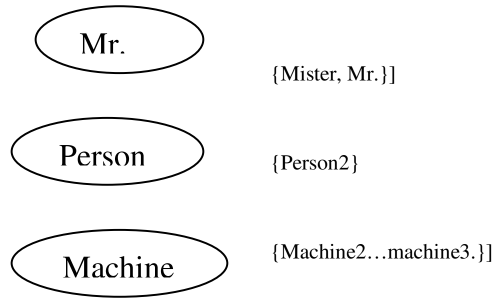

Figure 5. Step 2 , Interpretation 2

As we continue the process and insert the words "micro-computer", "device" and "pump", the number of alternatives will greatly increases. From this diagram, the strongest interpretations can easily be identified. In the remaining diagram, the possible strongest interpretation can be seen in figure 8 and 9

With the assumption that the text is cohesive, E&B define the best interpretation as the one with the most connections (edges in the graph). In this case, the second interpretation at the end of Step 3 is selected, which predicts the right sense for "machine". In the end, the definition of the score of an interpretation is base on the sum of its chain scores [5].

A chain score is determined by the number and weight of the relations between chain members. Experimentally, we fixed the weight of reiteration and synonym to 10, of antonym to 7, and of hyperonym and holonym to 4. Bazilay and Elhadad [5] algorithm computes all possible interpretation maintaining each one without self contradiction. When the number of possible interpretations is larger than a certain threshold, we prune the weak interpretations according to this criteria, this is to prevent exponential growth of memory usage. In the end, we select from each component the strongest interpretation.

