Broadcast News Gisting using Lexical Cohesion Analysis.

Nicola Stokes[1], Eamonn Newman[1], Joe Carthy[1], Alan F. Smeaton[2],

[1] Intelligent Information Retrieval Group,
Department of Computer Science, University College Dublin, Ireland.
{Nicola.Stokes, Eamonn.Newman, Joe.Carthy}@ucd.ie
[2] Centre for Digital Video Processing, Dublin City University, Ireland.
Alan.Smeaton@computing.dcu.ie

Abstract. In this paper we describe an extractive method of creating very short summaries or gists that capture the essence of a news story using a linguistic technique called lexical chaining. The recent interest in robust gisting and title generation techniques originates from a need to improve the indexing and browsing capabilities of interactive digital multimedia systems. More specifically these systems deal with streams of continuous data, like a news programme, that require further annotation before they can be presented to the user in a meaningful way. We automatically evaluate the performance of our lexical chaining-based gister with respect to four baseline extractive gisting methods on a collection of closed caption material taken from a series of news broadcasts. We also report results of a human-based evaluation of summary quality. Our results show that our novel lexical chaining approach to this problem outperforms standard extractive gisting methods.

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 texts central message to a reader. In digital library and multimedia applications that deal with streams of continuous unmarked data tasks like text segmentation, document classification and gisting are prerequisites for the successful organisation and presentation of these data streams to users.

In this paper, we focus on creating news story gists for streams of news programmes used in the DCU Físchlár-News-Stories system [1]. In its current incarnation the Físchlár-News-Stories system segments video news streams using audio and visual analysis techniques. Like all real-world applications these techniques will at times place erroneous story boundaries in the resultant segmented video streams. In addition, since the closed caption material accompanying the video is generated live during the broadcast, a time lag exists between the discussion of the piece of news in the audio stream and the appearance of the teletext in the video stream. Consequently, segmentation errors will be present in the closed caption stream, where for example the end of one story might be merged with the beginning of the next story. Previous

work in this area undertaken at the DUC summarisation workshops [2] and by other research groups has predominantly focussed on generating gists from clean data sources such as newswire [3], thus avoiding the real issue of developing techniques that can deal with the erroneous data that underlies this problem.

In Section 2 of this paper, we will discuss our approach to gisting which is based on a linguistic technique called lexical chaining, where a lexical chain is a sequence of semantically related words in a text e.g. {boat, ship, yacht, rudder, hull, bow}. The relationship between lexical cohesion analysis and lexical chaining is tackled in Section 2, while the exact details of our gisting system, the LexGister, and our novel approach to generating lexical chains in a news domain is described in Section 3. In Sections 4, 5 and 6, we describe the results of an intrinsic and automatic evaluation of our system generated gists on a collection of closed caption material taken from an Irish television news programme. We contrast these results with the performance of four baseline systems: a baseline lexical chaining approach, a *tf.idf* weighting approach, a 'lead' sentence approach, and a random extraction approach to the gisting task. Finally in Section 7, we review related title generation approaches and comment on some directions for future work.

2 Lexical Cohesion and Lexical Chaining

When reading any text it is obvious that it is not merely made up of a set of unrelated sentences, but that these sentences are in fact connected to each other in one of two ways cohesion and coherence. Lexical cohesion is the textual characteristic responsible for making the sentences of a text seem 'to hang together' [4], while coherence refers to the fact that 'there is sense in the text' [4].

Obviously coherence is a semantic relationship and needs computationally expensive processing for identification; however, cohesion is a surface relationship and is hence more accessible. Cohesion can be roughly classified into three distinct classes, *reference*, *conjunction* and *lexical cohesion* [5]. Conjunction is the only class, which explicitly shows the relationship between two sentences, "Mary spoke to John and he agreed with her view of the situation". Reference and lexical cohesion on the other hand indicate sentence relationships in terms of two semantically identical or related words. In the case of reference, pronouns are the most likely means of conveying referential meaning. For example, in the following sentences, "John was famished. He hadn't eaten all day", the pronoun *he* will only be understood by the reader if they refer back to the first sentence. Lexical cohesion on the other hand arises from the selection of vocabulary items and the semantic relationships between them. For example, "John went to the *supermarket* and bought some *food* for *dinner*. He also chose a nice bottle of red *wine* to accompany his *fillet steak*." In this case cohesion is represented by the semantic relationship between the lexical items *supermarket*, *food*, *dinner, wine,* and *fillet steak*. For automatic identification of these relationships it is far easier to work with lexical cohesion than reference since more underlying implicit information is needed to discover the relationship between the pronoun in the second sentence and the word it references.

