A Novel Approach Towards Automatic Text Summarization Using Lexical Chains

Sulabh Katiyar
P.G. Student, Department of CSE
NIT, Silchar
Silchar, India
sulabhkatiyar@gmail.com

Samir Kr Borgohain
Assistant Professor, Department of CSE
NIT, Silchar
Silchar, India
samirborg@gmail.com

***Abstract*—** Text summarization is a process of extracting text by virtue of reduction of document contents while preserving the salient information intact. By using different set of parameters like position, format and type of sentences in an input text, frequency of words in a text etc., techniques have been developed. But the parameters vary depending on source of input texts. This in turn affects the performance of the algorithms. In this paper, we present a new method of automatic text summarization by making use of lexical cohesion in the text. Until now lexical chains have been used to model lexical cohesion. These lexical chains are sequences of words having semantic relations between them. In our proposed algorithm, we have used a modification of lexical chains to model the relationships that exist between words.

Keywords-*text summarization;lexical cohesion; lexical chains*

*****

I. INTRODUCTION

Summary is the reductive transformation of a document's content by condensing a text into its shorter version while preserving the salient portions of its information. Summaries can be of two types: extract and abstract, depending on the source of the content. If a summary contains important sentences taken from a original text, it is called as an extract. Abstracts are formed from paraphrased or generated sentences. Building abstracts is difficult and requires greater understanding of the text. Ideally, a summarization system should perform full understanding, which is very difficult and only domain dependent solutions are currently available.

The automatic text summarization consists of two steps [1]:

- Constructing an intermediate source representation from the source text
- Summary generation: Building of a summary from the source representation.

A number of techniques exist for building intermediate source representation. Statistical approaches have been used in which summaries are built from a shallow linguistic analysis of the text such as word frequencies, location of text in document and cue words [2], [3], [4], [5] and [6]. The methods proposed by [7] and [8] have used machine learning in order to combine several shallow heuristics (cue phrase, length of sentences, location, title, word frequency etc.) from a corpus of research papers with abstracts produced manually. Mani [2] pointed out that in these approaches text units are not distinguished at all according to the semantic information they represent. Moreover their utility varies greatly between text genres as the number of formal cues and markers changes critically from, say scientific and research articles to opinion pieces to monologues.

Another way of representing linguistic structure is by means of coherence. The coherence is used in linguistics to define the semantic integrity of a text. For a connected discourse, coherence represents its macro-level semantic structure in terms of relations such as elaboration, cause and explanation between segments of the text. Some of the researchers have tried to use coherence as a basis for summarization [9] and [10]. However, without a comprehensive understanding of the text and its complex inferences, coherence is very difficult to be identified and complex to be implemented.

Although a complete understanding of the text is the long term goal of text summarization [11], researchers have looked into other low-cost measures for capturing semantic structures of the documents. Cohesion introduced by [12] is simpler than coherence and it can also help partly to determine the discourse structure in the text. Cohesion is a surface level feature. Examples of relations that underline cohesion are lexical cohesion (use of related terms), co-reference, ellipsis and conjunction. Lexical cohesion is easier to identify than co-reference, ellipsis and conjunction. Moreover, there is a close relationship between cohesion and discourse structure. The related words generally co-occur within a discourse unit in a document. Cohesion can be modeled with lexical chains. Lexical chains group semantically related words which are spread across sentences in the text into meaningful sequences that represent the cohesive structure of the text.

II. SURVEY ON LEXICAL CHAINS

The lexical chains were first reported by Morris and Hirst [13]. The authors have used the Roget's Thesaurus [14] to compute the word relationships and found out the chains. The Roget's Thesaurus is organized hierarchically and has eight major classes at the top of the hierarchy. Each class has subclasses, which, in turn are divided into sub-subclass. The category level is at the bottom. There are 1042 basic categories. Each category has a list of paragraphs that contain related words from a syntactic category. In each paragraph even finer groups are separated by semicolons. A semicolon group also has pointers to related categories. For retrieving words related to a given one, thesaurus contains an index which includes a list of words with similar sub-senses. For constructing lexical chains the authors took nouns, adjectives and verbs from the text as candidate words.

