Available online at www.sciencedirect.com
ScienceDirect
Information Processing and Management 43 (2007) 1663–1680
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Two uses of anaphora resolution in summarization

Josef Steinberger [a,*], Massimo Poesio [b,c], Mijail A. Kabadjov [b], Karel Ježek [a]

[a] University of West Bohemia, Univerzitni 8, Pilsen 306 14, Czech Republic
[b] University of Essex, Wivenhoe Park, Colchester CO4 3SQ, United Kingdom
[c] Universitá di Trento, Rovereto, TN 38100, Italy

Received 17 July 2006; received in revised form 8 January 2007; accepted 10 January 2007
Available online 6 March 2007

Abstract

We propose a new method for using anaphoric information in Latent Semantic Analysis (LSA), and discuss its application to develop an LSA-based summarizer which achieves a significantly better performance than a system not using anaphoric information, and a better performance by the ROUGE measure than all but one of the single-document summarizers participating in DUC-2002. Anaphoric information is automatically extracted using a new release of our own anaphora resolution system, GUITAR, which incorporates proper noun resolution. Our summarizer also includes a new approach for automatically identifying the dimensionality reduction of a document on the basis of the desired summarization percentage. Anaphoric information is also used to check the coherence of the summary produced by our summarizer, by a reference checker module which identifies anaphoric resolution errors caused by sentence extraction.

Keywords: Summarization; Latent semantic analysis; Singular value decomposition; Anaphora resolution

1. Introduction

Information about anaphoric relations could be beneficial for applications such as summarization and segmentation that involve extracting (possibly very simplified) discourse models from text. In this work we investigated exploiting automatically extracted information about the anaphoric relations in a text for two different aspects of the summarization task. First of all, we used anaphoric information to enrich the latent semantic representation of a document, from which a summary is then extracted. Secondly, we used anaphoric information to check that the anaphoric expressions contained in the summary thus extracted still have the same interpretation that they had in the original text.

Our approach to summarization follows what has been called a TERM-BASED approach (Hovy & Lin, 1997). In term-based summarization, the most important information in a document is found by identifying its main 'terms' (also sometimes called 'topics'), and then extracting from the document the most important information

* Corresponding author.
E-mail address: jstein@kiv.zcu.cz (J. Steinberger).

0306-4573/$ - see front matter
doi:10.1016/j.ipm.2007.01.010

about these terms. Such approaches are usually classified as 'lexical' approaches or 'coreference- (or anaphora-) based' approaches. Lexical approaches to summarization use word similarity and other lexical relations to identify central terms (Barzilay & Elhadad, 1997); we would include among these previous approaches based on LSA (Landauer & Dumais, 1997), such as Gong and Liu (2002) and Steinberger and Jezek (2004). Coreference- or anaphora-based approaches[1] (Baldwin & Morton, 1998; Bergler, Witte, Khalife, Li, & Rudzicz, 2003; Boguraev & Kennedy, 1999; Stuckardt, 2003) identify these terms by running a coreference- or anaphoric resolver over the text. We are not aware, however, of any attempt to use both lexical and anaphoric information to identify the main terms, other than our own previous work (Steinberger, Kabadjov, & Poesio, 2005).

In this paper, we present a new LSA-based sentence extraction summarizer which uses both lexical and anaphoric information. We already found in previous work that feeding (automatically extracted) information about anaphoric relations to a summarizer can improve its performance (Steinberger et al., 2005). In the work discussed here, however, we improve our previous methods in several respects. Firstly of all, we improved our methods for extracting a summary from a document in two ways. Firstly, we developed an improved version of our anaphoric resolver, GUITAR, so that now it can also identify coreference relations between proper nouns, which are often used to indicate key terms. Secondly, we improved our method for extracting summaries from an LSA-style representation by developing a new approach for finding the dimensionality reduction on the basis of the required (percentage) size of the summary. The new system was evaluated on the standard reference corpus from DUC-2002, making it possible to compare its performance not only with that of two LSA-based summarizers using only lexical information, but also with that of the other systems participating in DUC-2002, using the standard ROUGE evaluation measure.

In addition, we also propose here a method for using anaphoric information to check the entity-coherence of a summary once this has been extracted. Summarization by sentence extraction may produce summaries with 'dangling' anaphoric expressions – expressions whose antecedent has not been included in the summary, and therefore cannot be interpreted or are interpreted incorrectly. Our algorithm checks that the interpretation of anaphoric expressions in a summary is consistent with their interpretation in the original text. The algorithm can be used irrespective of whether the summary was produced using our own methods or other methods.

The structure of the paper is as follows. In Section 2, some background information is presented. Our previous work using pure Latent Semantic Analysis (LSA) for summarization is discussed, and we present two LSA-based summarizers that only use lexical information to identify the main topics of a document. We then make the case for using anaphoric information as well as lexical information, and introduce our anaphora resolution system (Section 3). Then, in Section 4, we discuss our methods for using anaphoric information in LSA, and their evaluation using the DUC-2002 corpus. In Section 5 we present the last step in our summarization approach, the summary reference checker. In the end, we discuss our vision of applying the presented ideas to multi-document summarization.

2. Using LSA for summarization

2.1. Previous work

LSA (Landauer & Dumais, 1997) is a technique for extracting the 'hidden' dimensions of the semantic representation of terms, sentences, or documents, on the basis of their use. It has been extensively used in educational applications such as essay ranking (Landauer & Dumais, 1997), as well as in NLP applications including information retrieval (Berry, Dumais, & O'Brien, 1995) and text segmentation (Choi, Wiemer-Hastings, & Moore, 2001).

More recently, a method for using LSA for multi- and single-document summarization has been proposed in Gong and Liu (2002). This purely lexical approach is the starting point for our own work. The heart of Gong and Liu's method is a document representation developed in two steps. The first step is the creation of a term

[1] We use the term 'anaphora resolution' to refer to the task of identifying successive mentions of the same discourse entity, as opposed to the task of 'coreference resolution' which involves collecting all information about that entity, including information expressed by appositions and other predicative constructions.

by sentences matrix $A = [A_1, A_2, \ldots, A_n]$, where each column A_i represents the weighted term-frequency vector of sentence i in the document under consideration. The vector $A_i = [a_{1i}, a_{2i}, \ldots, a_{ni}]^T$ is defined as

$$a_{ji} = L(t_{ji}) \cdot G(t_{ji}), \tag{1}$$

where t_{ji} denotes the frequency with which term j occurs in sentence i, $L(t_{ji})$ is the local weight for term j in sentence i, and $G(t_{ji})$ is the global weight for term j in the whole document. The weighting scheme we found to work best is using a binary local weight and an entropy-based global weight:

$$L(t_{ji}) = 1, \text{if term } j \text{ appears at least once in the sentence } i; \text{ otherwise, } L(t_{ji}) = 0, \tag{2}$$

$$G(t_{ji}) = 1 - \sum_i \frac{p_{ji} \log(p_{ji})}{\log(nsent)}, \tag{3}$$

where $p_{ji} = t_{ji}/g_{ji}$, g_{ji} is the total number of times that term j occurs in the whole document and *nsent* is the number of sentences in the document.