Here are a number of examples taken from CNN news transcripts that illustrate the five types of lexical cohesion as defined by Halliday [5] that are present in text:

- **Repetition** occurs when a word form is repeated again in a later section of the text e.g. "In *Gaza*, though, whether the Middle East's old violent cycles continue or not, nothing will ever look quite the same once Yasir Arafat come to town. We expect him here in the *Gaza Strip* in about an hour and a half, crossing over from Egypt".
- **Repetition through synonymy** occurs when words share the same meaning but have two unique syntactical forms. "Four years ago, it passed a domestic violence act allowing *police*, not just the victims, to press charges if they believe a domestic beating took place. In the past, *officers* were frustrated, because they'd arrive on the scene of a domestic fight, there'd be a clearly battered victim and yet, frequently, there'd be no one to file charges."
- **Word association through specialisation/generalisation** occurs when a specialised/generalised form of an earlier word is used. "They've put a possible *murder weapon* in O.J. Simpson's hands; that's something that no one knew before. And it shows that he bought that *knife* more than a month or two ahead of time and you might, therefore, start the theory of premeditation and deliberation."
- **Word association through part-whole/whole-part relationships** occurs when a part-whole/whole-part relationship exists between two words e.g. '*committee*' is made up of smaller parts called '*members*'. "The Senate Finance *Committee* has just convened. *Members* had been meeting behind closed doors throughout the morning and early afternoon."
- **Statistical associations between words** occur when the nature of the association between two words cannot be defined in terms of the above relationship types. e.g. *Osama bin Laden* and *the World Trade Centre*.

One method of exploring the lexical cohesive relationships between words in a text is to build a set of lexical chains for that text. As already stated lexical chains are clusters of semantically related words, where in most cases these words are nouns. In their seminal paper on lexical chaining, Morris and Hirst [4] showed how these word clusters could be used to explore the discourse structure of a text. Since then lexical chains have be used to address a variety of NLP and IR problems including hypertext construction [6], automatic document summarization [7, 8, 9, 10], the detection of malapropisms within text [11], as an IR term weighting and indexing strategy [12, 13], and as a means of segmenting text into distinct blocks of self-contained text [14, 15]. The focus of the research in this paper is to create gists for news stories based on a lexical cohesive analysis of each story provided by a set of lexical chains.

Most lexical chaining research has involved solving NLP/IR problems in a news story domain, using an online thesaurus (in most cases WordNet [16]) to capture the lexical cohesive relationships listed above. However, there are two problems associated with this approach to chaining. Firstly, WordNet does not keep an up-to-date repository of 'everyday' proper nouns like company names and political figures. The effect of this is that these parts of speech cannot participate in the chaining process and valuable information regarding the entities in a new story is ignored. In Section 3, we describe a novel lexical chaining algorithm that addresses this problem by building noun and proper noun-based lexical chains for each news story.

The second problem associated with previous lexical chaining methods relates to the omission of statistical word associations during the chaining process, which represent a large portion of lexical cohesive relationships in text. To address this problem our chaining algorithm uses co-occurrence statistics generated from an auxiliary corpus (TDT1 corpus [17]) using the log-likelihood association metric. These additional lexical cohesive relationships (amounting to 3,566 nouns that have an average of 7 collocates each) provide our chaining algorithm with many 'intuitive' word relationships which are not supported in the WordNet taxonomy like the relationships between the following set of words, {abuse, victim, allegation, abuser}. In Section 5, we describe the results of an experiment that verifies that these enhancements, when added to a basic chaining algorithm improve the performance of our gisting system described in the following section.

3 The LexGister

In this section we present our news gister, the LexGister system. This system takes a closed caption news story and returns a one-sentence gist or headline for the story. The system consists of three components a 'Tokeniser', a 'Chainer' which creates lexical chains, and an 'Extractor' that uses these chains to determine which sentence best reflects the content of that story.

3.1 The Tokeniser

The objective of the chain formation process is to build a set of lexical chains that capture the cohesive structure of each news story in the data set. Before work can begin on lexical chain identification, each segment or news story is processed by a part-of-speech tagger [18]. Once the nouns in the text have been identified, morphological analysis is then performed on these nouns; all plurals are transformed into their singular state, adjectives pertaining to nouns are nominalised and all sequences of words that match grammatical structures of compound noun phrases are extracted. This idea is based on a simple heuristic proposed by Justeson and Katz [19], which involves scanning part-of-speech tagged texts for patterns of adjacent tags that commonly match proper noun phrases like 'White House aid', 'PLO leader Yasir Arafat', and WordNet noun phrases like 'act of god', 'arms deal', and 'partner in crime'. This process also helps to improve the accuracy of the lexical chaining algorithm by removing ambiguity from the text. For example, consider the phrase 'New York Times' where each individual word differs in meaning to the phrase as a whole.

