News Story Gisting at University College Dublin

**William Doran, Nicola Stokes, Eamonn Newman, John Dunnion,
Joe Carthy, Fergus Toolan.**
Intelligent Information Retrieval Group,
Department of Computer Science,
University College Dublin, Ireland.
{William.Doran, Nicola.Stokes, Eamonn.Newman, John.Dunnion,
Joe.Carthy, Fergus.Toolan}@ucd.ie

Abstract

In this paper we present a machine learning approach to generating very short news story summaries (i.e. no more than 75 bytes long). Our technique uses a decision tree classifier to establish which phrases in a text should be included in the resultant summary. Our ROUGE evaluation results for task 1 (English text summarisation) and task 3 (translated Arabic text summarisation) indicate that this technique is an adequate solution to this problem.

1 Introduction

A gist is a very short summary, ranging in length from a single phrase to a sentence, that captures the essence of a piece of text in much the same way as a title or section heading in a document helps to convey the text's central message to a reader. In this paper, we present our news story gisting system which uses a machine learning technique to combine linguistic, statistical and positional information in order to generate very short news story summaries (i.e. less than 75 bytes long) for the DUC (Document Understand Conference) 2004 evaluation.

Our research has predominantly focused on analysing the lexical cohesive structure of a text using a linguistic technique called lexical chaining. Lexical chaining is a word clustering approach that uses an auxiliary knowledge source like the WordNet taxonomy to identify lexicosemantic associations between words in a text including synonymy (e.g. child/kid), specialisation/generalisation (e.g. apple/fruit), part/whole (e.g. spark plug/engine). It has been successfully used by many researchers to produce extractive summaries (Barzilay, Elhadad, 1997; Silber, McCoy, 2000; Brunn, Chali, Pinchak, 2001; Bo-Yeong, 2002; Alemany, Fuentes, 2003), where lexical chains provide a means of identifying sentences that discuss important themes in a document.

In Section 2, we describe how we have strengthened our lexical cohesion-based gisting approach with additional linguistic (part-of-speech tags), statistical (document and corpus-based term frequencies) and positional (word position) information. Using the DUC 2003 collection as training data, our decision tree classifier attempts to predict the summarisation potential of a word based on a set of features, which we believe help to distinguish between salient and non-informative gist terms in a text. The result of this process is a list of words representing the essence of a news story. A post-processing step then re-orders these terms with respect to their occurrence in the original source text in order to improve gist readability.

The performance of our gisting technique at DUC 2004 for task 1 (very short single document summarisation of English news documents) and task 3 (very short single document summarisation of translated Arabic news documents) is discussed in Section 3. In addition, we also examine the effect on ROUGE scores and system rankings when stopwords are excluded from these calculations. This is followed by a discussion of our overall conclusions.

2 System Overview

As already stated, our gisting method adopts a corpus-based, machine learning approach to the generation of very short news story summaries. More specifically, we use the C5.0 learning algorithm (Quinlan, 1998) to create a decision tree capable of predicting which words in the source text should be included in the resultant gist. In the following subsections we describe our three-step gisting process: 'Feature Assignment', 'Classifier Training' and 'Gist Generation'.

2.1 Word Feature Assignment

In order to create the decision tree classifier, a training set of positive and negative examples must be created. Our training set consists of a collection of words that have been assigned a set of attribute-value pairs. These attributes or features were chosen because of their ability to differentiate between good and bad summary terms in the source text. We used the DUC 2003 corpus as the training data for our classifier, where positive examples of summary terms are provided by the set of manually created summaries for each of the news stories in the corpus, and negative examples are provided by all non-summary words in the source text of each news story. In order to limit the amount of noise in the training collection, we only provide the classifier with feature vectors assigned to content words in each news story and its representative summaries, i.e. nouns, verbs and adjectives.

For each occurrence of a term in a document we calculate the values of the following features:

- The term frequency or *tf* of the word in the document;
- The inverse document frequency or *idf* of the term in an auxiliary news corpus (TDT, 1997);
- The relative position of a word with respect to the start of the document in terms of word distance;
- A lexical cohesion score that measures the lexical cohesive strength of the relationships between a word and the document in which it occurs. Our hypothesis is that if a word has a high cohesion score then it is a useful summary word because it is strongly associated with other important terms in the document;
- The four remaining features are assigned a binary score indicating whether a word is a noun, a verb, or an adjective, or occurs in a noun or proper noun phrase.

