Assessing the Impact of Lexical Chain Scoring Methods and Sentence Extraction Schemes on Summarization.

William Doran, Nicola Stokes, Joe Carthy, John Dunnion.

Department of Computer Science,
University College Dublin, Ireland.
{William.Doran, Nicola.Stokes,Joe.Carthy,
John.Dunnion}@ucd.ie

Abstract. We present a comparative study of lexical chain-based summarisation techniques. The aim of this paper is to highlight the effect of lexical chain scoring metrics and sentence extraction techniques on summary generation. We present our own lexical chain-based summarisation system and compare it to other chain-based summarisation systems. We also compare the chain scoring and extraction techniques of our system to those of several other baseline systems, including a random summarizer and one based on tf.idf statistics. We use a task-orientated summarisation evaluation scheme that determines summary quality based on TDT story link detection performance.

1 Introduction

Summarisation is a reductive transformation of a source text into a summary text by extraction or generation [13]. It is generally agreed that automating the summarisation procedure should be based on text understanding that mimics the cognitive processes of humans. However, this is a sub-problem of Natural Language Processing (NLP) and is a very difficult problem to solve at present. It may take some time to reach a level where machines can fully understand documents, in the interim we must utilise other properties of text, such as lexical cohesion analysis, that do not rely on full comprehension of the text.

Lexical cohesion is the textual property responsible for making the sentences of a text seem to "hang together", indicated by the use of semantically related vocabulary [10]. Cohesion is thus a surface indicator of the discourse structure of a document. One method of representing this type of discourse structure is through the use of a linguistic technique called lexical chaining. Lexical chains are defined as clusters of semantically related words. For example, {*house, loft, home, cabin*} is a chain, where *house* and *home* are synonyms, *attic* is part of a *house* and *cabin* is a specialisation of *house*. The lexical chaining algorithms discussed in this paper identifies such lexical cohesive relationships between words using the WordNet taxonomy [9].

Since lexical chains were first proposed by Morris and Hirst [10], they have been used to address a variety of Information Retrieval (IR) and NLP applications, such as term weighting for IR tasks [15], malapropism detection [14], hypertext generation [6] and topic detection in broadcast news streams [16], to name but a few. More importantly however, in the context of this paper, lexical chains have been successfully used as an intermediate source text representation for document summarisation. This application of lexical chaining was first implemented by Barzilay and Elhadad [3]. They used lexical chains to weight the contribution of a sentence to the main topic of a document, where sentences with high numbers of chain words are extracted and presented as a summary of that document.

In this paper, we put forward novel methods of building extractive summaries of single documents using lexical chains. However, unlike other attempts to improve upon Barzilay and Elhadad's work [1, 4, 12], we evaluate our weighting and extraction schemes directly with theirs using an extrinsic or task-based evaluation technique. An intrinsic evaluation is the preferred method of evaluating summary quality used by most summarisation researchers. This type of evaluation requires a set of human judges to either create a set of gold standard summaries or score summary quality compared to the original text. However, this evaluation method is time consuming, expensive and quite often subjective and hence is inappropriate for estimating the effect of different schemes on summary performance. Therefore in this paper we propose a more efficient evaluation alternative based on the TDT story-link detection task [2], where summary quality is evaluated with respect to how well a story link detection system can determine if a pair of document summaries are similar (on-topic) or dissimilar (off-topic). We are also interested in finding out whether this type of evaluation is sensitive enough to pick up differences in the summary extraction techniques discussed in this paper. In the remainder of the paper, we explain in more detail how lexical chaining based summarisation works. We evaluate our scoring metrics and compare the best performing metric to Barzilay and Elhadad's metric. We also present our experimental methodology and results. The final section gives our conclusions and some future work.

2 Lexical Chaining and Text Summarisation

The basic chaining algorithm follows the following steps. First, we select a set of candidate words, in our case, nouns. Then search through the list of chains and if a word satisfies the relatedness criteria with a chain word then the word is added to the chain, otherwise a new chain is created.