In general news story proper noun phrases will not be present in WordNet, since keeping an up-to-date repository of such words is a substantial and never ending problem. However, as already stated, any remaining proper nouns are still useful to the chaining process since they provide a further means of capturing lexical cohesion in the text though repetition relationships.

One problem with compound proper noun phrases is that they are less likely to have exact syntactic repetitions elsewhere in the text. Hence, we introduce into our lexical chaining algorithm a fuzzy string matcher that looks first for full syntactic match (*U.S_President* ⇔ *U.S_President*), then partial full-word match (U.S_*President* ⇔ *President*_Bush) and finally a 'constrained' form of partial word match between the two phrases (*cave*_dwellers ⇔ *cave*rs). In summary then, the Tokeniser produces tokenised text consisting of noun and proper noun phrases including information on their location in the text i.e. sentence number. This is then given as input to the next step in the gisting process, the lexical chainer.

3.2 The Lexical Chainer

The aim of the Chainer is to find relationships between tokens (nouns, proper nouns, compound nouns, nominalized adjectives) in the data set using the WordNet thesaurus and a set of statistical word associations, and to then create lexical chains from these relationships with respect to a set of chain membership rules. The chaining procedure is based on a single-pass clustering algorithm, where the first token in the input stream forms the first lexical chain and each subsequent token is then added to an existing chain if it is related to at least one other token in that chain by any lexicographical or statistical relationships.

A stronger criterion than simple semantic similarity is imposed on the addition of a phrase to a chain, where a phrase must be added to the most recently updated and strongest[1] related chain. In addition the distance between the two tokens in the text must be less than a certain maximum number of words, depending on the strength of the relationship i.e. stronger relationships have larger distance thresholds. These system parameters are important for two reasons. Firstly, these thresholds lessen the effect of spurious chains, which are weakly cohesive chains containing misidentified word associations due to the ambiguous nature of the word forms i.e. associating *gas* with *air* when *gas* refers to a *petroleum* is an example of misidentification. The creation of these sorts of chains is undesirable as they add noise to the gisting process described in the next section.

In summary then our chaining algorithm proceeds as follows: if an 'acceptable' relationship exists between a token and any chain member then the token is added to that chain otherwise the token will become the seed of a new chain. This process is continued until all keywords in the text have been chained. As previously stated our novel chaining algorithm differs from previous chaining attempts [6-12, 15] in two respects:

- It incorporates genre specific information in the form of statistical word associations.
- It acknowledges the importance of considering proper nouns in the chaining process when dealing with text in a news domain.

[1] Relationship strength is ordered from strongest to weakest as follows: repetition, synonymy, generalisation/specialisation and whole-part/part-whole, and finally statistical word association.

In the next section we detail how the lexical chains derived from a news story can be used to create a headline summarising the content of that story.

3.3 The Extractor

The final component in the LexGister system is responsible for creating a gist for each news story based on the information gleaned from the lexical chains generated in the previous phase. The first step in the extraction process is to identify the most important or highest scoring proper noun and noun chains. This step is necessary as it helps to hone in on the central themes in the text by discarding cohesively weak chains. The overall cohesive strength of a chain is measured with respect to the strength of the relationships between the words in the chain. Table 1 shows the strength of the scores assigned to each cohesive relationship type participating in the chaining process.

Table 1. Relationship scores assigned to chain words when calculating a chain score.

Relationship Type	Relationship Score
Repetition	1
Synonymy	0.9
Hyponymy, Meronymy, Holonymy, and Hypernymy	0.7
Path lengths greater than 1 in WordNet	0.4
Statistical Word Associations	0.4

The chain weight, *score*(*chain*), then becomes the sum of these relationship scores, which is defined more formally as follows:

$$score(chain) = \sum_{i=1}^{n} reps(i) + rel(i, j) \quad (1)$$

where *i* is the current chain word in a chain of length *n*, *reps*(*i*) is the number of repetitions of term *i* in the chain and *rel*(*i*,*j*) is the strength of the relationship between term *i* and the term *j* where *j* was deemed related to *i* during the chaining process. For example, the chain {hospital, infirmary, hospital, hospital} would be assigned a score of [*reps*(hospital) + *rel*(hospital, infirmary) + *reps*(infirmary) + *rel*(infirmary, hospital)] = 5.8, since 'infirmary' and 'hospital' are synonyms. Chain scores are not normalised, in order to preserve the importance of the length of the chain in the *score*(*chain*) calculation. Once all chains have been scored in this manner then the highest scoring proper noun chain and noun chain are retained for the next step in the extraction process. If the highest score is shared by more than one chain in either chain type then these chains are also retained.