Nouns and compound nouns are chained by searching for repetition and lexicosemantic relationships between words in the text. 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 mainly 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.*

Our lexical cohesion score is calculated with the aid of a linguistic technique called lexical chaining. Lexical chaining is a method that clusters words that are semantically similar in a document with the aid of a thesaurus, in our case WordNet. Our chaining algorithm, based on a method described in (Stokes, 2004) identifies 5 types of word relationship (in order of strength): repetition, synonymy, specialisation and generalisation, and words related through paths greater than 1 in WordNet. The first step in the 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 US/NN President/NN Bush/NP where /NN is a noun tag and /NP is a proper noun tag.

Once all lexical chains have been created for a text then a score is assigned to each chained word based on the strength of the chain in which it occurs. More specifically, as shown in the following equation, the chain strength score is the sum of each strength score assigned to each word pair in the chain.

$$Score(chain) = \sum((reps_i + reps_j) * rel(i, j))$$

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. Relationship strengths between chain words are defined as follows, where a repetition relationship is assigned a value of 1.0, a synonym relationship a value of 0.9, specialisation/generalisation and part-whole/whole-part a value of 0.7. Proper nouns chain word scores are assigned 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 lexical cohesion score of a chained word is then the strength score assigned to the chain where the word occurs.

2.2 Training the Classifier

The next step is the training of our decision tree summarisation model using the C5.0 machine-learning algorithm. As already stated, our training data consists of a collection of feature vectors assigned to all content words in the DUC 2003 task 1 gold-standard human-generated summaries and corresponding news story documents. The DUC 2003 evaluation provides four human summaries for each document. Words in the source text that occur in these model summaries are considered to be positive training examples, while document words that do not occur in these summaries are considered to be negative examples. Further use is made of these 4 summaries, where the model is trained to classify a word based on its summarisation potential. More specifically, the appropriateness of a word as a summary term is determined based on the class assigned to it by the decision tree. These classes are ordered from strongest to weakest as follows: 'occurs in 4 summaries', 'occurs in 3 summaries', 'occurs in 2 summaries', 'occurs in 1 summary', 'occurs in none of the summaries'. Therefore, if the classifier predicts that a word will occur in all four of the human-generated summaries, then it is considered to be a more appropriate summary word than a word predicted to occur in only three of the model summaries. This resulted in a total of 103267 training cases, where 5762 instances occurred in one summary, 1791 in two, 1111 in three, 726 in four, and finally 93877 instances were negative. A decision tree classifier was then produced by the C5.0 algorithm based on this training data. To gauge the accuracy of this classifier we used a training/test data split of 90%/10%, and found that on this test set the classifier had a precision of 63% and recall of 20%. Recall, in this context, is the total number of true positives returned by the classifier divided by the total number of true positives and false negatives, while precision is the total number of true positives returned by the classifier divided by the total number of true positives and false positives.

2.3 Generating the News Story Gist

Our 'DUC 2003 trained' decision tree was used to generate news gists for tasks 1 and 3 in the 2004 DUC evaluation. We ran the classifier on the news document clusters defined for each of these tasks, where the top 10 positively classified words were included in the resultant summary with precedence given to those words classified as occurring in four model summaries, then three and so on. In the case where the classifier could not return the required number of words (on average it returned 4.32 words per document), we then looked at the aggregate feature-weight scores assigned to each word and used the top ranked words according to this score to 'pad out' the gist to its required length. This aggregate score is simply the sum of the normalised feature values defined for each word. In addition, the algorithm also favours the addition of high scoring terms that occur in compound noun phrases with any of the 'classifier' gist words. The intuition behind this step is that if we add more context to a word by representing it in the gist in its phrasal form then we can improve the readability and quality of the resultant gist. This raises the issue of a trade-off between context and content, because DUC gists must be no more than 75 bytes long and as we increase the number of compounds we also reduce the number of content words that the classifier has predicted. To minimise this effect we employ 'severe' pruning heuristics that ensure that only the most salient words are used to create compounds in the gist. For example, if we have an adjective-noun compound, we remove the adjective if it cannot be mapped to a noun, e.g. 'magnetic field' will be accepted but 'large field' will not. In addition, our algorithm uses a list of common first names and family names to identify references to people in the gists. These names are then pruned and only the surnames are included in the gists, i.e. rebel leader *Abdullah* Ocalan => rebel leader Ocalan.