The relatedness criteria are the relationships outlined by St.Onge [14]. St. Onge used WordNet [9] as the knowledge source for lexical chaining. He devised three different relationships between candidate words: extra-strong, strong and medium-strong. Extra-strong relations are lexical repetitions of a word and strong relations are synonyms or near-synonyms. Strong relations can also indicate a shared hypernym/hyponym or meronym/holonym, such that one word is a parent-node or child-node of the other in the WordNet topology. Medium-strength relations follow sets of rules laid out by St. Onge. These rules govern the shape of the paths that are allowable in the WordNet structure. St. Onge's algorithm uses a greedy disambiguation procedure where a word's sense is determined only by the senses of words that occur before it in the text. In contrast, a non-greedy approach waits until all words in the document are processed and then calculates the appropriate senses of all the words.

In general, most lexical chain based summarizers follow the same approach by firstly generating lexical chains, then the 'strongest' of these chains are used to weight and extract key sentences in the text. Barzilay and Elhadad [3] form chains using a non-greedy disambiguation procedure. To score chains they calculate the product of two chain characteristics: the length of the chain, which is the total number of words in the chain plus repetitions and, the homogeneity of the chain, which is equal to 1 minus the number of distinct words divided by the length of the chain. Chain scores that exceed an average chain score plus twice the standard deviation are considered 'strong' chains. Barzilay et al. then select the first sentence that contains a 'representative' word from a 'strong' chain, where a 'representative' word has a frequency greater than or equal to the average frequency of words in that chain.

Most other researchers use this approach to building extractive summaries using lexical chains [1, 12], with the exception of Brunn et al. [4] who calculate chain scores as the pair-wise sum of the chain word relationship strengths in the chain. In the latter, sentences are ranked based on the number of 'strong' chain words they contain.

3 The LexSum System

Our chaining algorithm LexSum is based on [14, 16] and uses a greedy lexical chaining approach. The first step in our chain formation process is to assign parts-of-speech to an incoming document. The algorithm then identifies all noun, proper nouns and compound noun phrases by searching for patterns of tags corresponding to these types of phrases e.g. presidential/JJ campaign/NN, or U.S/NN President/NN Bush/NP where /NN is a noun tag and /NP is a proper noun tag.

The nouns and compound nouns are chained by searching for lexical cohesive relationships between words in the text by following constrained paths in WordNet similar to those described in [14] using lexicographical relationships such as synonymy (*car, automobile*), specialisation/generalisation (*horse, stallion*), part-whole/whole-part (*politicians, government*). However, unlike previous chaining approaches our algorithm produces two disjoint sets of chains: noun chains and proper noun chains. Finding relationships between proper nouns is an essential element of modelling the topical content of any news story. Unfortunately, WordNet's coverage of proper nouns is limited to historical figures (e.g. Marco Polo, John Glenn) and so our algorithm uses a fuzzy string matching function to find repetition relationships between proper nouns phrases like George_*Bush* ⇔ President_*Bush*.

In this paper we present five different chain scoring metrics, three of which are based on semantic relationships between the words of the chain, another is based on corpus statistics, and the final metric assigns the same score to each chain. Unlike Barzilay et al.'s approach, the first three metrics calculate chain scores based on the number of repetitions and the type of WordNet relations between chain members. The differences between the three lie in the way relations in the chain are handled. More specifically, as shown in equation (1), the chain score is the sum of each score assigned to each word pair in the chain. Each word pair's score is calculated as the sum of the frequencies of the two words, multiplied by the relationship score between them,

$$chain\_score1(chain) = \sum(reps_i + reps_j) * rel(i,j). \qquad (1)$$

where $reps_i$ is the frequency of word i in the text, and $rel(i,j)$ is a score assigned based on the strength of the relationship between word i and j, where a synonym relationship gets assigned a value of 0.9, specialisation/generalisation and part-whole/whole-part 0.7. Proper nouns chain scores are calculated depending on the type of match, 1.0 for an exact match, 0.8 for a partial match and 0.7 for a fuzzy match.