Once the key noun and proper noun phrases have been identified, the next step is to score each sentence in the text based on the number of key chain words it contains:

$$score(sentence) = \sum_{i=1}^{n} score(chain)_i \quad (2)$$

where $score(chain)_i$ is zero if word i in the current sentence of length n does not occur in one of the key chains, otherwise $score(chain)_i$ is the score assigned to the chain where i occurred.

Once all sentences have been scored and ranked, the highest ranking sentence is then extracted and used as the gist for the news article[2]. This final step in the extraction process is based on the hypothesis that the key sentence in the text will contain the most key chain words. This is analogous to saying that the key sentence should be the sentence that is most cohesively strong with respect to the rest of the text. If it happens that more than one sentence has been assigned the maximum sentence score then the sentence nearest the start of the story is chosen, since lead sentences in a news story tend to be better summaries of its content. Another consideration in the extraction phrase is the occurrence of dangling anaphors in the extracted sentence e.g. references to pronoun like 'he' or 'it' that cannot be resolved within the context of the sentence. In order to address this problem we use a commonly used heuristic that states that if the gist begins with a pronoun then the previous sentence in the text is chosen as the gist. We tested the effect of this heuristic on the performance of our algorithm and found that the improvement was insignificant. We have since established that this is the case because the extraction process is biased towards choosing sentences with important proper nouns, since key proper noun chain phrases are considered. The effect of this is an overall reduction in the occurrence of dangling anaphors in the resultant gist. The remainder of the paper will discuss in more detail performance issues relating to the LexGister algorithm.

4 Evaluation Methodology

Our evaluation methodology establishes gisting performance using manual and automatic methods. The automatic evaluation is based on the same framework proposed by Witbrock and Mittal [3], where recall, precision and the F measure are used to determine the similarity of a gold standard or reference title with respect to a system generated title. In the context of this experiment these IR evaluation metrics are defined as follows:

- **Recall** (***R***) is the number of words that the reference and system titles have in common divided by the number of words in the reference title.
- **Precision** (***P***) is the number of words that the reference and system titles have in common divided by the number of words in the system title.
- **F measure (*F1*)** is the harmonic mean of the recall and precision metrics.

$$F1 = \frac{2(R*P)}{R+P} \tag{3}$$

In order to determine how well our lexical chaining-based gister performs the automatic part of our evaluation compares the recall, precision and F1 metrics of four

[2] At this point in the algorithm it would also be possible to generate longer-style summaries by selecting the top n ranked sentences.

baseline extractive gisting systems with the LexGister. A brief description of the techniques employed in each of these systems is now described:

- A baseline lexical chaining extraction approach (**LexGister(b)**) that works in the same manner as the LexGister system except that it ignores statistical associations between words in the news story and proper nouns that do not occur in the WordNet thesaurus.
- A *tf.idf* [20] based approach (**TFIDF**) that ranks sentences in the news story with respect to the sum of their *tf.idf* weights for each word in a sentence. The *idf* statistics were generated from an auxiliary broadcast news corpus (TDT1 corpus [17]).
- A lead sentence based approach (**LEAD**) that in each case chooses the first sentence in the news story as its gist. In theory this simple method should perform well due to the pyramidal nature of news stories i.e. the most important information occurs at the start of the text followed by more detailed and less crucial information. In practice, however, due to the presence of segmentation errors in our data set, it will be shown in Section 5 that a more sophisticated approach is needed.
- A random approach (**RANDOM**) that randomly selects a sentence as an appropriate gists for each news story. This approach represents a lower bound on gisting performance for our data set.

Since the focus of our research is to design a robust technique that can gist on error prone closed caption material we manually annotated 246 RTÉ Irish broadcast news stories with titles. These titles were taken from the www.rte.ie/news website and map onto the corresponding closed caption version of the story, and so represent a gold standard set of titles for our news collection. The results discussed in Section 5 were generated from all 246 stories. However, due to the overhead of relying on human judges to rate gists for all of these news stories we randomly selected 100 LexGister gists for the manual part of our evaluation.