Once a list of important summary words has been identified in this manner, a final post-processing step re-orders these gist words with respect to their occurrence in the text in order to further improve the readability of the gist. The importance of this step is illustrated in the following example generated by our system:

- "extradition, protests, rebel leader Ocalan, Kurds, Turkey, Italy, Rome"

which is re-ordered to read as follows

- "Turkey Italy extradition rebel leader Ocalan Kurds protests Rome"

where the two closest gold standard gists are

- "Ocalan supporters demonstrate in Rome to block extradition to Turkey"
- "Turkey pressures Italy to extradite PKK leader Abdullah Ocalan".

3 DUC Evaluation Results

In this section we present our DUC 2004 evaluation results. We report on both the official results determined by the organizers of the workshop and also the results of our own experiments on the DUC data set with the ROUGE evaluation metric (Lin, Hovy, 2003). In particular, we examine the effect of excluding stopwords when calculating ROUGE scores. We also comment on the contribution of different features, described in Section 2.1, to the performance of the classifier with respect to summary quality.

3.1 Official DUC Results

We participated in task 1 (very short single document summaries) and task 3 (very short single document summaries of translated documents). We submitted three runs for task 1 and two runs for task 3. Each of our systems returned ten word summaries, where we allowed the ROUGE metric to prune the gists to the required 75 bytes in all runs except one. Our five submissions (including run numbers) were:

- **Task 1, Run 130**: This system returned a 75-byte summary, we pruned our 10-word summary by removing words with low term-frequencies in the document until the target length (or just below it) was reached.
- **Task 1, Run 131**: This system returned the 10 'strongest' gisting terms determined by our system.
- **Task 1, Run 132**: This system returned a list of 10 content words from the sentences that had the most gist words returned by the classifier.
- **Task 3, Run 133**: This system returned 10-word summaries of the concatenated machine translations (i.e. IBM and ISI translations) of the document in the same manner as Run 132.
- **Task 3 Run 134**: This system returned 10-word summaries of the manually translated documents in the same manner as Run 132.

The official ROUGE evaluations of task 1 and 3 were carried out on all submissions using summary lengths of 75 bytes. The summaries were stemmed using the Porter stemming algorithm; however, stopwords were not removed. The format of the evaluation was based on six scoring metrics: ROUGE-1 (R1), ROUGE-2 (R2), ROUGE-3 (R3), ROUGE-4 (R4), ROUGE-LCS (RL) and ROUGE-W (RW). The first four metrics are based on n-gram matches between the peer summary and the model summaries. ROUGE-LCS is based on a longest common sub-string between the peer and the models, and the ROUGE-W is a weighted version of the LCS measure. No overall performance metric was decided upon before the workshop.

In task 1 there were 39 participants, and in task 3 there were 11 participants for the machine translated summarisation task and 10 for the manually translated summarisation task. We did not participate in the third part of task 3 which provided systems with additional relevant documents for the machine translated summarisation task. Table 1 and 2 contain a summary of the final scores and ranks for each of our submitted runs for these tasks.

Peer	R1	R2	R3	R4	RL	RW
130	0.217	0.024	0.003	0.000	0.167	0.097
131	0.219	0.025	0.003	0.000	0.176	0.102
132	0.154	0.023	0.005	0.001	0.129	0.077
133	0.203	0.017	0.003	0.001	0.160	0.095
134	0.259	0.047	0.011	0.002	0.220	0.129

Table 1: Official DUC ROUGE scores for task 1 and task 3.

Peer	R1	R2	R3	R4	RL	RW
130	6	37	37	38	11	15
131	3	34	36	37	6	7
132	32	38	34	34	34	35
133	3	11	10	10	7	7
134	1	7	8	9	1	1

Table 2: Official DUC overall ranks for task 1 (out of 39 systems) and task 3 (run 133 out of 11 systems and run 134 out of 10 systems)

From these tables it can be seen that we performed quite well in ROUGE1, ROUGE-LCS and ROUGE-W. Unfortunately, we performed poorly on the other ROUGE metrics. A number of conclusions can be drawn from these results. Firstly, for task 1 our best performing system is 131. This result is somewhat surprising since in run 130 we attempted to prune less useful words from the summary using certain heuristics, while for system 131 we let Rouge prune the summary to 75 bytes by stripping off the extra bytes from the end of the summary.

