

Lexical Cohesion Based Topic Modeling for Summarization

Gonenc Ercan and Ilyas Cicekli

Dept. of Computer Engineering
Bilkent University, Ankara, Turkey
ercangu@cs.bilkent.edu.tr, ilyas@cs.bilkent.edu.tr

Abstract. In this paper, we attack the problem of forming extracts for text summarization. Forming extracts involves selecting the most representative and significant sentences from the text. Our method takes advantage of the lexical cohesion structure in the text in order to evaluate significance of sentences. Lexical chains have been used in summarization research to analyze the lexical cohesion structure and represent topics in a text. Our algorithm represents topics by sets of co-located lexical chains to take advantage of more lexical cohesion clues. Our algorithm segments the text with respect to each topic and finds the most important topic segments. Our summarization algorithm has achieved better results, compared to some other lexical chain based algorithms.

Keywords: text summarization, lexical cohesion, lexical chains.

1 Introduction

Summary is the condensed representation of a document's content. For this reason, they are low cost indicators of relevance. Summaries could be used in different applications both as informative tools for humans and as similarity functions for information retrieval applications. Summaries could be displayed in search results as an informative tool for the user. The user can measure the relevance of a document that he gets as a result of a search on Internet by just looking its summary. In order to measure similarities between documents, their summaries can be used instead of whole documents, and indexing algorithms can index their summaries instead of whole documents.

Depending on its content, summaries can be categorized into two groups: *extract* and *abstract*. If a summary is formed of sentences that appear in the original text, it is called as an *extract*. A summarization system targeting extracts should evaluate each sentence for its importance. Abstracts are the summaries that are formed from paraphrased or generated sentences. Building abstracts has additional challenges.

Different clues can be exploited to evaluate the importance of sentences. There are extractive summarization systems that take advantage of surface level features like word repetition, position in text, cue phrases and similar features that are easy to compute. Ideally, a summarization system should perform full understanding, which is very difficult and only domain dependant solutions are currently available.

Some summarization algorithms including ours, rely on more sophisticated clues that require deeper analyses of the text. A meaningful text is not a random sequence of

words, and it has a semantic integrity to explain one or more topics. In linguistics, coherence is used to define the semantic integrity of a document, and it can be thought as a hidden element which provides the feeling that a document is written intelligently. Since modelling coherence which indicates the semantic structure of a document is difficult, researchers looked other low-cost measures for the semantic structures of documents. Cohesion [8] is simpler than coherence and it can also help to determine the discourse structure in the text. Cohesion is a surface level feature, and it deals with the relationships between text units. Some cohesion relations are lexical cohesion (use of related terms), co-reference, ellipsis and conjunction. Co-reference, ellipsis and conjunction are harder to identify than lexical cohesion.

Modeling the lexical cohesion structure of a text depends on the semantic relations between words in the text. The lexical cohesion structure of the text can be modeled with lexical chains [10]. Lexical chains are connected graphs, where the vertices are intended senses (meanings) of the words and the edges are the semantic relations between these senses. A lexical chaining algorithm needs an ontology to acquire the semantic relations between senses. WordNet is such an ontology, which is used by our algorithm and other lexical chaining algorithms in the literature. In order to find the lexical chains for the text, the intended sense for each word in the text must be determined. This is also known as Word Sense Disambiguation (WSD).

In lexical chaining algorithms, the WSD is done by assuming that the intended sense of a word is more related to other words surrounding the word. Morris and Hirst define the first lexical chaining algorithm, which is a greedy algorithm that tries to disambiguate each word using the context before its occurrence [10]. Barzilay and Elhadad disambiguate the words by checking all possible interpretations of the text [1]. Galley and McKeown improve Barzilay’s algorithm both in terms of running time and WSD accuracy [7]. Galley imposes a “one sense per word” constraint and fuses the clues gathered from different occurrences of a word to a single decision for the word’s correct sense. In our algorithm, we are using a very similar algorithm to Galley et al.’s algorithm. Our lexical chaining procedure differs only in the WordNet relations used in the algorithm. Our algorithm also uses meronym and holonym relations while their algorithm does not consider these relations.