Five types of thesaural relations, with different scores depending on importance were chosen to categorize the word relationships. Once the lexical chains were computed strength of these chains were calculated based on repetition of contained words, density with respect to the positions of the sentences these words come from, and finally, length of the chain itself.

Morris and Hirst could not implement their algorithm, because of want of a machine- readable version of Rogets Thesaurus. An important drawback of their work is that words may appear in different senses in various chains they are contained in which leads to ambiguity in their determined relationships.

Hirst and St-Onge [15] implemented the Morris and Hirst algorithm when the electronic thesaurus Wordnet became available. They used it to detect and correct malapropisms in the text. Hirst and St-Onge took nouns as candidate words since they represent the 'aboutness' of the document. The verbs are not connected to nouns in WordNet hierarchy and most adjectives have convertible forms to nouns. The word relationships are defined in terms of the distance between their positions and the shape of the path connecting them in the WordNet hierarchy according to the relations between them. Eight patterns of paths are allowed between words and three types of relations are designated: extra-strong (between the repetition of a word and itself), strong (between words that are connected by any WordNet relation) and medium-strong (when the length of path between their synsets is greater than one). The allowable distance between related words, in terms of the span of sentences they occur in, is : for extra-strong relations, no limit in distance, for strong relations, limit of a window of seven sentences; and for medium strong relations, it is within three sentences back. For inserting a word into a chain, extra-strong relations, strong relations and medium strong relations are given the preference. If a chain is available, then the candidate word is inserted into it with its appropriate sense, and senses of words which are already present in the receiving chain are updated. This way sense of the word is gradually disambiguated. If no chain is found, then a new chain is created and the candidate word is inserted with all its possible senses in WordNet.

One of the limitations of Hirst and St-Onge work is that they disambiguate words in a greedy manner [16]. The word sense is determined as soon as the word is encountered. This means that its sense is determined only with respect to words yet seen in the text. Since the sense of a word which is used most in the text is the correct one, the word senses should be determined according to their relationships with all the words in the text. Otherwise, change in the order of sentences or paragraphs in the text would lead to different senses selected and different chains created [17].

Barzilay and Elhadad [16] propose a non greedy strategy in which there are as many interpretations as there are senses for a word. An interpretation is a group of chains built under the assumption that the chosen word sense is the correct one. A list of interpretations is kept and subsequently trimmed down by choosing the interpretations with highest scores in order to reduce the complexity. The scores are calculated by summing weights of links: 10 for reiteration and synonymy, 8 for either being the offspring, 7 for antonymy, 4 for meronymy and 2 for being siblings. Barzilay and Elhadad also have used Texttilling Segmentation Algorithm [15] to partition the text into portions which contain similar topics throughout. Later the chains from two or more segments are merged if they both contain a word with the same sense.

The nouns and noun compounds are used as candidate words. The scores of the chains depend on their length and number of distinct words therein. The strong chains are those which have scores above the threshold and is defined as the sum of average chain score and twice the standard deviation. For each of these chains, sentence that contains the first appearance of the representative chain member is selected. The representative chain members are those having frequency in the chain greater than the average.

One of the limitations of their strategy is exponential complexity [18]. Moreover, since they do not store all possible interpretations until all the words have been encountered [16], this method does not perform an exhaustive non-greedy word sense disambiguation.

Silber and McCoy [19] have presented a linear time algorithm for computing lexical chains and an evaluation mechanism to verify their suitability for text summarization. Unlike Barzilay and Elhadad, rather than trying to computing multiple interpretation for each word in the document, Silber and McCoy store the interpretations implicitly in a structure without actually creating them. This keeps the program linear in both space and time.