If there are m terms and n sentences in the document, then we will obtain an $m \times n$ matrix A for the document. The next step is to apply Singular Value Decomposition (SVD) to matrix A. The SVD of an $m \times n$ matrix A is defined as

$$A = U \Sigma V^T, \tag{4}$$

where $U = [u_{ij}]$ is an $m \times n$ column-orthonormal matrix whose columns are called left singular vectors. $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is an $n \times n$ diagonal matrix, whose diagonal elements are non-negative singular values sorted in descending order. $V = [v_{ij}]$ is an $n \times n$ orthonormal matrix, whose columns are called right singular vectors. The dimensionality of the matrices is reduced to r most important dimensions and thus, U is $m \times r$, Σ is $r \times r$ and V^T is $r \times n$ matrix.

From a mathematical point of view, SVD derives a mapping between the m-dimensional space specified by the weighted term-frequency vectors and the r-dimensional singular vector space.

From an NLP perspective, what SVD does is to derive the *latent semantic structure* of the document represented by matrix A: i.e. a breakdown of the original document into r linearly-independent base vectors which express the main 'topics' of the document. SVD can capture interrelationships among terms, so that terms and sentences can be clustered on a 'semantic' basis rather than on the basis of words only. Furthermore, as demonstrated in Berry et al. (1995), if a word combination pattern is salient and recurring in document, this pattern will be captured and represented by one of the singular vectors. The magnitude of the corresponding singular value indicates the importance degree of this pattern within the document. Any sentences containing this word combination pattern will be projected along this singular vector, and the sentence that best represents this pattern will have the largest index value with this vector. Assuming that each particular word combination pattern describes a certain topic in the document, each singular vector can be viewed as representing such a topic (Ding, 2005), the magnitude of its singular value representing the degree of importance of this topic.

The summarization method proposed by Gong and Liu (2002) uses the representation of a document thus obtained to choose the sentences to go in the summary on the basis of the relative importance of the 'topics' they mention, described by the matrix V^T. The summarization algorithm simply chooses for each 'topic' the most important sentence for that topic: i.e., the kth sentence chosen is the one with the largest index value in the kth right singular vector in matrix V^T.

2.2. Sentence selection by vector length

The main drawback of Gong and Liu's method is that when l sentences are extracted the top l topics are treated as equally important. As a result, a summary may include sentences about 'topics' which are not particularly important.

Ding (2005) proved that the statistical significance of each LSA dimension (i.e., topic) is approximately the square of its singular value. In our own previous work (Steinberger & Jezek, 2004), we exploited Ding's result by changing the selection criterion to include in the summary the sentences whose vectorial representation in

the matrix $B = \Sigma^2 \cdot V^T$ has the greatest 'length', instead of the sentences containing the highest index value for each 'topic'. Intuitively, the idea is to choose the sentences with greatest combined weight across all topics, possibly including more than one sentence about an important topic, rather than always choosing one sentence for each topic as done by Gong and Liu. More formally: after computing the SVD of a term by sentences matrix, we compute matrix B:

$$B = \begin{pmatrix} v_{1,1}\sigma_1^2 & v_{1,2}\sigma_1^2 & \dots & v_{1,n}\sigma_1^2 \\ v_{2,1}\sigma_2^2 & v_{2,2}\sigma_2^2 & \dots & v_{2,n}\sigma_2^2 \\ \dots & \dots & \dots & \dots \\ v_{r,1}\sigma_r^2 & v_{r,2}\sigma_r^2 & \dots & v_{r,n}\sigma_r^2 \end{pmatrix}. \tag{5}$$

Then, we measure the length s_k of each sentence vector in B:

$$s_k = \sqrt{\sum_{i=1}^{r} b_{i,k}^2}, \tag{6}$$

where s_k is the length of the vector of kth sentence in the modified latent vector space, and its significance score for summarization too. We then include in the summary the sentences with the highest values in vector s. We showed (Steinberger & Jezek, 2004) that this modification results in a significant improvement over Gong and Liu's method.

2.3. Automatic dimensionality reduction

The algorithm proposed by Steinberger and Jezek (2004) still requires a method for deciding how many LSA dimensions/topics to include in the latent space and therefore in the summary. If we take too few, we may lose topics which are important from a summarization point of view. But if we take too many, we end up including less important topics. One of the novelties of the present work with respect to our own previous proposals is a new approach for computing this level of dimensionality reduction (r) automatically.

When we perform SVD on an $m \times n$ matrix, we can view the new dimensions as some sort of pseudo sentences: linear combinations of the original terms (left singular vectors), sorted according to their significance within the document. From a summarization point of view, the number of extracted sentences is dependent on the summary ratio. We know what percentage of the full text the summary should be: part of the input to the summarizer is that a p% summary is needed. (The length is usually measured in the number of words, but there are other possibilities.) If the pseudo sentences were real sentences that a reader could interpret, we could simply extract the top r pseudo sentences, where $r = p/100 * n$. However, because the linear combinations of terms are not really readable sentences, we use the above sentence selection algorithm to extract the actual sentences that 'overlap the most' in terms of vector length with top r pseudo sentences. In addition, our algorithm takes into account the significance of each dimension by multiplying the matrix V^T by Σ^2.

The summarizer can thus automatically determine the number of significant dimensions dependent on the summarization ratio. The larger the summary (measured in the percentage of the full text), the more topics are considered important in the process of summary creation. And because we know the contribution of each topic from the square of its singular value we can measure how much information is considered important by the dimensionality reduction approach for each full text percentage. Fig. 1 shows the logarithmic dependency between summary ratio and sum of relative significances of r most important dimensions[2]: for instance, a 10% summary contains the sentences that best cover 40% of document information, whereas a 30% summary will contain the sentences that most closely include 70% of document information.

[2] Suppose, for example, we have singular values [10, 7, 5, ...], that their significances (squares of singular values) are [100, 49, 25, ...], and that the total significance is 500 (sum of all singular value squares). Then the relative significances are [20%, 9.8%, 5% ...]: i.e., the first dimension captures 20% of the information in the original document. Thus, when the latent space contains 30 dimensions in total and summary ratio is 10% THEN r is set to 3. The sum of the relative significances of the three most important dimensions is 34.8%.

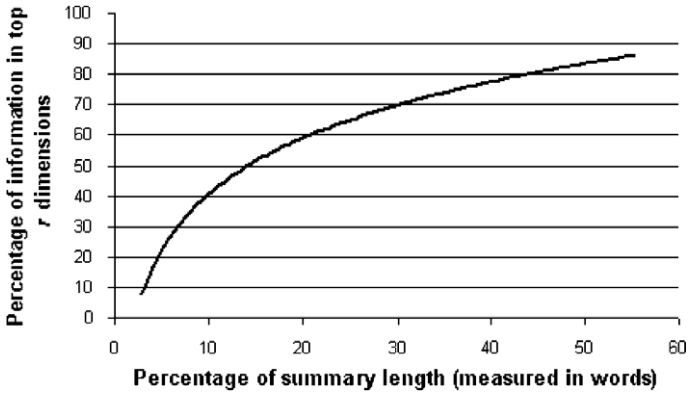

Fig. 1. The dependency of the sum of significances of r most important dimensions on the summary length. DUC-2002 data were used to create the curve.

3. Using anaphora resolution to find the most important terms

3.1. Motivations

Boguraev and Kennedy (1999) use the following news article to illustrate why being able to recognize anaphoric chains may help in identifying the main topics of a document.