Barzilay and Elhadad introduce a text summarization algorithm based on lexical chains [1]. They claim that cohesion relations could provide good results in text summarization. Their algorithm uses lexical chains to detect and represent topics. Their lexical chaining algorithm also depends on an explicit text segmentation algorithm. They report that they have experimented with different sentence extraction criterion, and selecting the first sentences of the strongest lexical chains yields the best results. A strength criterion used by Barzilay is shown in Equation 1, and *Homogeneity* is shown in Equation 2. In those equations, *Length* is the number of all members of the lexical chain, and *#DistinctMembers* is the number of the distinct members of the lexical chain.

$$Score(Chain) = Length * Homogeneity \quad (1)$$

$$Homogeneity = 1 - \frac{\#DistinctMembers}{Length} \quad (2)$$

Brunn et al.[2] also use lexical chains for text summarization. Just like Barzilay et al.'s algorithm, they use an explicit text segmenter. Two phase sentence selection procedure is applied. First the segments are ranked with lexical chain scores. From the best scoring text segments, the most scoring sentences are selected. Doran et al. [4] describe a similar summarization algorithm.

In all of these algorithms only lexical cohesion is used. These algorithms treat topics as single lexical chains. We believe that a single lexical chain can not represent a whole topic by itself. Usually a topic receives contributions from several lexical chains. Our algorithm tries to exploit other lexical cohesion clues like substitution, co-reference and ellipsis without detecting them. The lexical chains that tend to co-occur in a text can indicate a context specific relation between these lexical chains. Three lexical chains could correspond to a topic as their members could correspond to "What", "When" and "Where" portions of the topic. In our algorithm, we try to cluster related lexical chains in order to represent topic in the text.

We present our summarization algorithm in Section 2. In that section, we explain how we clusters the lexical chains and how we select important sentences using these clusters. In Section 3, we evaluate the results of our summarization system by comparing the results of the summarization systems in DUC2004[12]. Finally, we give some concluding remarks in Section 4.

2 Summarization Algorithm

Our algorithm divides the text into segments, according to topics, using the lexical chains extracted from the text. Topics in the text are roughly determined using lexical chains. Through clustering of lexical chains, our algorithm produces more granular segments. In each segment, it is assumed that the first sentence is a general description of the topic, and the first sentence of the segment is included in the summary.

Our algorithm is based on lexical chains, for this reason, our system requires deeper analysis of the text. An outline of our algorithm could be given as:

1. Sentence Detection
2. Part of Speech Tagging
3. Noun Phrase Detection
4. Lexical Chaining
5. Filtering Weak Lexical Chains
6. Clustering Lexical Chains Based on Co-occurrence
7. Extracting Sequences / Segmenting the Text Regard to Clusters.

Part of Speech Tagging is done using the MaxEnt Part of Speech Tagger [11]. We have implemented a noun phrase skimmer that uses the part of speech tags to detect noun phrases. Noun phrases usually end with a head noun. This head noun is accompanied by zero or more pre-modifiers, which usually are nouns or adjectives. The nouns and simple noun phrases of a document are found at the end of the first three steps of our algorithm, and the lexical chains are created for them in the fourth step.

Our lexical chaining algorithm is an implementation of Galley et al.'s algorithm [7], and it is also used in a keyphrase extraction system based on lexical chains [6].

After lexical chains are constructed for the text, there will be some weak lexical chains formed of single word senses. These lexical chains can cause complications in topic identification and segmentation. The formula in Equation 1 is introduced by Barzilay et al. [1], and this formula has been formulated to reflect the strength of lexical chains. Barzilay et al. report that this is the best formula that correlates with the human judges. After lexical chain construction, Barzilay suggests that lexical chains below a certain strength criterion should be filtered. We use strength criterion defined in Equation 3 to filter weak lexical chains before clustering lexical chains. In Equation 3, $Score(Chain)$ is the score of the lexical chain, $Avg(Scs)$ is the average of the scores of all lexical chains, and $StdDev(Scs)$ is the standard deviation of the scores. This equation is first introduced by Barzilay et al. [1] and they report that this criterion correlates with the human judgements.

$$Score(Chain) > Avg(Scs) + 2 * StdDev(Scs) \quad (3)$$

After the weak lexical chains are filtered, the remaining lexical chains are clustered using co-occurrence information. We hope that the remaining strong clusters represent major topics of the text, and important sentences are extracted from these strong clusters. The details of clustering and sentence extraction are discussed in the rest of this section.