The noun database was recompiled into a binary format and memory mapped. In the first pass of the algorithm, an array of 'meta-chains' was created, whose indices represented all the noun senses in the WordNet. For each noun found in the document, its senses are retrieved from the WordNet and their relationships are computed with all the meta-chains (i.e. their representative noun senses). In addition to that, a noun instance maybe included in a chain if it is related to some word that is already in the chain. The noun is placed into all the meta chains where it has identity, synonymy, or hyperonymy relation. These meta-chains represent every possible interpretation of the text.

For finding the best interpretation, second pass is called to determine which meta-chain the noun contributes to most. The contribution of the word is calculated according to its relation with other words in the meta-chain and distance between them in terms of span between the containing sentences. For each noun instance, if it is first word to be inserted into the chain, then identical Word relation score is given. If not, then the closest noun in the chain to which it is related is determined and score is given depending on the type of relation. In case of a tie, the higher sense number is used because in WordNet more specific concepts are indexed with higher numbers. The noun is subsequently deleted from all other meta-chains. Once all nouns have been deleted, scores of the resulting chains are computed and strong chains are selected similar to Barzilay & Elhadad.

The algorithm runs in linear time since the size of Wordnet is constant and the preprocessing step which includes Part-of-Speech Tagging is typically fast.

The algorithm has inaccuracies in the word sense disambiguation [18]. In our findings, we have observed that the method used for chain computation has some limitations. When a word sense is to be added to a meta-chain, if it is the first member to be added in it, it is given the score of identity relation when its contribution to the chain is being calculated, even if it is connected to the sense represented by the meta-chain by sibling or hyperonymy relation.

Brunn, Chali and Pinchak [20] explore a lexical chain building process in which nouns occurring in the sub-ordinate clause of the sentences are filtered out with the assumption that rather than adding information, the create 'noise' in the text. The remaining nouns are the candidate words. Each of their senses is 'exploded' into 'levels' of senses. It means the first level comprises of synonyms and antonyms, the second level is made up of the set of first hypernyms/hyponyms and meronyms/holonyms, etc and so on. Relationship between two word senses is said to exist if there is a non-empty intersection between their sets of levels and the score depends on the length of path taken in matching the two senses. Then the score of chains is calculated which depends on their length and preference of the relationships with respect to levels, i.e., word repetition, synonym/antonym, level-1, level-2, and so on. For larger texts, the number of relationships may be very large and thus permitted relationships are reduced to antonyms/synonyms only. The word sense disambiguation is implicit as it is assumed that only some of the chains will be retained in which senses of a word occur.

Once the chains are selected, segments are scored based on the number of occurrences of words in the segment which belong to some chain, normalized by their occurrences in other segments. This leads to higher contribution of chain members in the score of a particular segment which occur more in that segment.

The critical limitation is the assumption that during selection of chains based on their length and relationships, only the chains which have highest probable senses of a particular word will be retained as its lesser important senses will be grouped into smaller chains. In reality, the length of the chains doesn't vary much. Hence there is not much of a difference between lengths of the chains to which different senses of a word belong to. Moreover storing multiple levels of related senses for a particular word sense is done only for small size texts. Permissible relations being limited to synonyms/antonyms greatly reduce the efficacy of the method in finding relationships for larger texts.

Olena Medelyan [21] proposed a new method of computing lexical chains by graph clustering. The word relationships are modeled by a graph in which the nodes represent the words and the edges represent the relationships between them. For each new candidate word encountered in the chain, all the chains are searched for words with which it can have a relationship and if two or more chains are found they are merged to create a single chain. No explicit word sense disambiguation is done at this point. Once all words have been added to the set of graphs, strength of the chains is calculated as a function of graph diameter. The graph distance between two nodes is defined as the minimum length of the path connecting them and graph diameter of the chain is the maximum of the graph distances between all pairs of nodes. Then the strongly cohesive chains are defined as fully connected graphs where diameter is equal to 1. Weakly cohesive ones connect terms without cycles and the diameter is one less than the number of vertices in the graph. Moderately cohesive are the chains with diameter between these two extremes. Then a graph clustering algorithm- Chinese Whisper [22] is used to find strongly connected clusters and the graph is decomposed into strong chains. The sentence extraction follows this step in which sentences scores to be used for extraction as calculated as the sum of scores of words they contain. The word scores depend on the strength of the chains they are contained in which is calculated in the same way as [16].