PRIEST IS CHARGED WITH POPE ATTACK (7)

A Spanish priest was charged here today with attempting to murder the Pope. *Juan Fernandez Krohn*, aged 32, was arrested after a man armed with a bayonet approached the Pope while he was saying prayers at Fatima on Wednesday night. According to the police, *Fernandez* told the investigators today that *he* trained for the past six months for the assault. ... If found guilty, *the Spaniard* faces a prison sentence of 15–20 years.

As Boguraev and Kennedy point out, the title of the article is an excellent summary of the content: an entity (*the priest*) did something to another entity (*the pope*). Intuitively, this is because understanding that *Fernandez* and *the pope* are the central characters is crucial to provide a summary of texts like these.[3] Among the clues that help us to identify such 'main characters,' the fact that an entity is repeatedly mentioned is clearly important.

Methods that only rely on lexical information to identify the main topics of a text, such as the lexical-based methods discussed in the previous section, can only capture part of the information about which entities are frequently repeated in the text. As example (7) shows, stylistic conventions forbid verbatim repetition, hence the six mentions of *Fernandez* in the text above contain only one lexical repetition, '*Fernandez*'. The main problem are pronouns, that tend to share the least lexical similarity with the form used to express the antecedent (and anyway are usually removed by stopword lists, therefore do not get included in the SVD matrix). The form of definite descriptions does not always overlap with that of their antecedent, either, especially when the antecedent was expressed with a proper name. The form of mention which more often overlaps to a degree with previous mentions is proper nouns, and even then at least some way of dealing with acronyms is necessary (cfr. *European Union*/*E.U.*). On the other hand, it is well-known from the psychological literature that proper names often are used to indicate the main entities in a text. What anaphora resolution can do for us is to identify which discourse entities are repeatedly mentioned, especially when different forms of mention

[3] In many non-educational texts only a 'entity-centered' structure can be clearly identified, as opposed to a 'relation-centered' structure of the type hypothesized in Rhetorical Structures Theory and which serves as the basis for discourse structure-based summarization methods (Knott, Oberlander, O'Donnell, & Mellish, 2001; Poesio, Stevenson, Di Eugenio, & Hitzeman, 2004).

are used. We can then use the anaphoric chains identified by the anaphoric resolvers as additional terms in the initial matrix A in (4).

3.2. GUITAR

The anaphora resolution system we use, GUITAR (Poesio & Kabadjov, 2004; Kabadjov, Poesio, & Steinberger, 2005; Steinberger et al., 2005), is a publically available tool designed to be modular and usable as an off-the-shelf component of a NLP pipeline. In our previous work (Steinberger et al., 2005) we used a version that could only resolve pronouns and definite descriptions. However, as example (7) shows, proper nouns such as *Juan Fernandez Krohn* (and quasi-proper nouns such as *the Pope*) are generally used in at least one mention of the main entities of a text. This led to the second technical development in this paper, the use of version 3.2 of the system, also able to link proper nouns in coreference chains.

3.2.1. Preprocessing

The anaphora resolution proper part of GUITAR is designed to take XML input, in a special format called MAS-XML, and produce an output in the same format, but which additionally contains anaphoric annotation. The system can therefore work with a variety of preprocessing methods, ranging from a simple part-of-speech tagger to a chunker to a full parser, provided that appropriate conversion routines into MAS-XML are implemented. The version used for these experiments uses Charniak's parser (Charniak, 2000).

3.2.2. Anaphora resolution algorithms

We used two versions of the system in the experiments discussed in this paper. The earlier version, GUITAR 2.1, includes an implementation of the MARS pronoun resolution algorithm (Mitkov, 1998) to resolve personal and possessive pronouns. This system resolves definite descriptions using a partial implementation of the algorithm proposed in Vieira and Poesio (2000), augmented with a statistical discourse new classifier. The latest version, GUITAR 3.2, includes also an implementation of the shallow algorithm for resolving coreference with proper names proposed by Bontcheva, Dimitrov, Maynard, Tablan, and Cunningham (2002).

3.2.3. Evaluation

GUITAR has been evaluated over a variety of corpora. We report here the results with two different corpora: a corpus in which noun phrases have been identified by hand – the GNOME corpus, consisting of a variety of texts from different domains – and a corpus in which noun phrases are identified by the Charniak parser, 37 texts from the CAST corpus (Orasan, Mitkov, & Hasler, 2003) (these are news articles, mostly from the Reuters corpus). We expect the performance of the anaphoric resolver on the DUC corpus to be similar to its performance on the CAST corpus. The results of version 3.0 of the system with each corpus, and each type of anaphoric expression, are summarized in Table 1.

Table 1
Evaluation of GUITAR 3.2

Corpus	Anaphor	Target #	P	R	F
GNOME	DD	195	70.4	63.6	66.8
	PersPro	307	78.1	77.8	78
	PossPro	202	79.1	73.3	76.1
	PN	132	49	72	58.3
	TOTAL	836	70.2	72.5	71.3
CAST	DD	407	68.7	54	60.5
	PersPro	307	44.6	46.9	45.7
	PossPro	209	54.3	54.1	54.2
	PN	433	34.7	62.8	44.7
	TOTAL	1356	55.2	45.8	50.1

4. Combining lexical information and anaphoric information to build the LSA representation

'Purely lexical' LSA determines the main 'topics' of a document on the basis of the simplest possible notion of term, simple words, as usual in LSA. In this section we will see, however, that anaphoric information can be easily integrated in an mixed lexical/anaphoric LSA representation by generalizing the notion of 'term' used in SVD matrices to include *discourse entities* as well, and counting a discourse entity *d* as occurring in sentence *s* whenever the anaphoric resolver identifies a noun phrase occurring in *s* as a mention of *d*.

4.1. SVD over word terms and discourse entities

The simplest way of using anaphoric information with LSA is the SUBSTITUTION METHOD: keep using only words as terms, and use anaphora resolution as a preprocessing stage of the SVD input matrix creation. I.e., after identifying the anaphoric chains, replace all anaphoric nominal expressions with the first element of their anaphoric chain. In example (7), for example, all occurrences of elements of the anaphoric chain beginning with *A Spanish priest* would be substituted by *A Spanish priest*. The resulting text would be as follows:

PRIEST IS CHARGED WITH POPE ATTACK (8)

A Spanish priest was charged here today with attempting to murder the Pope. *A Spanish priest*, aged 32, was arrested after a man armed with a bayonet approached the Pope while he was saying prayers at Fatima on Wednesday night. According to the police, *a Spanish priest* told the investigators today that *a Spanish priest* trained for the past six months for the assault. ... If found guilty, *a Spanish priest* faces a prison sentence of 15–20 years.

This text could then be used to build an LSA representation as discussed in the previous section. We will show shortly, however, that this simple approach does not lead to improved results.

A better approach, it turns out, is what we call the ADDITION METHOD: generalize the notion of 'term', treating anaphoric chains as another type of 'term' that may or may not occur in a sentence. The idea is illustrated in Fig. 2, where the input matrix A contains two types of 'terms': terms in the lexical sense (i.e., words) and terms in the sense of discourse entities, represented by anaphoric chains. The representation of a sentence then specifies not only if that sentence contains a certain word, but also if it contains a mention of a discourse entity. With this representation, the chain 'terms' may tie together sentences that contain the same anaphoric chain even if they do not contain the same word. The resulting matrix would then be used as input to SVD as before.