2.1 Clustering Lexical Chains

All strong lexical chains in the document are clustered using co-occurrence statistics. A single lexical chain may not be sufficient to represent a single topic. Our summarization algorithm uses clusters of lexical chains in order to represent topics in the text. Figure 2 gives the important clusters of lexical chains constructed for the document in Figure 1¹.

A topic could be formed of words that are not necessarily co-related. For example, in Figure 2 cluster 2 is a good example. This cluster talks about an 'arrest' in 'London' on 'Sunday'. These three sets and their relations with each other can only be determined by the current context. We believe that through clustering, we are forming a relation between these lexical chains. In cluster 2, lexical chains in the cluster are forming up the relations 'what', 'where' and 'when' respectively. Our clustering algorithm uses a very simple assumption: "if two lexical chains tend to appear in same sentences, then there may be a relation between two sets in the given context". It is clear that, this will not hold in all cases. There will be falsely related lexical chains, however, a more accurate algorithm requires deeper semantic analysis. Our approach is just accurate enough for our segmentation algorithm.

In cohesion relations, like reference, substitution and ellipsis, a word is not repeated in each sentence but replaced or omitted. Through clustering, we can be able to account for cohesion clues other than lexical cohesion, for example ellipsis. By forming the link between two or more lexical chains by co-occurrence, it is possible to consider all lexical cohesion relations while segmenting the text.

¹ Proper names in the text, 'Pinochet' and 'Frei' are not present in WordNet. We have ignored nouns that are not in WordNet. Thus, 'Pinochet' and 'Frei' are not considered in our algorithm.

Cuban President Fidel Castro said Sunday he disagreed with the arrest in London of former Chilean dictator Augusto Pinochet, calling it a case of 'international meddling.' 'It seems to me that what has happened there (in London) is universal meddling,' Castro told reporters covering the Ibero-American summit being held here Sunday. Castro had just finished breakfast with King Juan Carlos of Spain in a city hotel. He said the case seemed to be 'unprecedented and unusual.' Pinochet, 82, was placed under arrest in London Friday by British police acting on a warrant issued by a Spanish judge. The judge is probing Pinochet's role in the death of Spaniards in Chile under his rule in the 1970s and 80s. The Chilean government has protested Pinochet's arrest, insisting that as a senator he was traveling on a diplomatic passport and had immunity from arrest. Castro, Latin America's only remaining authoritarian leader, said he lacked details on the case against Pinochet, but said he thought it placed the government of Chile and President Eduardo Frei in an uncomfortable position while Frei is attending the summit. Castro compared the action with the establishment in Rome in August of an International Criminal Court, a move Cuba has expressed reservations about. Castro said the court ought to be independent of the U.N. Security Council, because "we already know who commands there," an apparent reference to the United States. The United States was one of only seven countries that voted against creating the court. "The (Pinochet) case is serious ... the problem is delicate" and the reactions of the Chilean Parliament and armed forces bear watching, Castro said. He expressed surprise that the British had arrested Pinochet, especially since he had provided support to England during its 1982 war with Argentina over the Falkland Islands. Although Chile maintained neutrality during the war, it was accused of providing military intelligence to the British. Castro joked that he would have thought police could have waited another 24 hours to avoid having the arrest of Pinochet overshadow the summit being held here. "Now they are talking about the arrest of Pinochet instead of the summit," he said. Pinochet left government in 1990, but remained as army chief until March when he became a senator-for-life.