The second metric, equation (2), assigns a score to each word in the chain depending on its relation to the most frequent member of the chain. Therefore, the most frequent word will get the highest score, the next frequent will get a lesser score depending on its frequency and how it is related to the most frequent word in the chain. If a word is not related to the most frequent word in the chain, we then take into account the word in the chain that it is related to.

$$chain\_score2(chain) = \sum(reps_i * rel(i,j)). \qquad (2)$$

where $reps_i$ is the frequency of word i in the text, and $rel(i,j)$ is a score assigned based on the strength of the relationship between word i and j, where word i is the most frequent word in the chain. The scores of the relationships are the same as in equation (1).

The third scoring metric, equation (3), assigns a score to each chain depending on the number of words and number of relations in the chain. The first word added is deemed the most important and any subsequent additions simply add to the total score of the chain. The scores of subsequent additions are not adversely affected as in equation (1) and equation(2).

$$chain\_score3(chain) = \sum(reps_i) + \sum rel(i,j)). \qquad (3)$$

where $reps_i$ is the frequency of word i in the text, and $rel(i,j)$ is a score assigned based on the strength of the relationship between word i and j as in equation(1).

The fourth scoring metric proposed in this paper is based on corpus statistics from an auxiliary corpus. The reason we use a different corpus is because our evaluation corpus only comprises 625 documents [5] and these documents are grouped into similar clusters. These documents have similar word distributions thus biasing the statistics. We use a larger corpus of 15000 documents taken from the TDT pilot corpus [17].

$$chain\_score4(chain) = \sum(reps_i)+ idf(word_i) \qquad (4)$$

where $reps_i$ is the frequency of word i in the text, and $idf(word_i)$ is a score assigned based on the inverse document frequency of the word taken from another corpus.[17]. If a chain word is a phrase, the score of the phrase is the score of the individual words comprising it. In the previous four weighting schemes we calculate a relative score for each chain, by dividing each chain's score by the largest chain score.

The final scoring scheme, equation (5), assigns the same default score of 1 to all the chains.

$$\mathit{chain\_score5(chain)} = 1 \tag{5}$$

The next step in the algorithm ranks sentences based on the sum of the scores of the words in each sentence. The summary will be based on the top ranking sentences. We have implemented three different ranking schemes: the first is based on the burstiness of chain words, the second is based simply on the chain scores and the third is a variant of the first scheme that incorporates a bonus score for words that occur in the first paragraph.

In the first sentence-ranking scheme, equation (6), a word's score is a scaled version of its chain's score. The scaling factor is the minimum distance between a word and its predecessor or its successor in the chain. This idea is based on the fact that general topics tend to span large sections of a discourse whereas subtopics tend to populate smaller areas. [7]. Therefore, the score of a word will be increased if semantically similar words are close by it in the text i.e. the topic is in the focus of the reader,

$$\mathit{word\_score}(\mathit{word}_i) = a * \mathit{chain\_score}(\mathit{chain}(\mathit{word}_i)) \tag{6}$$

$$\alpha = 1 - (\min[\mathit{dist}(w_{i-1}, w_i), \mathit{dist}(w_i, w_{i+1})] / \mathit{dist}(w_1, w_n)) \tag{7}$$

where *dist*(w_i,w_j) is the number of words that separate two words in the text and *chain*($word_i$) is the chain $word_i$ belongs to. As explained earlier the sentence score is the sum of these word scores normalized with respect to the length of the sentence and the number of chain words it contains.

The second ranking scheme, equation (8), simply gives each word in a sentence the score of the chain to which it belongs,

$$\mathit{word\_score}(\mathit{word}_i) = \mathit{chain\_score}(\mathit{chain}(\mathit{word}_i)) \tag{8}$$

where *chain*($word_i$) is the chain $word_i$ belongs to.