4.2. First experiments: the CAST corpus

A pilot evaluation of the methods just discussed using an early version of GUITAR, GUITAR 2.1, was presented in Steinberger et al. (2005). We briefly report these preliminary results as background to the discussion of our novel experiments with the DUC-2002 corpus, discussed in the following section.

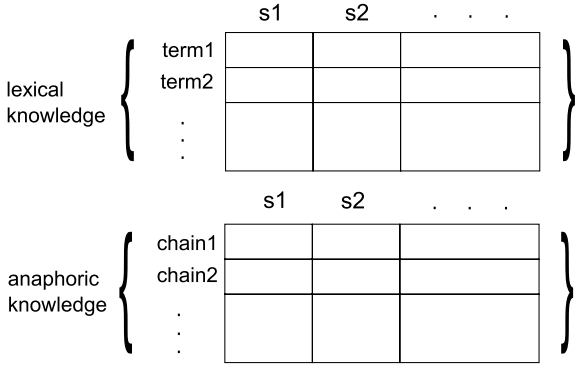

Fig. 2. Using discourse entities as terms.

4.2.1. The CAST corpus

In this pilot evaluation, we used the corpus of manually produced summaries created by the CAST project (Orasan et al., 2003). The CAST corpus contains news articles taken from the Reuters Corpus and a few popular science texts from the British National Corpus. Summaries are specified by providing information about the importance of sentences (Hasler, Orasan, & Mitkov, 2003): sentences are marked as **essential** or **important** for the summary. The corpus also contains annotations for **linked** sentences, which are not significant enough to be marked as important/essential, but which have to be considered as they contain information essential for the understanding of the content of other sentences marked as essential/important.

Four annotators were used for the annotation, three graduate students and one postgraduate. Three of the annotators were native English speakers, and the fourth had advanced knowledge of English. Unfortunately, not all of the documents were annotated by all of the annotators. To maximize the reliability of the summaries used for evaluation, we chose the documents annotated by the greatest number of the annotators; in total, our evaluation corpus contained 37 documents.

For acquiring manual summaries at specified lengths and getting the sentence scores (for relative utility evaluation) we assigned a score 3 to the sentences marked as essential, a score 2 to important sentences and a score 1 to linked sentences.

4.2.2. Evaluation measures

Evaluating summarization is a notoriously hard problem, for which standard measures like Precision and Recall are not very appropriate (Lin & Hovy, 2003; Radev, Jing, & Budzikowska, 2000). The main problem with such measures is that human judges often disagree on what are the top N most important sentences in a document, so by using precision and recall, two equally good summaries may be judged very differently. Suppose that a manual summary contains sentences [1 2] from a document, but that sentence 3 is an equally good alternative to sentence 2. Suppose now that two systems, A and B, produce summaries consisting of sentences [1 2] and [1 3], respectively. Using precision and recall as evaluation measures, system A will be ranked much higher than system B, whereas if sentences 2 and 3 are equally important, the two systems should get the same score.

In this early study, we addressed the problem by using a combination of evaluation measures. As a main measure we chose **relative utility** (RU) (Radev et al., 2000) because it could be computed automatically given the already existing annotations in the CAST corpus. RU allows model summaries to consist of sentences with variable ranking. With RU, the model summary represents all sentences of the input document with confidence values for their inclusion in the summary. For example, a document with five sentences [1 2 3 4 5] is represented as [1/5 2/4 3/4 4/1 5/2]. The second number in each pair indicates the degree to which the given sentence should be part of the summary according to a human judge. This number is called the utility of the sentence. Utility depends on the input document, the summary length, and the judge. In the example, the system that selects sentences [1 2] will not get a higher score than a system that chooses sentences [1 3] given that both summaries [1 2] and [1 3] carry the same number of utility points (5 + 4). Given that no other combination of two sentences carries a higher utility, both systems [1 2] and [1 3] produce optimal extracts. To compute relative utility, a number of judges, ($N \geqslant 1$) are asked to assign utility scores to all n sentences in a document. The top e sentences according to utility score[4] are then called a sentence extract of size e. We can then define the following system performance metric:

$$RU = \frac{\sum_{j=1}^{n} \delta_j \sum_{i=1}^{N} u_{ij}}{\sum_{j=1}^{n} \epsilon_j \sum_{i=1}^{N} u_{ij}}, \tag{9}$$

where u_{ij} is a utility score of sentence j from annotator i, ϵ_j is 1 for the top e sentences according to the sum of utility scores from all judges, otherwise is 0, and δ_j is equal to 1 for the top e sentences extracted by the system, otherwise is 0. (For details see Radev et al. (2000).) The second measure we used is **cosine similarity**, computed using the standard formula:

[4] In the case of ties, some arbitrary but consistent mechanism is used to decide which sentences should be included in the summary.

$$\cos(X, Y) = \frac{\sum_i x_i \cdot y_i}{\sqrt{\sum_i (x_i)^2} \cdot \sqrt{\sum_i (y_i)^2}}, \tag{10}$$

where X and Y are representations of a system summary and its reference summary based on the vector space model. The third measure is **Main Topic Similarity**. This is a content-based evaluation method based on measuring the cosine of the angle between first left singular vectors of a system summary's and its reference summary's SVDs. (For details see Steinberger & Jezek (2004).)

4.2.3. How much may anaphora resolution help? An upper bound

In order to determine whether anaphoric information might help, and which method of adding anaphoric knowledge to the LSA summarizer is best, we annotated by hand all the anaphoric relations in the 37 documents in the CAST corpus using the annotation tool MMAX (Mueller & Strube, 2003).

Results for the 15% (resp. 30%) summarization ratio using a variety of summarization evaluation measures are presented in Table 2 (resp. Table 3). The tables show that even with perfect knowledge of anaphoric links, the performance when using Substitution method does not change much. The problem that happened in some of the documents was that SVD deteriorated when a frequently used entity was substituted by its full nominal expression. As a result, the score of the sentence was extremly boosted when it contained the mention of the entity. And thus, sentences that contained the mention of this entity were all considered important, no matter what else they contained.

On the other hand, the addition method could potentially lead to substantial improvements.

4.2.4. Results with GUITAR 2.1

To use GUITAR, we first parsed the texts using Charniak's parser (Charniak, 2000). The output of the parser was then converted into the MAS-XML format expected by GUITAR by one of the preprocessors that come with the system. (This step includes heuristic methods for guessing agreement features.) Finally, GUITAR was ran to add anaphoric information to the files. The resulting files were then processed by the summarizer.

GUITAR achieved a precision of 56% and a recall of 51% over the 37 documents. For definite description resolution, we obtained a precision of 69% and a recall of 53%; for possessive pronouns resolution, precision was 53%, recall was 53%; for personal pronouns, precision was 44%, recall was 46%.

The results obtained by the summarizer using GUITAR's output are presented in Tables 4 and 5 (relative utility, f-score, cosine, and main topic).