While building lexical graphs, no distinction is made between the different senses of the word and is connected to all the words in the graph. It is assumed that when chains will be decomposed the weak chains representing the incorrect senses will be eliminated. But in practice, strong chains contain very few words because it is very difficult to have all the words connected to each other. In case of [16], the disambiguation was performed by looking at the total number of connections in the interpretations since the correct word senses will have relationships with the chains comprised of a lot of words from the text. It is not possible in this case as the strong chains are composed only of very closely related words and it will be difficult to distinguish between two senses of a word if they both belong to very small chains of two or three words. Moreover different types of relations between words are not distinguished in the chain.

III. OUR LEXICAL CHAIN ALGORITHM

We have proposed a method based on lexical chaining where lexical relationships are represented in the form of a graph G=(V,E) where vertices v_i Є V are either the words or word senses. The edges (v_i, v_j, w_{ij}) Є E are the relationships between the vertices having weights which represent the strength of relationships. We have used Brill's part-of-speech tagger [23] to extract nouns from the texts.

The steps involved in the construction of the graph G is listed below.

1. Add all the words as vertices to the graph.
2. Compute the pair-wise relationships among all the word vertices in the graph and draw undirected edges. The weight of an edge depends on the strength of the relationship between two word vertices. The strength of the relationship between two word vertices is measured by their distance in the WordNet taxonomy. For word vertices v_i, v_j Є words and their weight are represented in the form of tuple i.e. (v_i , v_j , *weight*) and are stored in a list called SCORE.
3. For each word vertex v_i in graph G, $1 \le i \le n$, repeat steps 3.a to 3.d
    a. Find all the other word vertices which are connected to v_i through the edges in the graph and store those word vertices in list W_i where $1 \le i \le n$ where n is the number of nouns in the text.
    b. Compute the pair-wise relationships between the words present in the list W_i.
    c. For all the words in W_i, find their relationships with all other words present in the list W_i and add the weights of the relationships. This sum is called the "sum-score" for that word with respect to the word w_i.

d. Update the weights stored in the list SCORES by adding the "sum-score" for each word w_j in the list W_i to the 'score' in the tuple (w_i, w_j,score).

4. At the end we have a graph in which words are connected to each other based on the strength of relationships and the popularity of words in the document. The graph is then decomposed into individual chains for use in sentence extraction step. The decomposition is done such that the graph is broken down into disjoint sets of vertices, i.e. individual chains. These chains are used in sentence extraction process, as described later section.

We illustrate with an example to describe the construction of lexical chains from a graph. Consider a graph G containing eight vertices and the edges drawn between the vertices as shown in Figure 1. The vertices are numbers as W_1...W_8. The graph G is constructed by applying steps 1 and 2 of the algorithm.

Apply step 3.a of the algorithm. Let W_2 be the chosen vertex from the given graph. The vertex W_2 has undirected edges to vertices W_3, W_5, W_7 and W_9 respectively. Let the weight between vertices (W_2, W_3) be 'a', (W_2, W_5) be 'b', (W_2 W_7) be 'c' and (W_2, W_9) be 'd'. Figure 2 represent the graph showing the edge connections from W_2 to W_3, W_5 W_7 and W_9. The other edges are not shown in the figure for better understanding and clarity.

The step 3.b of the algorithm calculates the pair-wise relationships between the vertex pairs (W_3, W_5), (W_5, W_7), (W_7, W_8) and (W_5, W_8) from the given graph is shown in Figure 3.