In addition, from the performance of system 132, we can conclude that a sentence extraction-based approach does not perform as well with respect to the ROUGE metric. With the exception of the lead sentence, it seems that it is very difficult to find a single sentence in the text that captures the focus of a news document. This observation has also been made by other gisting researchers who advocate an abstractive rather than extractive gisting approaches (Witbrock, Mittal, 1999; Banko et al., 2000; Zajic, Dorr, 2002). However, many of the systems that outperformed ours produced summaries that used exactly the same vocabulary as the lead sentence.

For task 3, differences in the ROUGE scores (Table 1) for our machine translated (system 133) and manually translated (system 134) submissions suggest that our gisting technique is severely affected by 'noisy' machine translated documents. However, for run 133 we concatenated the IBM and ISI translations for each document before generating a gist. Hence, it is unclear whether gist quality was degraded by this concatenation process or by a lack of robustness on the part of our system. Further experiments will need to be run to determine this. The scoring trends observed for task 1 were also present in the task 3 results, where our system performed well in ROUGE-1, ROUGE-LCS and ROUGE-W and poorly in the other metrics.

Based on an empirical observation of the data, we were able to gain some insight into why our system performed poorly on ROUGE-2, ROUGE-3 and ROUGE-4. By comparing our summaries with those of systems that performed well with respect to these metrics, we have observed that our post-processing heuristics had, on many occasions, not ordered gist words as well as was expected. Reasons for this discrepancy are still under investigation. We also noticed that in some cases our noun pruning heuristic helped to degrade performance by removing important first names from noun phrases like 'Hurricane *Mitch*' and '*Margaret* Thatcher' that were included in the model summaries.

3.2 Impact of Stopwords on the Rouge metric

We carried out further experiments to gauge the effect of stopwords on the ROUGE metric. There are two main extractive/abstractive approaches to the gisting task: either a list of words is returned by the system (as in the case of our approach) or a sentence or clause is returned as the gist. In the case of the former the system tries to return as many useful content words and then let the reader try to 'interpret' the gist, whereas systems based on the later approach try to return a more comprehensible summary often at the expense of useful content words. Hence, the inclusion/exclusion of stopwords when calculating ROUGE scores can favour one approach over the other depending on which words are included in the calculation. Tables 3 and 4 contain our system scores and ranks for the ROUGE metric when stopwords are removed.

Peer	R1	R2	R3	R4	RL	RW
130	0.293	0.039	0.006	0.001	0.226	0.141
131	0.298	0.046	0.009	0.002	0.239	0.148
132	0.208	0.038	0.009	0.002	0.174	0.113
133	0.270	0.028	0.004	0.001	0.212	0.134
134	0.349	0.075	0.022	0.005	0.287	0.180

Table 3: ROUGE scores for task 1 and 3 runs when stopwords are excluded.

Peer	R1	R2	R3	R4	RL	RW
130	3	29	28	37	4	5
131	1	23	27	27	2	3
132	29	32	26	25	31	32
133	1	10	10	9	2	4
134	1	6	6	4	1	1

Table 4: System rankings for task 1 and 3 runs when stopwords are excluded.

Comparing these tables to Tables 1 and 2, we can see that the removal of stopwords has a significant effect on some of our task 1 and 3 ROUGE scores. In particular, our ROUGE-1, ROUGE-L and ROUGE-W rankings improve quite considerable; however, in contrast there is no major increase in our ROUGE-2, ROUGE-3 and ROUGE-4 scores. With respect to the ROUGE scores of the other participants, we observed that systems that returned an excessive number of stopwords in their gists dropped several rank positions when ROUGE-1 scores were calculated without stopwords. A possible solution to this anomaly is to only base ROUGE-1 score calculations on the overlap of content words between the system gist and the set of gold standard gists for a particular news story. However, we believe it is still advantageous to base all other ROUGE-N calculations on stopword and content word overlap, because these scores can then be used to estimate the comprehensibility of a gist, i.e. if n-grams containing stopwords overlap with model summary n-grams also containing stopwords then this implies that the readability of the machine generated gist is good. If ROUGE scores were calculated in this way then this would ensure that gists containing many stopwords would not be assigned inflated scores by the ROUGE-1 metric and ROUGE-N scores would correctly favour systems that produce readable, accurate and grammatically correct gists.