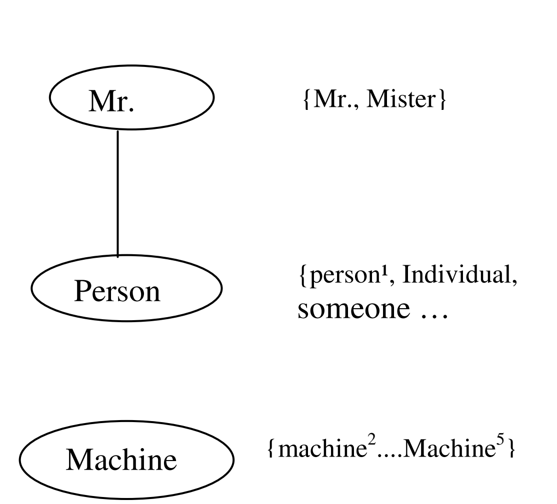

Figure 6: Step 2, Interpretation 3

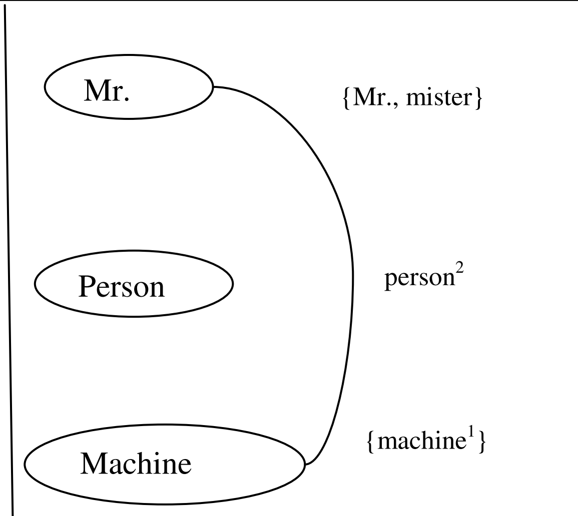

Figure 7: step 2, Interpretation 4

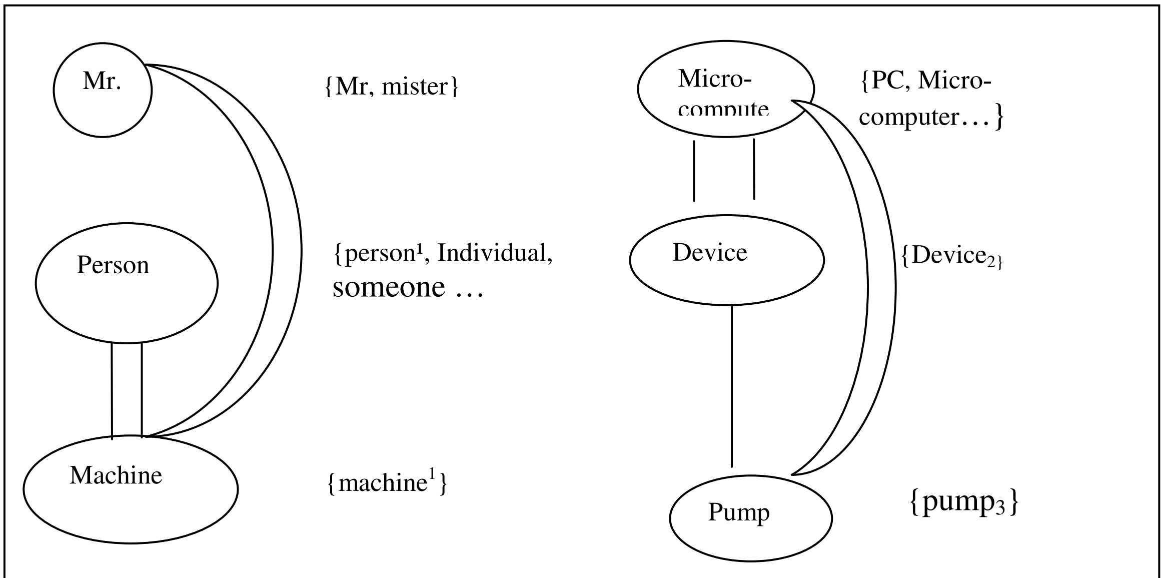

Figure 8" step3, Interpretation 1

Anthony Claret Onwutalobi
University of Helsinki
Department of Computer Science

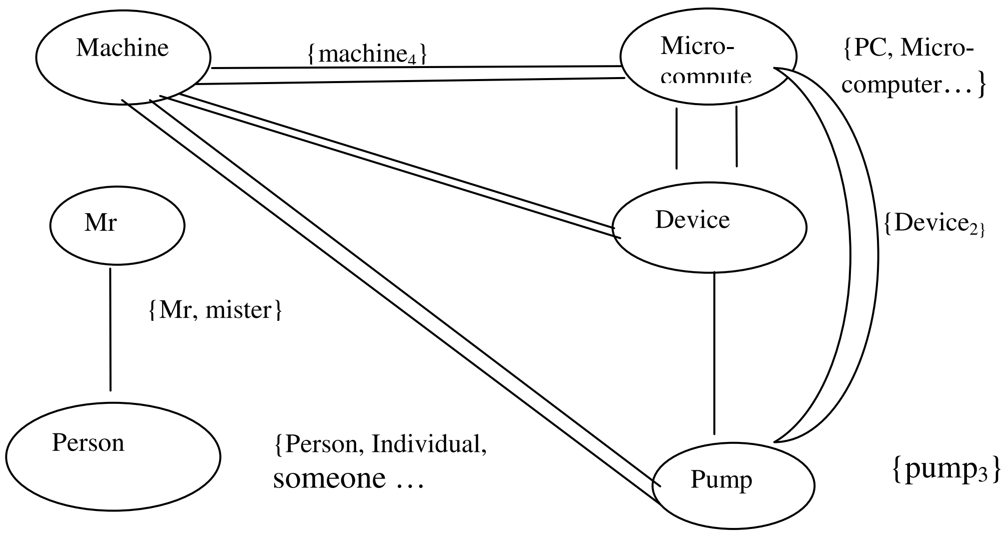

Figure 9” step3, Interpretation 2

In summary, what the algorithm help us to achieve is enabling us choose as candidate words simple nouns and noun compounds. Note that Nouns are the main contributors to the “aboutness” of a text, and noun synsets dominate in WordNet. In addition, we allow us extend the set of candidate words to include noun compounds and we also use as text units the segments obtained from Hearst’s algorithm of text segmentation [5]. We build chains in every segment according to relatedness criteria, and in a second stage, we merge chains from the different segments using much stronger criteria for connectedness only: two chains are merged across a segment boundary only if they contain a common word with the same sense.

IDENTIFYING STRONG LEXICAL CHAINS

1. Compute the aggregate score of each chain by summing the scores of each individual element in the chain.
2. Pick up the chains whose score is more than the mean of the scores for every chain computed in the document.
3. For each of the strong chains, identify representative words, whose contribution to the chain is maximum.
4. Choose the sentence that contains the first appearance of a representative chain member in the text.

Anthony Claret Onwutalobi
University of Helsinki
Department of Computer Science

Building summaries using lexical chains

We now investigate how lexical chains can serve as a source representation of the original text to build a summary. The next question is how to build a summary representation from this source representation. The most prevalent discourse topic will play an important role in the summary. We first present the intuition why lexical chains are a good indicator of the central topic of a text. Given an appropriate measure of strength, we show that picking the concepts represented by strong lexical chains gives a better indication of the central topic of a text than simply picking the most frequent words in the text (which forms the zero-hypothesis).

For example, let's consider this sentence,

God is the creator of the universe; He created man and gave him authority to subdue the earth. These creator cautioned man not to disobey him. Yet man could not contain himself and disobey God. Since man could not manage the universe, He decides to withdraw the authority from man and handed it over to Lucifer (Satan). But He did not feel good about this decision and decided to forgive man by sending Jesus Christ to come and redeem man.