Apply Step 3.c of the algorithm. Add the weights of the relationships of words W_5, W_7 and W_8 depending on the relationships between word vertices. So, the sum-score of W_5 is 'h', W_7 is 'e+h+g' and W_8 is 'e+f'.

Apply Step 3.d of the algorithm. Revise the relationships between W_2 and other word vertices that are related to it. The new weight of vertex pairs (W_2, W_3) is 'a+h', (W_2, W_5) is 'b+e+h+g', (W_2 W_7) is 'c+f+g' and (W_2, W_8) is 'd+e+f'. This ensures that W_2 has stronger relationships with those words that are more popular in the text, thereby reinforcing the accurate sense of the word in the present context. Figure 4 represents the result of step 3.c and 3.d of the algorithm.

The above steps are repeated for each vertices of the graph for calculating the weights.

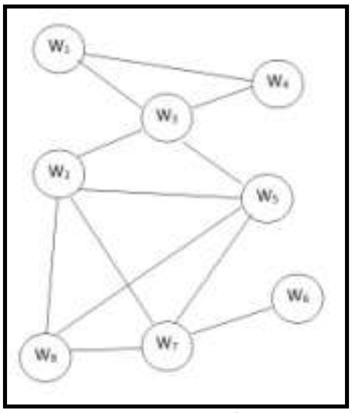

Figure 1. Graph G

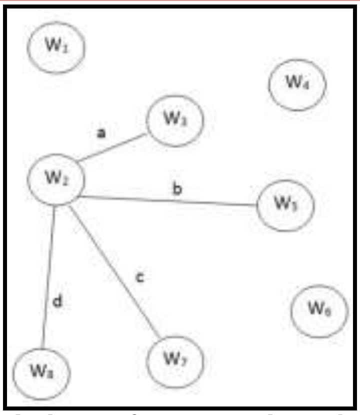

Figure 2. Calculation of pair-wise relationships between word vertices.

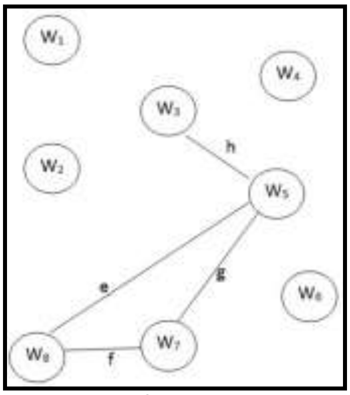

Figure 3. Edge weights of word vertices W_3, W_5, W_7 and W_8

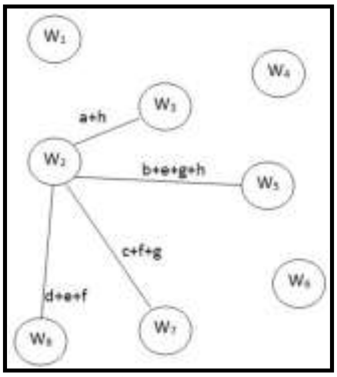

Figure 4. Calculation of pair-wise relationships between word vertices

Sentence extraction: Once the graph has been constructed, we compute the lexical chains from the graph. The lexical chains are constructed by decomposing the graph into longest sub-chains such that they contain different words, i.e. decomposing the graph into disjoint set of vertices.

Now for sentence extraction, we need to find the strongest chains. The strongest chains are found by the same metrics used by Barzilay and Elhadad.

Score(Chain)>Average(scores) +2 * Standard Deviation. (1)

This metric is used to determine the strong chains in the document which will be used for extraction of sentences.

Sentences are extracted in a similar way as in [16]. For each of the strong chains computed, sentence that contains the first appearance of the representative chain member is selected. The representative chain members are those having frequency in the chain greater than the average.