Fig. 1: An Example News Article

For each lexical chain LC_i , a sentence occurrence vector V_i is formed. $V_i = \{s_{1_i}, \dots, s_{k_i} \dots s_{n_i}\}$ where n is the number of sentences in the document. Each s_{k_i} is the number of LC_i members in the sentence k . If sentence k has 3 members of LC_i then s_{k_i} is 3. Two lexical chains LC_i and LC_j goes into the same cluster if their sentence occurrence vectors V_i and V_j are similar. As a result of clustering of lexical chains, we will get the following two properties:

- Lexical chains that co-occur will be in the same cluster. These lexical chains form a set of related topics that talk about a single topic.
- Lexical chains that span different sentences will be in different clusters. Two lexical chains that are in different clusters are considered to be unrelated.

Our clustering algorithm starts from an initial cluster distribution, where each lexical chain is in its own cluster. Thus, our clustering algorithm starts with n clusters, where n is the number of lexical chains. Iteratively, the most similar cluster pair is found and they are merged to form a single cluster. Clustering stops when the similarity between the most similar clusters is lower than a predefined threshold value.

The similarity between two clusters is measured by finding the similarity between the least similar members of the cluster. This is called *complete link clustering*. Since cluster members are lexical chains in our algorithm, a similarity function measuring the co-occurrence between two lexical chains is needed. We have used cosine similarity for

<p>Cluster1 : $LC_1 = \{\text{Castro, Castro, chief, Castro, Castro, Castro, Castro, Castro, leader}\}$ $V_1 = \{1,1,1,0,0,0,2,1,1,0,1,0,0,1,0,1\}$ $LC_2 = \{\text{establishment, United States, parliament, United States, government, government, gov-ernment}\}$ $V_2 = \{0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,1\}$</p> <p>Cluster2 : $LC_1 = \{\text{action, march, meddling, arrest, arrest, arrest, surprise, arrest, meddling, arrest, arrest}\}$ $V_1 = \{2,1,0,0,1,0,2,0,1,0,0,0,1,0,1,1,1\}$ $LC_2 = \{\text{London, London}\}$ $V_2 = \{1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0\}$ $LC_3 = \{\text{Sunday, Sunday}\}$ $V_3 = \{1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0\}$</p> <p>Cluster 3 : $LC_1 = \{\text{summit, summit, summit, summit}\}$ $V_1 = \{0,1,0,0,0,0,0,1,0,0,0,0,0,0,1,1,0\}$</p> <p>Cluster 4 : $LC_1 = \{\text{Chile, Argentina, Chile, Chile}\}$ $V_1 = \{0,0,0,0,0,1,0,1,0,0,0,0,1,1,0,0,0\}$ $LC_2 = \{\text{war, war}\}$ $V_2 = \{0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0\}$</p> <p>Cluster 5 : $LC_1 = \{\text{court, court, court}\}$ $V_1 = \{0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0\}$</p>

Fig. 2: Lexical Chain Clusters for the Example in Figure 1

this purpose. Lexical chain occurrence vector V_i is a vector in an m dimensional space, where m is the number of sentences. The angle between two vectors could be used to find the similarity of two vectors. Between two vectors that are in the same direction, there will be an angle of 0 degrees. Cosine of two vectors can be calculated by Equation 4. This value is between 0 and 1, where 1 means most similar.

$$\cos(\theta) = \frac{V_i \cdot V_j}{\|V_i\| \|V_j\|} \quad (4)$$

Equation 4 is a well known formula from linear algebra, to find the cosine of the angle between two vectors. In the equation, $\|V_i\|$ represents the Euclidean length for the vector, that is the square root of the sum of squares of vector's dimension values .

2.2 Sequence Extraction or Text Segmentation

Some previous algorithms for lexical chain based summarization such as Brunn et al. [2], and Barzilay et al. [1] use explicit segmentation algorithms that does not take advantage of semantic relations. In our algorithm, the text is segmented from the perspective of each lexical chain cluster, and the hot spots for each topic are found. For each cluster, connected sequences of sentences are extracted as segments. Sentences that are cohesively connected are considered as sentences that are talking about the same topic.

$$V_1 = \{ \boxed{1,1,1}, 0,0,0, \boxed{0,2,1,1,0,1}, 0,0, \boxed{1}, 0, \boxed{1} \}$$

$$V_2 = \{ \boxed{0,0,0}, 0,0,0, \boxed{1,1,1,1,1,1}, 0,0, \boxed{0}, 0, \boxed{1} \}$$