The final ranking scheme is identical to the first scheme, except that if a word occurs in the first paragraph then its score is doubled, thus biasing chain words that occur in the first paragraph. This takes into account the structure of news documents where the first paragraph tends to contain a summary of the article and subsequent articles elaborate and expand the story. So for all the ranking schemes we total the scores for each word in the sentence. This sentence score is then normalised by the number of chain words per unit length.

In the next section we will compare all possible combinations of the scoring and sentence ranking metrics. We will evaluate all these summarization systems and compare the best performing system against several baseline systems.

4 Experimental Methodology

As explained above, we use a task-oriented evaluation methodology to determine the performance of our lexical chain based summarizers, this type of evaluation can be automated and hence more efficient than an intrinsic evaluation that involves the time and effort of a set of human judges. It also provides us with a means of evaluating summary performance on a larger than normal data set of news stories used in the DUC evaluation, i.e. 326 TDT documents and 298 TREC documents [5]. While intrinsic evaluation gauges summary quality directly by rating summary informativeness and coherency, extrinsic evaluation gauges the impact the summary generation procedure has on some task, thus indirectly determining summary quality. Several such tasks have been outlined as useful by TIPSTER [8], such as ad-hoc retrieval, categorization and question answering tasks.

In this paper we use the TDT Story Link Detection Task [2]. TDT is a research initiative that investigates the event-based organisation of news stories in a broadcast news stream. Story Link Detection (SLD) is the pair-wise comparison of stories to establish whether they discuss the same event. Thus for each distinct set of summaries generated (by each system), we evaluate summary quality by observing whether the SLD system can distinguish between on-topic and off-topic document summary pairs. Hence, the hypothesis underlying this type of summary evaluation is that an SLD system will perform well on summaries that have retained the core message of each news story, while it will perform poorly on summaries that in general failed to recognise the central theme of the documents in the data set. Our SLD system is based on an IR vector space model where document similarity is determined using the cosine similarity function [18]. As in the TDT initiative, we evaluate story link detection performance using two error metrics: percentage misses (document pairs that are incorrectly tagged as off-topic) and false alarms (document pairs that are incorrectly tagged as on-topic). A Detection Error Trade-off (DET) graph is then plotted for misses and false alams rates at various similarity thresholds (ranging from 0 to 1) where a DET curve is produced for each set of generated summaries. Optimal SLD performance can then be determined by observing which of these curves lies closest to the origin, i.e. has the lowest miss and false alarm rates.

5 Results

Firstly, we evaluated all possible combinations of the five scoring metrics and three ranking metrics. These combinations are listed in Table1 below.

Table 1. This table contains the system letters assigned to all the possible combinations of the five scoring metrics and the three ranking schemes.

	Chain_score1	Chain_score2	Chain_score3	Chain_score4	Chain_score5
Ranking_1	A	B	C	D	E
Ranking_2	F	G	H	I	J
Ranking_3	K	L	M	N	O

We generated summaries for all fifteen of the systems at summary compression rates of 10, 20, 30, 40, 50 and 60 percent. Each of these summary sets was given as input to the SLD system and DET graphs were produced This graph is indicative of the general trend for all the compression rate. Below we have Figure 1 which shows the best performing systems of the second ranking scheme, where sentence words are given the score of the chain to which they belong. We have left out the results of the two lesser performing metrics for the sake of clarity in the graph.

The results of this experiment lead us to believe that the differences between scoring metrics is very small. However, the best performing sentence ranking scheme across the five scoring metrics is the second ranking scheme, equation (8), and the best performing scoring metric is the second scoring metric, equation (2). These facts are reinforced by the fact that system G is the overall best performing system. (System G is the combination of the best sentence ranking and scoring metrics).