IV. DATA SET

We have used the dataset provided for Text Summarization at annually held Document Understanding Conferences (DUCs) conducted by National Institute of Standards and Technology, U.S. Department of Commerce. The DUCs has been held from 2001 to 2007 and after that the summarization task was subsumed under Text Analysis Conferences (TACs) being conducted by National Institute of Standards and Technology from 2008. At the Data Understanding Conferences, tasks were defined and data was provided. The participants are asked to evaluate their systems with respect to the task. The data consisted of text documents with human produced abstracts. The abstracts were the desired summaries. The abstracts were produced both as single document summaries for each document and as multi-document summaries of sets of documents.

From DUC 2001, we have used 300 documents for single document summarization and 30 test data sets for multi-document summarization. A total of 600 documents are used for single document summarization and 60 document sets for multi-document summarization from DUC 2002. The DUC 2003 dataset consists of 30 TDT clusters (taken from Topic Detection and Tracking Project 1998, Information Access Division, NIST) with 298 documents (10 documents per cluster) and approximately 352 sentences per cluster and 30 TREC clusters (taken from Text Retrieval Conferences chosen by NIST assessors on topics of interest to them) with 326 documents (10 documents per cluster) and 335 sentences per cluster and 30 TREC Novelty clusters (with 66 relevant sentences per cluster). The manually created summaries were also provided for four tasks. We have used the TDT cluster as it had manually created abstracts for Task 2 (100 word multi document summaries). The DUC 2003 dataset consists of 50 TDT English news clusters of different topics with 10 documents per cluster, 24 TDT Arabic news clusters and 50 TREC English news clusters with 10 documents per clusters. We have selected TDT cluster dataset because they contained abstracts pertaining to Task 2 of DUC 2002.

V. EXPERIMENTS AND RESULTS

We have compared our algorithm with Barzilay and Elhadad and Silber and McCoy's work. We have used the evaluation scheme suggested by Silber and McCoy in which the authors have suggested a method to evaluate lexical chains independently of the sentence extraction phase for formulating the summaries. The underlying basis of their evaluation scheme is the premise that if lexical chains are a good intermediate representation, then each noun in the summary should be used in the same sense as corresponding word instance inserted into a strong chain in the original document. Moreover, all (or at least, most) strong chains in the document should have their representative nouns in the summary. Hence, the idea is to determine the extent to which the concepts contained in strong lexical chains in the original document appear in the summary and whether concepts which appear in the summary (as determined by the lexical chain construction on the summary) also belong to strong chains produced from the document. If both the metrics give 100% coverage this would mean that all and only the concepts identified by strong lexical chains in the document occur in the summary. The two metrics are formulated as:

- Metric 1: The number and percentage of strong chains from the original text that are represented in the summary. Representation in summary means:
  - – At least one of the nouns belonging to the strong chain of the document appears in the summary.
  - – The noun appears in the same sense in the summary as the sense in the document.

  This metric is analogous to recall.
- Metric 2: The percentage of noun instances appearing in the summary which represent one of the strong chains from the document. This is analogous to precision.

The experiments were carried out for DUC 2001, DUC 2002, DUC 2003 and DUC 2004 datasets for both single document and multi-document summarization. The recall and precision scores are presented for the three algorithms namely Barzilay and Elhadad, Silber and McCoy and our Lexical Chain algorithm.

1. Experiments conducted on DUC 2001 dataset
   a) Single document summarization

For single document summarization, we have used 300 documents from the dataset. The mean, median and standard deviation values for recall and precision metric are calculated for all the 300 documents which are shown in Table 1 and Table 2 respectively.