Tables 4 and 5 clearly show that using GUITAR and the addition method leads to significant improvements over our baseline LSA summarizer. The improvement in Relative Utility measure was significant (95% confidence by the t-test). On the other hand, the substitution method did not lead to significant improvements,

Table 2
Improvement over lexical-based LSA with manually annotated anaphoric information – summarization ratio: 15%

Evaluation method	Lexical LSA	Manual substitution	Manual addition
Relative utility	0.595	0.573	0.662
F-score	0.420	0.410	0.489
Cosine similarity	0.774	0.806	0.823
Main topic similarity	0.686	0.682	0.747

Table 3
Improvement over lexical-based LSA with manually annotated anaphoric information – summarization ratio: 30%

Evaluation method	Lexical LSA	Manual substitution	Manual addition
Relative utility	0.645	0.662	0.688
F-score	0.557	0.549	0.583
Cosine similarity	0.863	0.878	0.886
Main topic similarity	0.836	0.829	0.866

Table 4
Evaluation of the results with GUITAR – summarization ratio: 15%

Evaluation method	Lexical LSA	GuiTAR substitution	GuiTAR addition
Relative utility	0.595	0.530	0.640
F-score	0.420	0.347	0.441
Cosine similarity	0.774	0.804	0.805
Main topic similarity	0.686	0.643	0.699

Table 5
Evaluation of the results with GUITAR – summarization ratio: 30%

Evaluation method	Lexical LSA	GuiTAR substitution	GuiTAR addition
Relative utility	0.645	0.626	0.678
F-score	0.557	0.524	0.573
Cosine similarity	0.863	0.873	0.879
Main topic similarity	0.836	0.818	0.868

as was to be expected given that no improvement was obtained with 'perfect' anaphora resolution (see previous section).

4.2.5. ROUGE evaluation of pilot study

We also evaluated the results using the ROUGE measure (see Section 4.3.3) – Tables 6 and 7 – obtaining improvements with the addition method, but the differences were not statistically significant.[5]

4.2.6. Pilot study conclusion

In conclusion, this pilot study showed that (i) we could expect performance improvements over purely lexical LSA summarization using anaphoric information, (ii) that significant improvements at least by the Relative Utility score could be achieved even if this anaphoric information was automatically extracted, and (iii) that, however, these results were only achievable using the Addition method.

What this earlier work did not show was how well are our results compared with the state of the art, as measured by evaluation over a standard reference corpus such as DUC-2002, and using the by now standard ROUGE measure. Furthermore, these results were obtained using a version of the anaphoric resolver that did not attempt to identify coreference links realized using proper names, even though proper names tend to be used to realize more important terms. Given our analysis of the effect of improvements to the anaphoric resolver (Kabadjov et al., 2005), we expected an improved version to lead to better results. The subsequent experiments were designed to address these questions.

4.3. Experiments with the DUC-2002 corpus

In the work just discussed we had not compared our purely lexical summarizer with other summarizers, which raised the question of whether the improvements we found were simply due to the poor quality of the original summarizer. In addition, we only achieved non-significant improvements by the ROUGE measures. To address these problems with our earlier work, we developed an improved version of the system, and we evaluated both the lexical and the anaphoric + lexical summarizers using the DUC-2002 corpus and the ROUGE measure, which would make it easier to contrast our results with those published in the literature.

[5] The values are larger than it is usual in standard DUC comparison of summaries and abstracts because summaries and extracts were compared here. Truncation of summaries to exact length could not be performed because the summary length was derived proportionally from source text length. ROUGE version and settings: ROUGEeval-1.4.2.pl-c95-m-n2-s-24-acast.xml.

Table 6
ROUGE scores for the pilot study – summarization ratio: 15%

System	ROUGE-1	ROUGE-2	ROUGE-SU4	ROUGE-L
Manual addition	0.64257	0.57896	0.55134	0.56609
GuiTAR addition	0.62297	0.55353	0.52693	0.54783
Lexical LSA	0.60359	0.53140	0.50115	0.53516
GuiTAR substitution	0.59273	0.50666	0.47908	0.52006
Manual substitution	0.53144	0.40629	0.37347	0.46431

Table 7
ROUGE scores for the pilot study – summarization ratio: 30%

System	ROUGE-1	ROUGE-2	ROUGE-SU-4	ROUGE-L
Manual addition	0.64257	0.57896	0.55134	0.56609
GuiTAR addition	0.62297	0.55353	0.52693	0.54783
Lexical LSA	0.60359	0.53140	0.50115	0.53516
GuiTAR substitution	0.59273	0.50666	0.47908	0.52006
Manual substitution	0.53144	0.40629	0.37347	0.46431

4.3.1. A new anaphoric resolver and a new summarizer

The version of our system used for this second experiment differs from the versions discussed in previous work in two respects. First of all, we developed the new method for automatic dimensionality reduction, discussed in 2.3. Secondly, we developed a new version of our anaphoric resolver, GUITAR 3.2, which, as discussed above, also resolves proper names.

We expected this new version could lead to improvements in performance, as well as being more usable.

4.3.2. The DUC-2002 corpus

DUC-2002 included a single-document summarization task, in which 13 systems participated. 2002 is the last version of DUC that included single-document summarization evaluation of informative summaries. Later DUC editions (2003 and 2004) contained a single-document summarization task as well, however only very short summaries (75 Bytes) were analyzed. However, we are not focused on producing headline-length summaries. The DUC-2002 corpus used for the task contains 567 documents from different sources; 10 assessors were used to provide for each document two 100-word human summaries. In addition to the results of the 13 participating systems, the DUC organizers also distributed baseline summaries (the first 100 words of a document). The coverage of all the summaries was assessed by humans.

4.3.3. The ROUGE evaluation metric

In DUC-2002, the SEE evaluation tool was used, but in later editions of the initiative the ROUGE measure was introduced (Lin & Hovy, 2003), which is now standard. We used ROUGE to compare our systems with those that participated in DUC.

The ROUGE-N score is computed as follows:

$$C_n = \frac{\sum_{C \in RSS} \sum_{gram_n \in C} Count_{match}(gram_n)}{\sum_{C \in RSS} \sum_{gram_n \in C} Count(gram_n)}, \tag{11}$$

where $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a reference summary (from a reference summary set- RSS) and $Count(gram_n)$ is the number of n-grams in the reference summary. (Notice that the average n-gram coverage score, C_n, is a recall metric.) There are other ROUGE, such as ROUGE-L–a longest common subsequence measure–and ROUGE-SU4–a skip bigram measure with the addition of unigrams as counting unit (Lin, 2004).

As (11) shows, ROUGE-N is actually a family of metrics; however, different ROUGE scores correlate in different ways with human assessments. As shown in Table 8, there is a strong correlation between humans and ROUGE-1 (and ROUGE-L). However, we used all those four ROUGE scores to determine significance results.

Table 8
Correlation between ROUGE scores and human assessments

Score	Correlation
ROUGE-1	0.92574
ROUGE-2	0.80090
ROUGE-SU-4	0.78396
ROUGE-L	0.92561

4.3.4. ROUGE evaluation over DUC-2002 data

We show in Table 9 the ROUGE scores[6] of two purely lexical LSA summarizers – GLLSA (Gong and Liu's approach) and LELSA (Length strategy, our approach); of our summarizer combining both lexical and anaphoric information (LELSA+AR); and of the 13 systems which participated in DUC-2002.[7] We also list a baseline and a random summarizer (the lowest baseline). Table 10 shows a multiple comparison of ROUGE scores between systems. Systems not sharing a common letter are significantly different (at the 95% confidence level).