Although the F measure and the other IR based metrics give us a good indication of the quality of a gist in terms of its coverage of the main entities or events mentioned in the gold standard title, a manual evaluation involving human judges is need to consider other important aspects of gist quality like readability and syntax. We asked six judges to rate LexGister's titles using five different quality categories ranging from 5 to 1 where 'very good = 5', 'good = 4', 'ok = 3', 'bad = 2', and 'very bad = 1'. Judges were asked to read the closed caption text for a story and then rate the LexGister headline based on its ability to capture the focus of the news story. The average score for all judges over each of the 100 randomly selected titles is then used as another evaluation metric, the results of which are discussed in Section 6. This simple scoring system was taken from another title evaluation experiment conducted by Jin and Hauptmann [21].

5 Automatic Evaluation Results

As described in Section 4 the recall, precision and F1 measures are calculated based on a comparison of the 246 generated news titles against a set of reference titles taken from the RTÉ news website. However, before the overlap between a system and reference headline for a news story is calculated both titles are stopped and stemmed using the standard InQuery stopword list [33] and the Porter stemming algorithm [34]. The decision to stop reference and system titles before comparing them is based on the observation that some title words are more important than others. For example if the reference title is 'Government still planning to introduce the proposed anti-smoking law' and the system title is 'The Vintners Association are still looking to secure a compromise' then they shares the words 'the', 'still', and 'to', then it will have successfully identified 3 out of the 9 words in the reference title, resulting in misleadingly high recall (0.33) and precision (0.3) values. Another problem with automatically comparing reference and system titles is that there may be instances of morphological variants in each title, like 'introducing' and 'introduction', that without the uses of stemming will make titles appear less similar than they actually are.

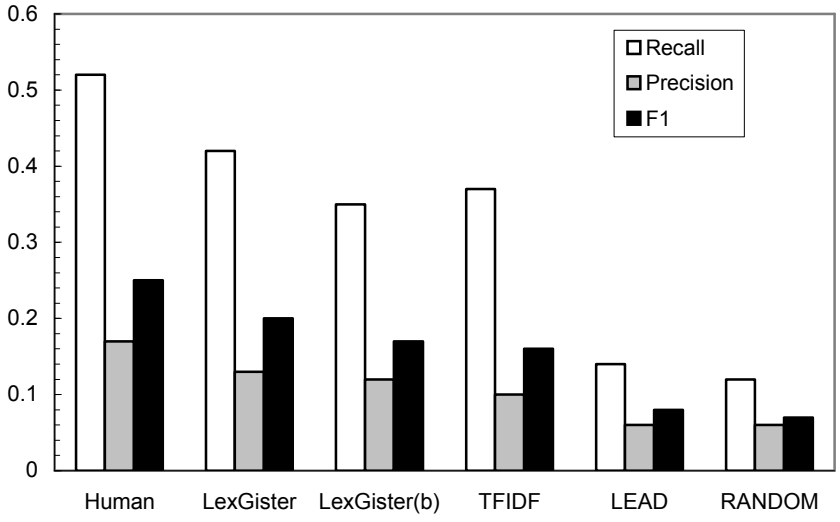

Fig. 1. Recall, Precision and F1 values measuring gisting performance for 5 distinct extractive gisting systems and a set of human extractive gists.

Figure 1 shows the automatic evaluation results, using the stopping and stemming method, for each of our four extractive gisting methods discussed in Section 3. For this experiment we also asked a human judge to extract the sentence that best represented the essence of each story in the test set. Hence, the F1 value 0.25 achieved by these human extracted gists represents an upper bound on gisting performance. As expected our lower bound on performance, the RANDOM system, is the worst per-

forming system with an F1 measure of 0.07. The LEAD sentence system also performs poorly (F1 0.08), which helps to illustrate that a system that simply chooses the first sentence in this instance is not an adequate solution to the problem. A closer inspection of the collection shows that 69% of stories have segmentation errors which accounts for the low performances of the LEAD and RANDOM gisters. On the other hand, the LexGister outperforms all other systems with an F1 value of 0.20. A breakdown of this value shows a recall of 0.42, which means that on average 42% of words in a reference title are captured in the corresponding system gist generated for a news story. In contrast, the precision value for the LexGister is much lower where only 13% of words in a gist are reference title words. The precision values for the other systems show that this is a characteristic of extractive gisters since extracted sentences are on average two thirds longer than reference titles. This point is illustrated in the follow example where the recall is 100% but the precision is 50%, in both cases stopwords are ignored.

- **Gist:** "The world premier of the Veronica Guerin movie took place in Dublin's Savoy Cinema, with Cate Blanchett in the title role."
- **Reference Title:** "Premier of Veronica Guerin movie takes place in Dublin".