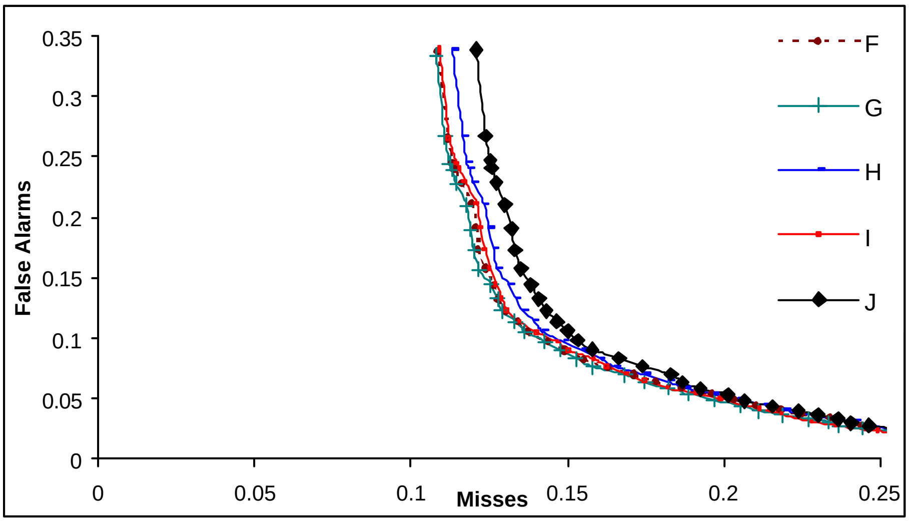

Fig. 1. This DET graph shows the Story Link Detection comparison of the different combinations of scoring and ranking metrics for a compression rate of 50%

We also evaluate our best combination, LexSum, against three baseline systems LEAD, TF-IDF, and RANDOM, using the same evaluation strategy as above. The LEAD system creates summaries from the lead paragraph of each document, since news stories tend to contain a summary of the article in the first paragraph. The TF-IDF system extracts sentences which have high *tf-idf* weights values, where *tf-idf* is a term weighting scheme that is commonly used in IR research [18]. The final baseline extracts sentences at random from the source document and uses these as a summary. We also created a system, B&E that replicates Barzilay and Elhadad's scoring metric. We modified the B&E extraction technique to enable us to generate summaries of different lengths.

We again generated summaries for all the summarisers at summary compression rates of 10, 20, 30, 40, 50 and 60 percent (of the top ranked sentences in the text). Figure 2 is a DET graph illustrating the results for each summarisation system running at 50% compression. Again, this graph is indicative of the general trend for all the compression rates. Both lexical chain systems outperform the baseline systems for all percentages except at 10% where the LEAD performs better. As expected the RANDOM summariser has the worst performance. The fact that lexical chain based summarisers outperform TFIDF, suggests that observing patterns of lexical cohesion is a more accurate means of identifying core themes in documents than using corpus statistics like *tf.idf.* Another observation from these experiments is that B&E's weighting scheme marginally outperforms ours at high false alarm and low miss rates; however this result is not statistically significant.

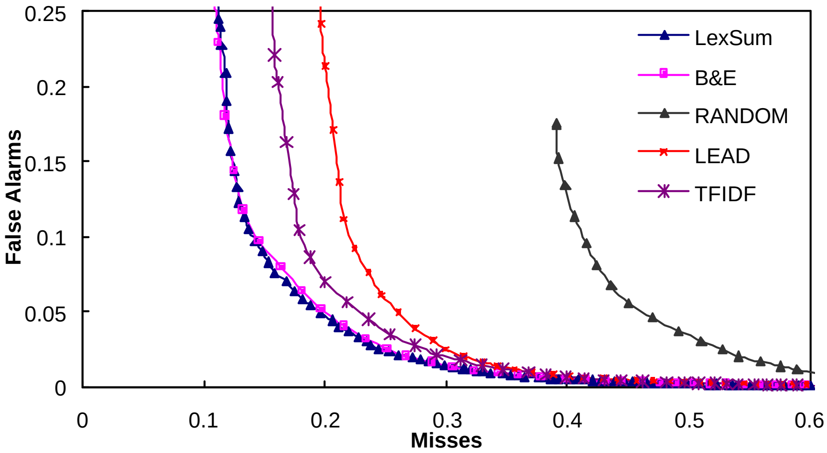

Fig. 2. This DET graph shows the Story Link Detection results of summaries (at a compression rate of 50%)