The first result highlighted by these tables is that the two LELSA summarizers incorporating our improved dimensionality reduction methodology are state of the art. The performance of LELSA summarizer is significantly worse only than that the best system in DUC-2002, system 28, in ROUGE-1, ROUGE-2 and ROUGE-SU4, and significantly better than that of 9 in ROUGE-1, 7 in ROUGE-2, 7 in ROUGE-SU4 and 10 in ROUGE-L of the systems that participated in that competition. The second result is that both LELSA systems significantly outperform Gong and Liu's LSA approach (GLLSA). However, our LELSA+AR summarizer is even better: it is significantly better than 11 systems in ROUGE-1, 9 in ROUGE-2, 9 in ROUGE-SU4 and 13 in ROUGE-L, it is significantly better than the baseline in ROUGE-L at the 90% confidence level, and it is not significantly worse than any of the systems.

4.4. An example: a summary before and after anaphora resolution

Examples (12) and (13) illustrate the difference between a summary created by the pure LSA summarizer and the corresponding one created by the summarizer enhanced by anaphora resolution (addition method).

JURORS DEADLOCKED ON 13 CHARGES
(summary before anaphora resolution) (12)

Jurors who have reached verdicts on 52 counts in the McMartin preschool molestation case said Wednesday they are deadlocked on the remaining 13 charges in the nation's longest, costliest criminal trial. Superior Court Judge William Pounders received a note from the jurors as they ended their day's deliberation and called a hearing for Thursday to discuss the deadlock and possibly opening the 52 sealed verdicts. In an interview Wednesday evening, Pounders said he would deal with the deadlocked counts first, either declaring a mistrial on those counts and reading the sealed verdicts, or sending the jury back to resume deliberations on the undecided counts.

JURORS DEADLOCKED ON 13 CHARGES
(summary after anaphora resolution) (13)

Jurors who have reached verdicts on 52 counts in the McMartin preschool molestation case said Wednesday *they* are deadlocked on the remaining 13 charges in the nation's longest, costliest criminal trial. Superior Court Judge William Pounders received a note from *the jurors* as *they* ended *their* day's deliberation and called a hearing for Thursday to discuss the deadlock and possibly opening the 52 sealed verdicts. *The jurors* are deciding whether *Raymond Buckey, 31,* and *his mother, Peggy McMartin Buckey, 63,* are guilty or innocent of charges *they* molested children at *their family-owned McMartin Pre-School in Manhattan Beach.*

[6] All system summaries were truncated to 100 words as traditionally done in DUC. ROUGE version and settings: ROUGEeval-1.4.2.pl-c95-m-n2-l100-s-24-aduc.xml.

[7] The two systems with the poorest performance produce only headlines, which are much shorter than 100 words. This may be the reason for their poor results.

Table 9
ROUGE scores

System	ROUGE-1	ROUGE-2	ROUGE-SU-4	ROUGE-L
28	0.42776	0.21769	0.17315	0.38645
LeLSA+AR	0.42280	0.20741	0.16612	0.39276
21	0.41488	0.21038	0.16546	0.37543
DUC baseline	0.41132	0.21075	0.16604	0.37535
19	0.40823	0.20878	0.16377	0.37351
LeLSA	0.40805	0.19722	0.15728	0.37878
27	0.40522	0.20220	0.16000	0.36913
29	0.39925	0.20057	0.15761	0.36165
31	0.39457	0.19049	0.15085	0.35935
15	0.38884	0.18578	0.15002	0.35366
23	0.38079	0.19587	0.15445	0.34427
GLLSA	0.38068	0.17440	0.13674	0.35118
16	0.37147	0.17237	0.13774	0.33224
18	0.36816	0.17872	0.14048	0.33100
25	0.34297	0.15256	0.11797	0.31056
Random	0.29963	0.11095	0.09004	0.27951
17	0.13528	0.05690	0.04253	0.12193
30	0.07452	0.03745	0.02104	0.06985

Table 10
95% Significance groups for ROUGE scores

System	Significance groups			
	ROUGE-1	ROUGE-2	ROUGE-SU-4	ROUGE-L
28	A	A	A	AB
LeLSA+AR	AB	AB	AB	A
21	ABC	AB	AB	ABCD
DUC baseline	ABCD	AB	AB	ABCD
19	BCD	AB	ABC	BCD
LeLSA	BCD	BC	BC	ABC
27	BCD	ABC	ABC	BCDE
29	CDE	ABC	BC	CDEF
31	DE	CD	CDE	DEF
15	EF	CDE	CDE	EF
23	EFG	BC	BCD	FG
GLLSA	FG	DE	E	EF
16	FG	E	E	G
18	G	DE	DE	G
25	H	F	F	H
Random	I	G	G	I
17	J	H	H	J
30	K	I	I	K

From the examples it can be seen that the first two sentences selected by the summarizers are the same, whereas the third one is different. When using anaphora resolution, sentence selection was affected by strong anaphoric chains referring to salient entities (e.g., *the jurors*, *Raymond Buckey*, *Peggy McMartin Buckey*). The presence of the dominant entity, *the jurors*, in all three sentences served as 'glue' and kept the three sentences together throughout the process influencing the outcome of that summarizer. ROUGE scores for this particular document were significantly better when anaphora resolution was used.

5. A summary (entity) coherence checker

Anaphoric expressions can only be understood with respect to a context. This means that summarization by sentence extraction can wreak havoc with their interpretation: there is no guarantee that they will have an

interpretation in the context obtained by extracting sentences to form a summary, or that this interpretation will be the same as in the original text. Consider the following example.

PRIME MINISTER CONDEMNS IRA FOR MUSIC SCHOOL EXPLOSION (14)

(**S1**) [Prime Minister Margaret Thatcher]$_1$ said Monday [[the Irish Republican Army]$_2$ members who blew up [the Royal Marines School of Music]$_3$ and killed [10 bandsmen]$_4$ last week]$_5$ are monsters who will be found and punished.

(**S2**) ''[The young men whom we lost]$_4$ were murdered by [common murderers who must be found and brought to justice and put behind bars for a very long time]$_5$,'' [she]$_1$ said following a tour of [[the school's]$_3$ wrecked barracks]$_6$ in Deal, southeast England.

...

(**S3**) [Gerry Adams, president of [Sinn Fein, the legal political arm of [the IRA]$_2$]$_8$]$_7$ issued a statement disputing [Mrs. Thatcher's]$_1$ remarks, saying ''[she]$_1$ knows in [her]$_1$ heart of hearts the real nature of the conflict, its cause and the remedy''.

...

(**S4**) ''[We]$_8$ want an end to all violent deaths arising out of the present relationship between our two countries,'' [Adams]$_7$ said.

...

(**S5**) [The IRA]$_2$ claimed responsibility for the explosion, and police said they are looking for [three men with Irish accents who rented a house overlooking [the barracks]$_6$]$_5$.

If sentence S2 were to be extracted to be part of the summary, but S1 was not, the pronoun *she* would not be understandable as it would not have a matching antecedent anymore. The reference to *the school* would also be uninterpretable. The same would happen if S5 were extracted without also extracting S2; in this case, the problem would be that the antecedent for *the barracks* is missing.