Fig. 3: Text Segmentation for Cluster 1 Given in Figure 2

For each lexical chain cluster Cl_j , we form sequences separately. For each sentence S_k , if sentence S_k has a lexical chain member in Cl_j , a new sequence is started with sentence S_k or the sentence is added to the current sequence. If there is a current sequence, sentence S_k is added to this sequence; otherwise sentence S_k starts a new sequence. If there is no cluster member for sentence S_k , then the current sequence is ended. By using this procedure, text is segmented with respect to a cluster, identifying topic concentration points.

Figure 3 gives an example of the text segmentation for the document in Figure 1 with respect to cluster 1 that is given in Figure 2. In cluster 1, there are two lexical chains. The sentence occurrence vectors for these lexical chains are plotted in Figure 3, and boxed areas correspond to the sequences in the text. The topic seems to be concentrated in the second sequence, and the second sequence has contributions from both of the lexical chains and spans more than the other sequences.

After finding sequence, each sequence s_i is scored using the formula in Equation 5.

$$S(s_i) = S(Cl_i) * L_i * \frac{(1 + SLC_i) * PLC_i}{f^2} \quad (5)$$

$S(s_i)$ in Equation 5 is the score of segment with respect to cluster i . In Equation 5, L_i is the number of sentences in the sequence s_i , SLC_i is the number of lexical chains that starts in the sequence s_i , PLC_i is the number of lexical chains having a member in the sequence s_i , and f is the number of lexical chains in cluster i . Score of the cluster $S(Cl_i)$, is the average score of the lexical chains in the cluster. Our scoring function tries to model the connectedness of the segment using this cluster score. In order to evaluate this score, the scores of the lexical chains in the cluster are calculated with the formula in Equation 1. The number of sentences in the segment reflects how long the topic is discussed locally. Our algorithm tries to select the segments that lexical chains are starting in, and this will encourage the selection of the segments where the topic is first introduced in.

2.3 Sentence Selection

Humans tend to first explain the topic more generally, and then they give details in the following sentences. With this motivation, our algorithm extracts the first sentence of each sequence. So, if the extracted sequences are truly topic segments for the text, then our algorithm will extract the first sentence of the new topic. This technique depends on the assumption that, first sentences are general descriptions of the topic and this general description does contain sufficient information to represent the text segment in the summary.

The Chilean government has protested Pinochet's arrest, insisting that as a senator he was traveling on a diplomatic passport and had immunity from arrest. Cuban President Fidel Castro said Sunday he disagreed with the arrest in London of former Chilean dictator Augusto Pinochet, calling it a case of international meddling. Castro compared the action with the establishment in Rome in August of an International Criminal Court, a move Cuba has expressed reservations about. He expressed surprise that the British had arrested Pinochet, especially since he had provided support to England during its 1982 war with Argentina over the Falkland Islands.

Fig. 4: Extract of the Text in Figure 1

For a summary of length n sentences, n best scoring sequence's first sentences are included in the summary. However, two different sequences found from different lexical chain clusters can start with the same sentence. A problem with this approach may be that n could be higher than the number of sequences starting with a unique sentence, so the number of sentences to be included in the summary is limited by the number of sequences starting with unique sentences. It is possible for two sequences extracted from different lexical chain clusters to overlap in text area. The order of sentences in the output summary depends on the score of the sequence, which the sentence is extracted from. Sentences selected from the best scoring sequence will be the first sentence in the output summary.

We will try to demonstrate our algorithm using the news article in Figure 1. After lexical chaining and clustering, top ranking clusters are given in Figure 2. In cluster 4, the connection between 'Chile' and 'Argentina' is 'war'. This is discovered from the given context using co-occurrence in the given text. Clustering increases the connectedness of sentences, resulting in granular text segments. Sequences are extracted for these clusters. As a result of this process, summary in Figure 4 is extracted.