5 Conclusions and Future Work

In this paper, we have analysed some of the factors that affect lexical chain based summarisation using an extrinsic evaluation methodology. We found that the effect of the weighting scheme has little effect on the summaries. It is likely that both lexical chain based systems are selecting the same sentences, the extent of this trend warrants further investigation. Both chaining systems perform better than the TF.IDF and LEAD systems, justifying the extra computation involved in lexical chaining. We have also shown that combining different scoring and ranking schemes can have an effect on performance, this is only a slight effect at the moment but it may prove useful at finer levels of system granularity. Also, the SLD evaluation method proved to be sensitive enough to show the differences between the baseline systems and the lexical chain based systems. It is our intention to carry out an intrinsic evaluation of the summarisation systems described in this paper and compare these human-deduced summary quality ratings with the results of the automated evaluation presented above.

References

[1] Alemany, L. and Fuentes M., 2003, *Integrating Cohesion and Coherence for Text Summarization.* In the Proceedings of the EACL Student Workshop, 2003.

[2] Allan J., 2002, Introduction to Topic Detection and Tracking, In *Topic Detection and Tracking: Event-based Information Organization*, Kluwer Academic Publishers, pp. 1-16.

[3] Barzilay R. and Elhadad M., 1997, Using Lexical Chains for Summarisation. In *ACL/EACL-97 summarisation workshop.* Pp 10-18, Madrid.

[4] Brunn M., Chali Y., and Pinchak C. 2001,Text summarisation using lexical chains, In *Workshop on Text Summarisation in conjunction with the ACM SIGIR Conference 2001*, New Orleans, Louisiana, 2001.

[5] DUC 2003 `http://www-nlpir.nist.gov/projects/duc/`

[6] Green, S. 1997, *Automatically generating hypertext by computing semantic similarity*, PhD thesis, University of Toronto.

[7] Hearst, M. 1994, Multi-paragraph segmentation of expository text, In *Proceedings of the 32th Annual Meeting of the Association for Computational Linguistics,* 9–16. Las Cruces, New Mexico: Association for Computational Linguistics.

[8] Mani I., House, D., Klein, G., Hirschman, L., Obrst, L., Firmin, T., Chrzanowski, M. and Sundheim, B. 1998, The TIPSTER SUMMAC text summarisation evaluation: Final report. MITRE Technical Report MTR 98w0000138, MITRE.

[9] Miller G.A., Beckwith R., Fellbaum C., Gross, D., and Miller, K. 1990, *Five papers on WordNet.* Technical Report, Cognitive Science Laboratory, 1990.

[10] Morris, J. and Hirst, G. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text, *Computational Linguistics* 17(1): 21-43.

[11] Salton, G., Singhal, A., Mitra, M., and Buckley, C. 1997, Automatic text structuring and summarisation. *Information Processing and Management* 33(2):193–208.

[12] Silber, G. and McCoy, K.. 2000, Efficient Text Summarisation Using Lexical Chains, In *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI'2000).*

[13] Spark-Jones, K. 2001, Factorial Summary Evaluation, In *Workshop on Text Summarisation in conjunction with the ACM SIGIR Conference 2001.* New Orleans, Louisiana.

[14] St. Onge, D. 1995, *Detection and Correcting Malapropisms with Lexical Chains,* M.Sc Thesis, University of Toronto, Canada.

[15] Stairmand, M. 1996, *A Computational Analysis of Lexical Cohesion with Applications in Information Retrieva,.* Ph.D. Dissertation, Center for Computational Linguistics, UMIST, Manchester.

[16] Stokes, N., J. Carthy, A.F Smeaton, SeLeCT: A Lexical Chain-based News Story Segmentation System. To appear in the AI Communications Journal.

[17] TDT Pilot Corpus, `http://www.nist.gov/speech/tests/tdt/`

[18] van Rijsbergen, C.J., Information Retrieval, Butterworths, 1979