3.3 Feature Importance

Further experiments on the DUC collection were also carried out in order to gauge the impact of certain word features on summary quality. As stated in Section 2.1, our decision tree classifier was trained on a combination of eight distinct linguistic, statistical and positional word attributes: a lexical cohesion score, part-of-speech information, word position, term frequency *tf* and inverse document frequency *idf*. We used the task 1 evaluation documents to estimate the importance of these features by observing the effect of removing features on summary quality (i.e. the ROUGE-1 score) and the accuracy of the decision tree classifier (i.e. the recall and precision values provided by the C5.0 algorithm defined in Section 2.1). Table 5 shows the results of our feature impact analysis.

Feature	Recall	Precision	Rouge1
All	19.0	68.9	0.3054
No position	11.2	73.8	0.2813
No *tf*	3.1	59.9	0.2538
No *idf*	15.2	69.0	0.2900
No Lex	17.8	67.6	0.3018
No Noun	19.4	68.3	0.3080
No Verb	18.8	68.0	0.3050
No Adj	19.1	68.9	0.3051
No Noun Compound	18.6	67.6	0.3002

Table 5: Shows the impact of removing features on the precision/recall figures for 103000 training cases with 87000 test cases.

From this table we a can see that, term frequency, word position and *idf* are the features that have the greatest impact on the quality of the summary and the accuracy of the classifier. The features that have the least impact are those related to the part-of-speech tag information. Interestingly, our lexical cohesion score also seems to add little to the overall decision tree classification process.

4 Conclusions

In this paper we have presented the details of our first submission to the DUC workshop. We presented a detailed outline of our system based on a decision tree classifier generated by the C5.0 machine-learning algorithm. We submitted runs for both task 1 and task 3. Overall, our ROUGE-1, ROUGE-LCS and ROUGE-W scores were consistently high for both tasks. However, we did not perform as well on the other ROUGE-N scoring metrics.

Our experiments on the impact of stopwords on ROUGE calculations showed that it has a varying effect on ROUGE-1 scores depending on whether stopwords are included in the system gists being evaluated. We also ran additional experiments to determine which word features were the most useful predictors of salient gist terms. We found that term frequency and word position were the best predictors of appropriate summary words.

Acknowledgements

The support of Enterprise Ireland is gratefully acknowledged.

References

Alemany L., M. Fuentes. 2003. *Integrating cohesion and coherence for automatic summarization.* In the Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-03).

Banko M., V. Mittal, M. Witbrock. 2000. *Generating Headline-Style Summaries*. In the Proceedings of the Association for Computational Linguistics (ACL-00).

Barzilay R., M. Elhadad. 1997. *Using Lexical Chains for Text Summarization*. In the Proceedings of (ACL-97/EACL-97), Workshop on Intelligent Scalable Text Summarization, pp. 10-17.

Bo-Yeong Kang. 2002. *Text Summarization through Important Noun Detection Using Lexical Chains.* M.S. Thesis, Kyungpook National University.

Brunn M., Y. Chali, C.J. Pinchak. 2001. *Text Summarization Using Lexical Chains*. In the Proceedings of the Document Understanding Conference (DUC-2001), pp. 135 – 140.

Lin C-Y, Hovy E. 2003. *Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics.* In Proceedings of HLT-NAACL-2003.

Quinlan R. 1998. C5.0: *An Informal Tutorial.* RuleQuest, http://www/rulequest.com/see5-unix.html

Silber H.G., K.F. McCoy. 2002. *Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization*. Computational Linguistics, Vol. 28, No. 4, pp. 487-496.

Stokes N. 2004. *Applications of Lexical Cohesion Analysis in the Topic Detection and Tracking domain.* Ph.D. thesis, Department of Computer Science, University College Dublin.

TDT Pilot Study Corpus: www.nist.gov/speech/tests/tdt/

Witbrock M., V. Mittal. 1999. *Ultra-Summarisation: A Statistical approach to generating highly condensed non-extractive summaries.* In the Proceedings SIGIR-99, pp. 315-316.

Zajic D., B. Dorr. 2002. *Automatic headline generation for newspaper stories.* In the Proceedings of the ACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2002).