3 Evaluation Metrics

Evaluating summarization algorithms is a difficult task and it is a separate research area in Natural Language Processing. A summary's quality can be evaluated in different aspects: selected contents importance, and presentation quality. Presentation quality itself is composed of two aspects: grammatical correctness and coherence. Since we are extracting sentences from the original text, the grammatical correctness in sentences is guaranteed to be as good as the source document's grammatical correctness. Coherence in our solution is a problem as our algorithm does not consider anaphora resolution and information ordering. However, since we extract the first sentences of topic segments, anaphoric references in our extracts are not common.

An evaluation method is the evaluation of summaries by human judges. However, comparing the contents of automatically built summaries with human extracted summaries is a more fair methodology. Automatic evaluation is done using distributed similarity techniques. The similarity between the model summary and the system output reflects the summary quality. The overlap of text units between the system output and the model summaries is used as a quality metric. In the evaluation procedure, it is more appropriate to use multiple model summaries by different summarizers, since summarization is a subjective task.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [9] is one of the most popular summarization evaluation methodologies. ROUGE calculates the recall of text units using N-grams, LCS (Longest Common Subsequences) and Weighted Longest Common Subsequences. All of these metrics are aimed to find the percentage of overlap between the system output and the model summaries. ROUGE-N score is the percentage of overlap calculated using N-grams. ROUGE-L score is calculated using LCS and ROUGE-W score is calculated using Weighted LCS.

3.1 Experiment Setting and Results

We have tested our summarization algorithm with the news article corpus used in DUC2004 [12]. In order to properly evaluate our algorithm, and compare with existing algorithms, we have attempted task 1 of DUC2004. In this task, all summarization systems provide a 75 character summary for each of the 500 articles. Each summary is automatically evaluated against 4 model summaries extracted by professional humans. While calculating ROUGE scores, words in both the model and the system output are stemmed using Porter Stemmer. Weight for calculating the WLCS is assigned as 1.2. These are the values used in DUC2004 and we have used the same values to be compatible with their evaluation.

Table 1 shows the scores for our system, the best system and the worst system of the 40 systems participated in DUC2004. The average score of the participants of DUC2004 is also given in this table. We also included the scores of two systems, which are also participants of DUC2004 and they also use lexical cohesion methods for summarization. Lethbridge University's [3] summarization system also attacks automated summarization problem using lexical chains. Their algorithm uses an explicit text segmenter, and after building lexical chains they score each segment using the lexical chains. From the best segments, they select sentences. This algorithm is derived from Brunn et. al.'s algorithm [2]. Another algorithm using lexical cohesion in DUC2004 is the system developed in Dublin University [5]. This system extracts phrases instead of sentences. System ranks each phrase using TFxIDF (Term Frequency x Inverse Document Frequency), position of word, lexical cohesion score and POS tags. They use C5.0 machine learning algorithm to classify these phrases.

Our implementation of Barzilay et al.'s algorithm uses our lexical chaining procedure, but uses their selection procedure. Their algorithm selects the first sentence where a lexical chain member occurs in. In their algorithm, a strong lexical chain contributes to the summary with only one sentence. They assume that a lexical chain is a topic and the first sentence is the most important sentence.

Since a lexical chaining algorithm's word sense disambiguation accuracy is as low as %63, it is possible that the first member of a lexical chain is an error. In our algorithm, lexical chains are used as an intermediate tool to find topic segments. Segments are identified by combining the cues obtained from co-occurring lexical chains. Co-occurring lexical chains may capture context specific relations and other cohesion patterns. Our segments reflect the lexical cohesion hot spots, while the whole lexical chain reflects a set of related terms that may be scattered to the whole document. We select the first sentences of the most lexically cohesive segments. We believe that our sentence selection procedure is more prone to errors in lexical chaining than Barzilay's algorithm.