Examples such as the one just shown suggested another use for anaphora resolution in summarization– correcting the references in the summary. Our idea was to replace anaphoric expressions with a full noun phrase in the cases where otherwise the anaphoric expression could be misinterpreted. We discuss this method in detail next.

5.1. The reference correction algorithm

Our correction algorithm works as follows:

(1) Run anaphora resolution over the source text, and create anaphoric chains.
(2) Identify the sentences to be extracted using a summarization algorithm such as the one discussed in the previous sections.
(3) For every anaphoric chain, replace the first occurrence of the chain in the summary with its first occurrence in the source text. After this step, all chains occurring in both source and summary start with the same lexical form.
    For example, in the text in (14), if sentence S4 is included in the summary, but S3 is not, the first occurrence of chain 7 in the summary, *Adams*, would be substituted by *Gerry Adams, president of Sinn Fein, the legal political arm of the IRA*.
(4) Run the anaphoric resolver over the summary.
(5) For all nominal expressions in the summary: if the expression is part of a chain in the source text and it is not resolved in the summary (the resolver was not able to find an antecedent), or if it is part of a different chain in the summary, then replace the anaphoric expression with the head of the first chain expression from the source text.

This method can be used in combination with the summarization system discussed in earlier sections, or with other systems; and becomes even more important when doing sentence compression, because intrasentential antecedents can be lost as well. However, automatic anaphora resolution can introduce new errors. We discuss our evaluation of the algorithm next.

Table 11
Evaluation of step 3, the first chain occurrence replacement

Statictic	Overall	Per document
Chains in full text	2906	18.8
Chains in summary	1086 (37.4% of full text chains)	7.0
First chain occurrence was in the summary	714 (65.7% of the chains in summaries)	4.6
First element of chain had same lexical form	101 (9.3% of the chains in summaries)	0.7
First chain occurrence replaced	271 (25% of the chains in summaries)	1.7
Correctly replaced	186 (Precision: 68.6%)	1.2

5.2. Evaluation

To measure the recall of the reference checker algorithm we would need anaphoric annotations, that were not available for DUC data. We measured its precision manually as follows. To measure the precision of the step where the first occurrences of a chain in the summary were replaced by the first mention of that chain in the source text, we took a sample of 155 documents[8] and measured precision by hand, obtaining the results shown in Table 11.

We can observe that full texts contained on average 19 anaphoric chains, and summaries about 7. In 66% of the summary chains the sentence where the first chain occurrence appeared was selected into the summary, and in 9% there was no need to replace the expression because it already had the same form as the first element of the chain. So overall the first chain occurrence was replaced in 25% of the cases; the precision was 68.6%. This suggest that the success in this task correlates with anaphora resolver's quality.

After performing anaphora resolution on the summary and computing its anaphoric chains, the anaphors without an antecedent are replaced. We analyzed a sample of 86 documents[9] to measure the precision by hand. Overall, 145 correct replacements were made in this step and 65 incorrect, for a precision of 69%. Table 12 analyzes the performance on this task in more detail.

The first row of the table lists the cases in which an expression was placed in a chain in the summary, but not in the source text. In these cases, our algorithm does not replace anything.

Our algorithm however does replace an expression when it finds that there is no chain assigned to the expression in the summary, but there is one in the source text; such cases are listed in the second row. We found that this replacement was correct in 32 cases; in 14 cases the algorithm replaced an incorrect expression.

The third row lists summarizes the most common case, in which the expression was inserted into the same chain in the source text and in the summary. (I.e., the first element of the chain in the summary is also the first element of the chain in the source text.) When this happens, in 83% of cases GUITARS' interpretation is correct; no replacement is necessary.

Finally, there are two subcases in which different chains are found in the source text and in the summary (in this case the algorithm performs a replacement). The fourth row lists the case in which the original chain is correct; the last, cases in which the chain in the source text is incorrect. In the first column of this row are the cases in which the anaphor was correctly resolved in the summary but it was substituted by an incorrect expression because of a bad full text resolution; the second column shows the cases in which the anaphor was incorrectly resolved in both the full text and the summary, however, replacement was performed because the expression was placed in different chains.

5.3. A summary before and after reference checking

Examples (15) and (16) illustrate the difference between a summary before and after reference checking. A reader of (15) may not know who *the 71-year-old Walton* or *Sively* are, and what *store* it is the text is talking about. In addition, the pronoun *he* in the last sentence is ambiguous between *Walton* and *Sively*. On the other

[8] We could not annotate all of the documents in the corpus. Therefore, we took a random sample of documents, where at least one document from each cluster was included.

[9] We took a random sample of documents, where at least one document from each cluster was included.

Table 12
Evaluation of step 5, checking the comprehensibility of anaphors in the summary

Observed state	Correct	Incorrect
Full text: expression in no chain Summary: expression in a chain	16 (66.7%)	8 (33.3%)
Full text: expression in a chain Summary: expression in no chain	32 (replaced +) (69.6%)	14 (replaced −) (30.4%)
Expression in the same chain in the full text and its summary	336 (83%)	69 (17%)
Expression in a different chain in the full text and its summary (correctly resolved in full text)	81 (replaced +) (71.7%) (correctly resolved in summary)	32 (replaced +) (28.3%) (incorrectly resolved in summary)
Expression in a different chain in the full text and its summary (incorrectly resolved in full text)	39 (replaced −) (76.5%) (in summary correctly resolved)	12 (replaced −) (23.5%) (in summary incorrectly resolved)
Replacements overall	145 (69%)	65 (31%)

Replaced '+' means that the expressions were correctly replaced; replaced '−' that the replacement was incorrect.

hand, *the singer* in the last sentence can be easily resolved. This is because the chains *Walton*, *Sively* and *the store* do not start in the summary with the expression used for the first mention in the source text. These problems are fixed by step 3 of the summary coherence checker algorithm. The ambiguous pronoun *he* in the last sentence of the summary is resolved to *Sively* in the summary and *Walton* in the source text.[10] Because the anaphor occurs in different chains in the summary and in the full text, it has to be substituted by the head of the first chain occurrence noun phrase, *Walton*. *The singer* in the last sentence is resolved identically in the summary and in the full text: the chains are the same, so there is no need for replacement.

WAL-MART FOUNDER PITCHES IN AT CHECK-OUT COUNTER

(summary before reference checking) (15)

The 71-year-old Walton, considered to be one of the world's richest people, grabbed a note pad Tuesday evening and started hand-writing merchandise prices for customers so their bills could be tallied on calculators quickly. *Walton*, known for his down-home style, made a surprise visit to ***the store*** that later Tuesday staged a concert by *country singer Jana Jea* in *its* parking lot. *Walton* often attends promotional events for *his* Arkansas-based chain, and ***Sively*** said *he* had suspected the boss might make an appearance. ***He*** also joined *the singer* on stage to sing a duet and led customers in the Wal-Mart cheer.