We show in this paragraph above a sample text about God creation. Here, the concept of God is denoted by the words "god" with 6 occurrences, "he" with 2, and "creator" with 2. But the summary representation has to reflect that all these words represent the same concept. Otherwise, the summary generation stage would extract information separately for each term. The chain representation approach avoids completely this problem, because all these terms occur in the same chain, which reflects that they represent the same concept.

Scoring chains

In order to use lexical chains as outlined above, one must first identify the strongest chains among all those that are produced by the algorithm. As is frequent in summarization, there is no formal way to evaluate chain strength (as there is no formal method to evaluate the quality of a summary).

The table depicts the lexical chains construction and scoring.

	Sense Index	Sense meaning	Element 1	Element 2	Element 3
Chain 1	0	Mister	{mr,1}		
	1	person	{john,1}	{Machine, 0.5}	
	2	unit	{Computer,1}		
	3	device	{Computer,1}	{Machine, 1}	
	4	organization	{Machine,0.5}	{Unit 0.5}	
Chain N	N-1				

Scoring scheme
Identical word =1
Synonym =1
Hypernym/hyponym =0.5
From the above example, we must recognize that for us to choose the right sense of the word the 'whole picture' of chain distribution in the text must be considered. We propose to develop a chaining model according all possible alternatives of word sense and then choose the best one among them.

Analysis of our algorithm

The experiments were conducted with the intention of determining how well our algorithm duplicates the experimental results of Barzilay and Elhadad. In conducting such an analysis, we must consider the known differences in our algorithms. The first, and possible most apparent difference in our algorithms, is in the detection of noun phrase collocations. The algorithm presented by Barzilay and Elhadad uses a shallow grammar parser to detect such collocations in the source text prior to processing [5]. Our algorithm simply uses word compounds appearing in WordNet (WordNet stores such words connected by an underscore character). This difference may account for some of the difference observers in the results.

Another inherent difference between the algorithms is that Barzilay and Elhadad attempt to proper nouns which our algorithm does not address. Although not clear how it is done, Barzilay and Elhadad do some processing to determine relations between proper nouns, and their semantic meanings.

Upon analysis, these differences seem to account for most of the differences between the result of our algorithm with segmentation, and the algorithm of Barzilay and Elhadad

Limitation and Future scope

As this is ongoing research, there are many aspects of our work that have yet to be addressed. Issues regarding the extraction of lexical chains, segmentation, scoring, and eventual generation of the summary text must be examined further. Segmentation, as implemented by Barzilay and Elhadad, is inefficient. It may be possible to incorporate segmentation information by making the distance metric of our new scoring system dynamic. By using segmentation information to determine the distance metric, we may be able to take advantage of segmentation without the expense of merging together chains computed from individual segments [5].

Examination of the performance of our algorithm on larger documents should be conducted. Moreover, further analysis on the effects of pruning, as required by Barzilay and Elhadad, on these larger documents are also warranted

The scoring system proposed in this research requires optimization. Currently, its values are set based on the linguistic intuition of the authors. In future works, we hope to use machine learning techniques to train these values from human-created summaries.

Lastly, experiment to evaluate the effectiveness of a summary must be conducted. These experiments are necessary to examine how well our summary can assist a user in making a decision or performing a task. Since no two people would summarize the same documents in precisely the same way, evaluation is one of the most difficult parts of text summarization

References:

[1] I. Mani, Automatic summarization, 2001. John Benjamins publishing Co.:1-22

[2] Jones, Karen Sparck. What might be in summary? Information retrieval, 1993

[3] Luhn, H.P. The automatic creation of literature abstracts. In H.P. Luhn: Pioneer of information science Schultz, editor. Spartan, 1968

[4] Morris, J. and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of the text. In computational Linguistics, 18(1):pp21-45. 1991.

[5] Barzilay, Regina and Michael Elhadad Using Lexical chains for Text summarization in proceedings of the intelligent Scalable Text summarization workshop (ISTS'97), ACL Madrid, 1997.

[6] Stairmand, M. 1996. A computational Analysis of Lexical Cohesion with Applications in Information Retrieval. PH.D Dissertation, Center for computational Linguistics UMIST, Manchester.

[7] H.Gregory Silber and Katheleen F. McCoy Efficient Text Summarization Using Lexical Chains dated 22/11/08. Source: http://web.media.mit.edu/~lieber/IUI/Silber/Silber.pdf

[8]Hirst, G., and St-Onge, D. 1998. Lexical chains representation of context for the detection and correction of malapropisms. In Fellbaum, C., ed., WordNet: An electronic lexical database and some its Applications. Cambridge, MA: The MIT Press.