This example also shows that some form of sentence compression is needed if the LexGister were required to produce titles as opposed to gists, which would in turn help to increase the recall of the system. However, the high precision of the LexGister system verifies that lexical cohesion analysis is more adept at capturing the focus of a news story than a statistical-based approach using a *tf.idf* weighting scheme. Another important result from this experiment is the justification of our novel lexical chaining algorithm discuss in Section 3.2 that includes statistical word associations and proper nouns not occurring in WordNet in the chaining process. Figure 1 illustrates how the LexGister system (F1 0.20) outperforms the baseline version, LexGister(b), using a less sophisticated lexical chaining algorithm (F1 0.17). Although our data set for this part of the experiment may be considered small in IR terms, a two-sided t-test of the null hypothesis of equal means shows that all system results are statistically significant at the 1% level, except for the difference between the RANDOM and LEAD results and the TFIDF and LexGister(b) results which are not significant.

One of the main criticisms of an automatic experiment like the one just described is that it ignores important summary attributes like readability and grammatical correctness. It also fails to recognise cases where synonymous or semantically similar words are used in a system and reference title for a news story. This is a side effect of our experimental methodology where the set of gold standard human generated titles contain many instances of words that do not occur in the original text of the news story. This makes it impossible in some cases for an extractive approach to replicate an original title. For example consider the following gists where 'Jerusalem' is replaced by 'Israel' and 'killed' is replaced by 'die': "10 killed in suicide bombing in Jerusalem" and "10 die in suicide bombing in Israel". Examples like these account for a reduction in gisting performance and illustrate how essential an intrinsic or user-oriented evaluation is for determining the 'true' quality of a gist. In the following section we describe the results of an experiment involving human judges that addresses these concerns.

6 Manual Evaluation Results

As described in Section 4, the manual evaluation of the LexGister output involves determining the quality of a gist using human judges as assessors. 100 randomly selected news stories from our closed caption data set were used for this part of the evaluation. Judges were asked to rate gists with a score ranging from 5 (a very good attempt) to 1 (a bad attempt). The average of the scores assigned by each of the six judges was then taken as the overall rating for the headlines produced by the LexGister system, where the average score was 3.56 (i.e. gists where 'ok' to 'good') with a standard deviation of 0.32 indicating strong agreement among the judges.

Since judges were asked to rate gist quality based on readability and content there were a number of situations where the gist may have captured the crux of the story but its rating was low due to problems with its fluency or readability. These problems are a side effect of dealing with error prone closed caption data that contains both segmentation errors and breaks in transmission. To estimate the impact of this problem on the rating of the titles we also asked judges to indicate if they believed that the headline encapsulated the essence of the story disregarding grammatical errors. This score was a binary decision (1 or 0), where the average judgement was that 81.33% of titles captured the central message of the story with a standard deviation of 10.52 %. This 'story essence' score suggests that LexGister headlines are in fact better than the results of the automatic evaluation suggest, since the problems resulting from the use of semantically equivalent yet syntactically different words in the system and reference titles (e.g. Jerusalem, Israel) do not apply in this case. However, reducing the number of grammatical errors in the gists is still a problem as 36% of headlines contain these sorts of errors due to 'noisy' closed caption data. An example of such an error is illustrated below where the text in italics at the beginning of the sentence has been incorrectly concatenated to the gist due to a transmission error.

> "*on tax rates relating from* Tens of thousands of commuters travelled free of charge on trains today."

It is hoped that the sentence compression strategy set out in the following section discussing Future Work will be able to remove unwanted elements of text like this from the gists. One final comment on the quality of the gists relates to the occurrence of ambiguous expressions, which occurred in 23% of system generated headlines. For example, consider the following gist which leaves the identity of 'the mountain' to the readers imagination:

> "A 34-year-old South African hotel worker collapsed and died while coming down the mountain".

To solve this problem a 'post-gisting' component would have to be developed that could replace a named entity with the longest sub-string that co-refers to it in the text [22], thus solving the ambiguous location of 'the mountain'.

Finally, a similar gisting experiment was conducted by Jin and Hauptmann [21] who found that their language modeling-based approach to title generation achieved an F1 of 0.26 and a human judgement score of 3.07 (compared with an F1 of 0.20 and a human judgement score of 3.56 for the LexGister system). Their data set consisted of 1000 documents randomly selected from the 1997 collection of broadcast news transcriptions published by Primary Source Media. All 1000 documents were used in

the automatic evaluation, while 100 randomly selected documents were chosen for the manual evaluation. Although these results are not directly comparable with ours, they somewhat verify that the performance of our method is approaching the performance of other state-of-the-art broadcast news title generation systems. Also, considering that the upper bound on performance in this experiment is an F1 of 0.25 and the LexGister achieves an F1 of 0.20, this is further evidence that our approach is an adequate solution to this problem.