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	77.8	78.2	3.64
Barzilay and Elhadad	82.3	81.3	3.14
Lexical Chain	83.4	80.1	2.89

Table 1: Mean, Median and Standard Deviation values for recall metric for DUC 2001 dataset for single document summarization

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	79.2	77.6	3.85
Barzilay and Elhadad	83.7	81.8	3.21
Lexical Chain	83.4	83.6	3.36

Table 2: Mean, Median and Standard Deviation values for precision metric for DUC 2001 dataset for single document summarization

b) Multi-document summarization

The Recall and Precision scores were calculated for each of the 30 Test Documents Sets. Table 3 and Table 4 represent the mean, median and standard deviation

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	81.3	80	3.64
Barzilay and Elhadad	86.5	87	3.34
Lexical Chain	86.4	86	3.29

Table 3: Mean, Median and Standard Deviation values for recall metric for DUC 2001 dataset for multi-document summarization

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	83.6	83	2.41
Barzilay and Elhadad	86.8	85	2.26
Lexical Chain	88.3	87	3.15

Table 4: Mean, Median and Standard Deviation values for precision metric for DUC 2001 dataset for multi-document summarization

2. Experiments conducted on DUC 2002 dataset
   a) Single document summarization

The Recall and Precision scores were calculated for each of the 600 Documents.

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	78.57	79.2	2.96
Barzilay and Elhadad	83.6	84.5	2.83
Lexical Chain	85.62	86.3	1.43

Table 5: Mean, Median and Standard Deviation values for recall metric for DUC 2002 dataset for single document summarization

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	77.78	76.12	2.59
Barzilay and Elhadad	82.03	83.15	1.99
Lexical Chain	82.93	84.3	2.63

Table 6: Mean, Median and Standard Deviation values for precision metric for DUC 2002 dataset for multi-document summarization

b) Multi-document summarization

The Recall and Precision scores were calculated for each of the 60 Test Documents Sets.

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	79.83	79.2	2.6
Barzilay and Elhadad	82.66	80.1	2.12
Lexical Chain	83.85	81.3	2.73

Table 7: Mean, Median and Standard Deviation values for recall metric for DUC 2002 dataset for multi-document summarization

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	77.32	79.2	4.55
Barzilay and Elhadad	81.43	81.6	4.12
Lexical Chain	82.86	82.1	3.69

Table 8: Mean, Median and Standard Deviation values for precision metric for DUC 2002 dataset for multi-document summarization

3. Experiments conducted on DUC 2003 dataset for multi-document summarization

The Recall and Precision scores were calculated for each of the 30 documents clusters.

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	78.3	77.3	3.19
Barzilay and Elhadad	79.7	81	3.16
Lexical Chain	80.3	82.2	3.23

Table 9: Mean, Median and Standard Deviation values for recall metric for DUC 2003 dataset for multi-document summarization

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	74.13	76.1	2.71
Barzilay and Elhadad	80.2	80	2.15
Lexical Chain	81.96	86.3	1.43

Table 10: Mean, Median and Standard Deviation values for precision metric for DUC 2003 dataset for multi-document summarization

4. Experiments conducted on DUC 2004 dataset for multi-document summarization

The Recall and Precision scores were calculated for each of the 50 documents clusters.

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	78.11	78	4.21
Barzilay and Elhadad	82.7	81	3.97
Lexical Chain	83.91	81.6	3.07

Table 11: Mean, Median and Standard Deviation values for recall metric for DUC 2004 dataset for multi-document summarization

Algorithms	Mean	Median	Standard Deviation
Silber and McCoy	76.8	76	3.7
Barzilay and Elhadad	81.3	81.5	3.6
Lexical Chain	82.9	83.2	2.95

Table 11: Mean, Median and Standard Deviation values for precision metric for DUC 2004 dataset for multi-document summarization

We have conducted experiments for single document summarization on DUC 2001 and DUC 2002 datasets and multi-document summarization on DUC 2001 to DUC 2004 datasets. The observations may be summarized as given below.

- The Lexical Chain algorithm show better results (i.e. higher recall and precision scores) for all the documents
- The Lexical Chain algorithm show better mean and median values for both recall and precision as compared to Barzillay and Elhadad and Silber and McCoy algorithms.
- The Lexical Chain algorithm shows better standard deviation (i.e. smaller value which means less variation in individual values) than the other two algorithms for recall.