Table 1. ROUGE Scores of our System and Other Participants of DUC2004

	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W
Barzilay	0.17861	0.04381	0.15577	0.09508
Lethbridge	0.12135	0.02504	0.10852	0.06604
Dublin	0.22192	0.02543	0.1766	0.10169
Our System	0.19549	0.05247	0.17078	0.1034
Average	0.1858	0.04082	0.15803	0.09470
Best System	0.2511	0.06528	0.20109	0.11953
Worst System	0.12088	0.00731	0.10678	0.06564

3.2 Results

Scores of our system is promising as it is above Barzilay’s algorithm. Also Lethbridge University’s algorithm has obtained results below our system. System by Dublin university is above our algorithm in ROUGE-1 scores. However they have lower scores in other ROUGE scores, this is mainly because their algorithm outputs phrases. In DUC2004 evaluation, stop words are not removed when calculating recall. The model summaries for evaluation are formed of sentences containing stop words, for this reason their system has lower matches of sequences of words.

Table 2. ROUGE Ranks of our System and Other Participants of DUC2004

	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W
Barzilay	28	15	26	22
Lethbridge	41	38	41	41
Dublin	5	36	7	9
Our System	17	8	9	7

Table 2 shows the rank of each system when compared to participants of DUC2004 single document summarization task. Our system ranked in the first 10 in all of the scores except ROUGE-1 score, which is calculated using uni-grams. In overall, our system achieved very good results. These results reflect that our system has obtained competing results for the algorithms in DUC2004. Since our algorithm outperforms lexical cohesion based algorithms, such as Barzilay’s algorithm, Dublin University’s algorithm and Lethbridge University’s algorithm, we can consider it as a promising attempt.

4 Conclusion

Our motivation for this work is based on the observation that a topic is formed of a group of lexical chains. This is mainly due to the fact that in the current context of the text, words can be related to each other with domain specific relations that can not be acquired from a general ontology. Our algorithm tries to find these relations from

the current text. Although this seems to be a weak assumption, we have seen in our experiments that our algorithm achieved better results than other lexical chains based algorithms.

Our system achieves very good results in DUC2004, ranking in the first 10. Our system is purely extractive, some other competing algorithms are using techniques such as: sentence reduction, anaphora resolution and elimination of repetition. In other competing algorithms, there are some systems that focus on news article domain, tracking events. Reduction of sentences could improve ROUGE score as summaries extracted are limited in size, some systems does have similar approaches. Resolving anaphora, improves the performance as model summaries does not usually contain anaphora.

Acknowledgments

This work is partially supported by The Scientific and Technical Council of Turkey Grant “TUBITAK EEEAG-107E151”.

References

1. Barzilay, R., Elhadad, M.: Using Lexical Chains for Text Summarization. In Mani, I., Maybury, M.T., eds.: *Advances in Automatic Text Summarization*. The MIT Press (1999) 111–121
2. Brunn, M., Chali, Y., Pinchak, C.J.: Text summarization using lexical chains. In: *Proceedings of the Document Understanding Conference (DUC01)*, New Orleans, LA (2001)
3. Chali, Y., Kolla, M.: University of lethridge summarizer at DUC04. In: *Proceedings of the Document Understanding Conference (DUC04)*, Boston, USA (2004)
4. Doran, W.P., Stokes, N., Carthy, J., Dunnion, J.: Assessing the impact of lexical chain scoring methods and sentence extraction schemes on summarization. In: *Proceedings of the 5th International Conferences on Intelligent Text Processing and Computational Linguistics (CICLing-2004)*, (2004) 627–635
5. Doran, W., Stokes, N., Newman, E., Dunnion, J., Carthy, J., Toolan, F.: News story gisting at university college dublin. In: *Proceedings of the Document Understanding Conference (DUC04)*, Boston, USA (2004)
6. Ercan, G., Cicekli, I.: Using lexical chains for keyword extraction. *Information Processing & Management* **43** (2007) 1705–1714
7. Galley, M., McKeown, K.: Improving word sense disambiguation in lexical chaining. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, (2003) 1486–1488
8. Halliday, M., Hasan, R.: *Cohesion in English*. Longman, London (1976)
9. Lin, C.Y., Hovy, E.H.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: *Proceedings of Proceedings of HLT-NAACL-2003*, Edmenton, Canada, (2003)
10. Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* **17** (1991) 21–43
11. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: *Proceedings of HLT-NAACL-2003*, Edmenton, Canada, (2003)
12. *Proceedings of the Document Understanding Conference (DUC04)*, Boston, USA (2004)