7 Related Research and Future Work

In this paper we have explored various extractive approaches to gisting, some other notable approaches in this area include Kraaij et al.'s [23] probabilistic approach, Alfonseca et al.'s [24] genetic algorithmic approach, and Copeck et al.'s [25] approach based on the occurrence of features that denote appropriate summary sentences. These lexical, syntactic and semantic features include the occurrence of discourse cues, the position of the sentence in the text, and the occurrence of content phrases and proper nouns. Biasing the extraction process with additional textual information like these features is a standard approach to headline generation that has proved to be highly effective in most cases [23-26].

An alternative to extractive gisting approaches is to view the title generation process as being analogous to statistical machine translation. Wittbrock and Mittal's paper on 'ultra-summarisation' [3], was one of the first attempts to generate headlines based on statistical learning methods that make use of large amounts of training data. More specifically, during title generation a news story is 'translated' into a more concise version using the Noisy Channel model. The Viterbi algorithm is then used to search for the most likely sequence of tokens in the text that would make a readable and informative headline. This is the approach adopted by Banko et al. [27], Jin and Hauptmann [21], Berger and Mittal [28] and more recently by Zajic and Dorr [29].

These researchers often state two advantages of their generative technique over an extractive one. Firstly, extractive techniques cannot deal with situations where important information may be scattered across more than one sentence. However, from an observation of our gisting results the extent of this problem may not be as pronounced as has been suggested. This is largely due to the fact that titles are so 'short and snappy' that finding the central message of the story is often sufficient and adding less important details occurring in other interesting sentences is not necessary. Also extractive techniques can work very well on gisting in a news story context as suggested by Dorr and Zajic [30] in their DUC data survey that found that the majority of headline words occur in the first sentence of a news article. The second criticism of extractive techniques related to their inability to create compact representations of a text that are smaller than a sentence. However, one of the advantages of extracting a readable and syntactically correct unit of text is that it can then be compressed using discourse analysis techniques and other linguistically rich methods. In contrast, the readability of a generated title is dependant on a 'title word ordering phrase' [3], which is based on

statistical probabilities rather than any explicit consideration of grammatical correctness.

The next stage in our research is to follow the lead of current trends in title generation and use linguistically motivated heuristics to reduce a gist to a skeletal form that is grammatically and semantically correct [9, 30-32]. We have already begun working on a technique that draws on parse tree information for distinguishing important clauses in sentences using the original lexical chains generated for the news story to weight each clause. This will allow the LexGister to further hone in on which grammatical unit of the sentence is most cohesive with the rest of the news story resulting in a compact news story title. Comparing the performance of the LexGister with a generative approach to gisting is also a future goal of our research.

8 Conclusions

In this paper we have discussed our novel lexical chaining-based approach to news story gisting in the broadcast news domain. More specifically, the aim of our research is to develop a robust gisting strategy that can deal with 'noisy' closed caption material from news programmes and provide users in an interactive multimedia system with a compact headline representing the main gist of the information in a news story. We have shown the effectiveness of our technique using an intrinsic and automatic evaluation methodology. In the automatic evaluation, we compared the performance of our LexGister system to four other baseline extraction systems using recall, precision and F1 metrics to measure gist quality against a set of gold standard titles. The LexGister outperforms all systems including another lexical chaining-based gister which used a more simplistic chaining strategy. This result verifies that our novel lexical chaining approach, which incorporates both non-WordNet proper nouns and statistical word associations into the chain generation process, can greatly improve the quality of the resultant gists. The results of a user-based evaluation of gist quality also concluded that the LexGister is capable of generating informative and human readable gists for closed caption news stories.

Acknowledgements

The support of the Informatics Directorate of Enterprise Ireland is gratefully acknowledged. The authors are also grateful to the anonymous reviewers for their helpful comments.