WAL-MART FOUNDER PITCHES IN AT CHECK-OUT COUNTER

(summary after reference checking) (16)

Wal-Mart founder Sam Walton, considered to be one of the world's richest people, grabbed a note pad Tuesday evening and started hand-writing merchandise prices for customers so their bills could be tallied on calculators quickly. *Walton*, known for his down-home style, made a surprise visit to ***his store in this Florida Panhandle city*** that later Tuesday staged a concert by *country singer Jana Jea* in *its* parking lot. *Walton* often attends promotional events for *his* Arkansas-based chain, and ***store manager Paul Sively*** said *he* had suspected the boss might make an appearance. ***Walton*** also joined *the singer* on stage to sing a duet and led customers in the Wal-Mart cheer.

6. Future work: multi-document summarization

We are currently working to apply the methods proposed here to multi-document summarization.

The single-document LSA approach can be easily extended to process multiple documents by including all sentences in a cluster of documents in the SVD input matrix. The latent space would be then reduced to *r* dimen-

[10] The previous sentence in the source is: "*Walton* continued talking with customers during the concert".

sions according to the dimensionality reduction approach as done currently (see Section 2.3). The sentence selection approach can be used as well; however, care has to be taken to avoid including very similar sentences from different documents. Therefore, before including a sentence in the summary we have to check if there are any sentences whose similarity with the observed one is above a given threshold. (The easiest way of measuring the similarity between two sentences is to measure the cosine of the angle between them in the term space.)

Cross-document coreference, on the other hand, is a fairly different task from within-document coreference, as even in the case of entities introduced using proper names one cannot always assume that the same object is intended, let alone in the case of entities introduced using definite descriptions. We are currently working on this problem.

7. Conclusion and further research

In this paper we presented three main contributions. First of all, we developed a method for using both anaphoric and lexical information to build an LSA representation which works well for summarization and, we believe, can also be used in other applications – e.g., in our work on text segmentation (Kabadjov, 2007). This method involves two novel ideas. First of all, we improved the Gong and Liu method for extracting a summary from an LSA representation, by providing a new method for automatically determining the dimensionality reduction given the summary ratio. Secondly, we developed the Addition method for incorporating anaphoric information in an LSA matrix. Combined, these two developments yield a summarizer which performs as well as the best DUC-2002 systems, and better than our purely lexical-based summarizer, even if this system is already performing as well as some of the best DUC systems. In other word, adding anaphoric information yields an improvement in performance even if the performance of our anaphoric resolver is still far from perfect.

The third contribution is a method for using anaphoric information to make the text produced by the summarizer more coherent by looking for anaphors that may be interpreted incorrectly in the summary, and replacing them with the original expressions. The precision at this task is 69%.

Combined, these contributions suggest that using anaphoric information does indeed lead to better summarizers. In future work, we plan to apply these ideas to multi-document summarization, also using sentence compression algorithms.

Acknowledgement

This research was partly supported by project 2C06009 (COT-SEWing).

References

Baldwin, B., & Morton, T. S. (1998). Dynamic coreference-based summarization. In *Proceedings of EMNLP, Granada, Spain*.

Barzilay, R., & Elhadad, M. (1997). Using lexical chains for text summarization. In *Proceedings of the ACL/EACL workshop on intelligent scalable text summarization, Madrid, Spain*.

Bergler, S., Witte, R., Khalife, M., Li, Z., & Rudzicz, F. (2003). Using knowledge-poor coreference resolution for text summarization. In *Proceedings of DUC, Edmonton, Canada*.

Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent IR. *SIAM Review, 37*(4).

Boguraev, B., & Kennedy, C. (1999). Salience-based content characterization of text documents. In I. Mani & M. T. Maybury (Eds.), *Advances in automatic text summarization*. Cambridge, US: MIT Press.

Bontcheva, K., Dimitrov, M., Maynard, D., Tablan, V., & Cunningham, H. (2002). Shallow methods for named entity coreference resolution. In *Chaînes de références et résolveurs d'anaphores, workshop TALN 2002, Nancy, France*.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of NAACL, Philadelphia, US*.

Choi, F. Y. Y., Wiemer-Hastings, P., & Moore, J. D. (2001) Latent semantic analysis for text segmentation. In *Proceedings of EMNLP, Pittsburgh, US*.

Ding, Ch. H. Q. (2005). A probabilistic model for latent semantic indexing. *Journal of the American Society for Information Science and Technology, 56*(6), 597–608.

Gong, Y., & Liu, X. (2002). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of ACM SIGIR, New Orleans, US*.

Hasler, L., Orasan, C., & Mitkov, R. (2003). Building better corpora for summarization. In *Proceedings of corpus linguistics*. United Kingdom: Lancaster.

Hovy, E., & Lin, C. (1997). Automated text summarization in summarist. In *ACL/EACL workshop on intelligent scalable text summarization, Madrid, Spain*.

Kabadjov, M. A. (2007). Anaphora resolution and applications. PhD Dissertation, University of Essex, UK.

Kabadjov, M. A., Poesio, M. & Steinberger, J. (2005). Task-based evaluation of anaphora resolution: the case of summarization. In *RANLP workshop "crossing barriers in text summarization research", Borovets, Bulgaria*.

Knott, A., Oberlander, J., O'Donnell, M., & Mellish, C. (2001). Beyond elaboration: the interaction of relations and focus in coherent text. In T. Sanders, J. Schilperoord, & W. Spooren (Eds.), *Text representation: linguistic and psycholinguistic aspects*. John Benjamins.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: the latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review, 104*, 211–240.

Lin, Ch. (2004). ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the workshop on text summarization branches out, Barcelona, Spain*.

Lin, Ch., & Hovy, E. (2003). Automatic evaluation of summaries using *n*-gram co-occurrence statistics. In *Proceedings of HLT-NAACL, Edmonton, Canada*.

Mitkov, R. (1998). Robust pronoun resolution with limited knowledge. In *Proceedings of COLING, Montreal, Canada*.

Mueller, C., & Strube, M. (2003). MMAX: a tool for the annotation of multi-modal corpora. In *Proceedings of the 2nd IJCAI workshop on knowledge and reasoning in practical dialogue systems, Seattle, US*.

Orasan, C., Mitkov, R., & Hasler, L. (2003). CAST: a computer-aided summarization tool. In *Proceedings of EACL, Budapest, Hungary*.

Poesio, M., & Kabadjov, M. A. (2004). A general-purpose, off-the-shelf anaphora resolution module: implementation and preliminary evaluation. In *Proceedings of LREC, Lisbon, Portugal*.

Poesio, M., Stevenson, R., Di Eugenio, B., & Hitzeman, J. M. (2004). Centering: a parametric theory and its instantiations. *Computational Linguistics, 30*(3).

Radev, D. R., Jing, H., & Budzikowska, M. (2000). Centroid-based summarization of multiple documents. In *ANLP/NAACL workshop on automatic summarization, Seattle, US*.

Steinberger, J., & Jezek, K. (2004). Text summarization and singular value decomposition. In *Proceedings of ADVIS, Izmir, Turkey*.

Steinberger, J., Kabadjov, M. A., & Poesio, M. (2005). Improving LSA-based summarization with anaphora resolution. In *Proceedings of HLT/EMNLP, The Association for Computational Linguistics, Vancouver, Canada* (pp. 1–8).

Stuckardt, R. (2003). Coreference-based summarization and question answering: a case for high precision anaphor resolution. In *International symposium on reference resolution, Venice, Italy*.

Vieira, R., & Poesio, M. (2000). An empirically-based system for processing definite descriptions. *Computational Linguistics, 26*(4).