VI. CONCLUSION AND FUTURE WORK

We have proposed a new method of text summarization using lexical cohesion and produces better results than the existing lexical chain algorithms.

We have observed that human produced extracts have a large number of sentences which contain proper nouns. This is expected because the sentences containing names of entities are related to the topic of the text. Thus the performance of lexical cohesion based technique can be improved if those sentences are included in the text.

ACKNOWLEDGMENT

We would like to acknowledge Ms Angela Elli, National Institute of Standards and Technology, US Department of Commerce, for providing the datasets of Document Understanding Conferences 2001 to 2004.

REFERENCES

[1] K. S. Jones, "Automatic summarising: The state of the art," Information Processing & Management, 43.6, 2007, 1449-1481

[2] M. Mani, Advances in automatic text summarization, MIT Press, 1999.

[3] M. Palomer and E. Lloret, Text summarization in progress: a literature review, Artificial Intelligence Review, Springer, 2012.

[4] H. Luhn, The automatic creation of literature abstracts, IBM Journal of Research and Development, 1958.

[5] Edmundson, New methods in automatic extracting, Journal of ACM, 1969, pp 264-285.

[6] E. Hovy and C.Y. Lin, "Automated text summarization and the SUMMARIST system," Proceedings of a workshop held on Baltimore, Maryland, October 13-15, 1998.

[7] J Kupiec, J Pedersen and F. Chen, "A trainable document summarizer," Proceedings of the 18th annual international ACM SIGIR conference on Research and Development in information retrieval, ACM, 1995.

[8] S. Teufel and M. Moens, "Sentence extraction as a classification task," Proceedings of the ACL, Vol. 97, No. 1997, 1997.

[9] D. Marcu, Discourse trees are good indicators of importance in text, In Advances in automatic text summarization, 1999, pp. 123-136.

[10] W. C. Mann and S. A. Thompson, "Relational propositions in discourse," Discourse processes 9.1, 1986, pp. 57-90.

[11] K. S. Jones, "What might be in a summary?," Information retrieval 93, 1993, pp. 9-26.

[12] M. Halliday and R. Hasan,Cohesion in English, Longman Group, 1976.

[13] J. Morris and G. Hirst, Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text, Association Computational Linguistics, 1991.

[14] P. M. Roget, Roget's Thesaurus of English Words and Phrases, TY Crowell Company, 1911.

[15] G. Hirst and D. St. Onge, "Lexical chains as representations of context for the detection and correction of malapropisms," WordNet: An electronic lexical database, 1998, 305-332.

[16] R. Barzilay and M. Elhadad, Using lexical chains for text summarization, Advances in automatic text summarization, 1999, pp. 111-122.

[17] O. Medelyan, "Computing lexical chains with graph clustering," Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop, Association of Computational Linguistics, 2007.

[18] M. Galley and K. McKeown, "Improving word sense disambiguation in lexical chaining," IJCAI, Vol. 3, 2003.

[19] H.G. Silber and K. F. McCoy, "Efficient text summarization using lexical chains," Proceedings of the 5th international conference on intelligent user interfaces, ACM, 2000.

[20] M. Brunn, Y Chali and C. J. Pinchak, "Text summarization using lexical chains," Proceedings of Document Understanding Conference, 2001.

[21] O. Medelyan, "Computing lexical chains with graph clustering," Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop, Association for Computational Linguistics, 2007.

[22] C. Biemann, Chinese Whisper an Efficient Graph Clustering Algorithm and its Application to Natural Lanaguage Processing Problems, Proceedings of the HLT-NAACL-06 Workshop on Textgraphs, 2006, pp. 73-80.

[23] E. Brill, A simple rule-based part-of-speech tagger, Proceedings of the Third Conference on Applied Computational Linguistics, Trento, Italy, 1992.