References

1. Smeaton A.F., H. Lee, N. O'Connor, S Marlow, N. Murphy, TV News Story Segmentation, Personalisation and Recommendation. AAAI 2003 Spring Symposium on Intelligent Multimedia Knowledge Management, Stanford University 24-26 March 2003.
2. Document Understanding Conferences (DUC): www-nlpir.nist.gov/projects/duc/intro.html
3. Witbrock, M., V. Mittal, Ultra-Summarisation: A Statistical approach to generating highly condensed non-extractive summaries. In the Proceedings of the ACM-SIGIR, pp. 315-316, 1999.
4. Morris J., G. Hirst, *Lexical Cohesion by Thesaural Relations as an Indicator of the Structure of Text*, Computational Linguistics 17(1), 1991.
5. Halliday M.A.K., Spoken and Written Language. Oxford University Press, 1985.
6. Green S.J., Automatically Generating Hypertext By Comparing Semantic Similarity. University of Toronto, Technical Report number 366, October 1997.
7. Barzilay R., M. Elhadad, Using Lexical Chains for Text Summarization. In the proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL, 1997.
8. Silber G.H., Kathleen F. McCoy, Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization. Computational Linguistics 28(4): 487-496, 2002.
9. Fuentes M., H. Rodriguez, L. Alonso, Mixed Approach to Headline Extraction for DUC 2003. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
10. Chali, Y., M. Kolla, N. Singh, Z. Zhang, The University of Lethbridge Text Summarizer at DUC 2003. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
11. St-Onge D., *Detecting and Correcting Malapropisms with Lexical Chains*, Dept. of Computer Science, University of Toronto, M.Sc. Thesis, 1995.
12. Stairmand M.A, A Computational Analysis of Lexical Cohesion with Applications in IR, PhD Thesis, Dept. of Language Engineering, UMIST. 1996.
13. Stokes, N., J. Carthy, First Story Detection using a Composite Document Representation, In the Proceedings of the Human Language Technology Conference, pp. 134-141, 2001.
14. Stokes N., J. Carthy, A.F. Smeaton, Segmenting Broadcast News Streams using Lexical Chains. In the Proceedings of STAIRS, pp. 145-154, 2002.
15. Okumura M., T. Honda, Word sense disambiguation and text segmentation based on lexical cohesion. In proceedings of COLING-94, pp. 755-761, 1994.
16. Miller G.A., R. Beckwith, C. Fellbaum, D. Gross, K. Miller, Five Papers on WordNet. CSL Report 43, Cognitive Science Laboratory, Princeton University, July 1990.
17. Allan J., J. Carbonell, G. Doddington, J. Yamron, Y. Yang. *Topic Detection and Tracking Pilot Study Final Report.* In the proceedings of the DARPA Broadcasting News Workshop, pp. 194-218, 1998.
18. Justeson, J. S., S.M. Katz., Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering (11): 9-27, 1995.
19. Xu J., J. Broglio, and W.B. Croft. The design and implementation of a part of speech tagger for English. Technical Report IR-52, University of Massachusetts, Amherst, Center for Intelligent Information Retrieval, 1994.
20. Salton G., M.J. McGill. Introduction to Modern Information Retrieval. New York: McGraw-Hill, 1983.
21. Jin R., A.G. Hauptmann. A new probabilistic model for title generation. In the Proceedings of the International Conference on Computational Linguistics, 2002.
22. Dimitrov, M.A light-weight approach to co-reference resolution for named entities in text, Master's Thesis, University of Sofia, 2002.

23. Kraaij, W., M. Spitters, A. Hulth. Headline extraction based on a combination of uni- and multi-document summarization techniques. In the Proceedings of the ACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2002), 2002.
24. Alfonseca, E., P. Rodriguez. Description of the UAM system for generating very short summaries at DUC 2003. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
25. Copeck T., S. Szpakowicz. Picking phrases, picking sentences. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
26. Zhou, L., E. Hovy. Headline Summarization at ISI. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
27. Banko M., V. Mittal, M. Witbrock. Generating Headline-Style Summaries. In the Proceedings of the Association for Computational Linguistics, 2000.
28. Berger, A.L., V.O. Mittal: OCELOT: a system for summarizing Web pages. In the Proceedings of the ACM-SIGIR, pp.144-151, 2000.
29. Zajic, D., B. Dorr. Automatic headline generation for newspaper stories. In the Proceedings of the ACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2002), 2002.
30. Dorr, B., D. Zajic. Hedge Trimmer: A parse-and-trim approach to headline generation. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
31. McKeown, K., D. Evans, A. Nenkova, R. Barzilay, V. Hatzivassiloglou, B. Schiffman, S. Blair-Goldensohn, J. Klavans, S. Sigelman. The Columbia Multi-Document Summarizer for DUC 2002. In the Proceedings of the ACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2002), 2002.
32. Daume, H., D. Echihabi, D. Marcu, D.S. Munteanu, R. Soricut. GLEANS: A generator of logical extracts and abstracts for nice summaries. In the Proceedings of the ACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2002), 2002.
33. Callan J.P., W.B. Croft and S.M. Harding. *The INQUERY Retrieval System*, Database and Expert Systems Applications. In the Proceedings of the International Conference in Valencia, Spain, A.M. Tjoa and I. Ramos (ed.), Springer-Verlag, New York, 1992.
34. Porter, M.F. An algorithm for suffix stripping, *Program*, 14(3) :130-137, 1980.