

# Task-Based Evaluation of Anaphora Resolution: The Case of Summarization

**Mijail A. Kabadjov**  
University of Essex  
Colchester, UK  
malexa@essex.ac.uk

**Massimo Poesio**  
University of Essex  
Colchester, UK  
poesio@essex.ac.uk

**Josef Steinberger**  
University of West Bohemia  
Pilsen, Czech Republic  
jstein@kiv.zcu.cz

## Abstract

One of the types of semantic interpretation processes that may help ‘crossing the barriers’ in text summarization is anaphora resolution. In this paper, we show that summarization is a good task for evaluating the performance of an anaphoric resolver, in the sense that it encourages developing anaphoric resolvers that build good-quality discourse models, and the performance of the anaphoric resolver correlates well with the performance on the task. Two versions of the GUITAR anaphora resolution system, 1.1 and 2.1, were used in conjunction with a LSA-based summarizer; we demonstrate that whereas both versions of the system result in improvements over the pure LSA system, only the latest version, 2.1, leads to significant improvements.

## 1 Introduction

One of the types of semantic interpretation processes that may help ‘crossing the barriers’ in text summarization is anaphora resolution. Viceversa, using anaphoric resolvers at tasks such as summarization may help obtaining a better evaluation of their performance. Most evaluations of anaphora / coreference resolution systems, just like most evaluations of other NLP systems, are system-internal: the system’s ability to resolve anaphors is measured (Hobbs, 1978; Lappin and Leass, 1994; Mitkov,

1998; Vieira and Poesio, 2000; Ng and Cardie, 2002). Until five years ago, no other type of evaluation was possible, given the lack of performance in the prerequisite technologies (primarily parsing) and the lack of application systems performing NL interpretation tasks demanding enough to warrant investigating whether higher interpretation components such as anaphora resolution would improve their performance. This situation, has changed, however, with the development of high-performance parsers on the one hand, and with the growing interest in tasks such as information extraction, segmentation, and summarization, that more clearly may benefit from some sort of semantic interpretation. Yet, a certain degree of skepticism remains. The preliminary results of the project on which we are working suggest however that anaphora resolution *can* be useful for some tasks, provided that the performance of the anaphoric resolver is good enough; and that summarization is one task that may benefit. We report elsewhere that we successfully demonstrated that adding a high-performance anaphora resolution component does improve the performance even of a reasonably high-quality summarizer. In this paper, we show that summarization is a good task for evaluating the performance of an anaphoric resolver. Specifically, we show that whereas version 1.1 of our anaphora resolution system, GUITAR, only resulted in non-significant improvements to the performance of our summarizer, improving GUITARs’ performance led to significant improvements.

The structure of this paper is as follows. We begin by discussing why we think summarization may benefit from anaphora resolution. We then discuss

the Latent Semantic Analysis (LSA)-based summarizer we developed in previous work (Steinberger and Jezek, 2004). Next, we introduce the anaphoric resolver we are using, GUITAR (Poesio and Kabadjov, 2004), explain the difference between its versions 1.1 and 2.1, and discuss two ways in which the output of GUITAR may be used to improve the performance of a summarizer like the one we are using. Finally, we discuss how use of the different versions of GUITAR affected the performance of the summarizer.

## 2 Summarization and Anaphora Resolution

Many approaches to summarization can be very broadly characterized as TERM-BASED: they attempt to identify the main ‘topics,’ which generally are TERMS—expressions referring to objects—and then to extract from the document the most important information about these terms (Hovy and Lin, 1997). These approaches can be divided again very broadly in ‘lexical’ approaches, among which we would include LSA-based approaches, and ‘coreference-based’ approaches. Lexical approaches to term-based summarization use lexical relations to identify central terms, either directly over words using lexical chains (Barzilay and Elhadad, 1997) or by trying to identify ‘implicit topics’ by conflating together words using methods inspired by Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997), as done by Gong and Liu (2002).

Words are only the most basic type of ‘term’ that can be used to characterize the content of a document. Being able to identify the most important *objects* mentioned in the document clearly would lead to an improved analysis of what’s important in a text, as shown by the following news article cited by Boguraev and Kennedy (1999):

### (1) PRIEST IS CHARGED WITH POPE ATTACK

*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, understanding that Fernandez and the pope are the central characters is crucial to provide a good summary of texts like these.<sup>1</sup> Among the clues that help us to identify such ‘main characters’, the fact that an entity is repeatedly mentioned is clearly important.

Purely lexical methods, including the LSA-based methods we used in our own previous work (see next Section), only capture part of the information about which entities are frequently repeated. As example (1) 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 (*the Spaniard*) doesn’t 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.*).

Coreference- (or anaphora-) based approaches (Baldwin and Morton, 1998; Boguraev and Kennedy, 1999; Azzam et al., 1999; Bergler et al., 2003; Stuckardt, 2003) attempt to identify these repeatedly mentioned 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. In addition, to our knowledge no authors have convincingly demonstrated that feeding anaphoric information to a summarizer significantly improves the performance of a summarizer using a standard evaluation procedure (a reference corpus and baseline, and widely accepted evaluation measures).

In this work, we tested a mixed approach to in-

---

<sup>1</sup>It should be noted that for many newspaper articles, indeed 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 (Knott et al., 2001; Poesio et al., 2004).

tegrate anaphoric and word information: using the output of GUITAR to modify the SVD matrix used to determine the sentences to extract. We first discuss our previous work with LSA-inspired methods, then return to GUITAR and how we used its output to help summarization.

### 3 LSA-based Summarization

LSA (Landauer and Dumais, 1997) is a technique for extracting the ‘hidden’ dimensions of the semantic representation of terms, sentences, or documents, on the basis of their contextual use. It is a very powerful technique already used for NLP applications such as information retrieval (Berry et al., 1995) and text segmentation (Choi et al., 2001) and, more recently, multi- and single-document summarization.

The original version of the approach to using LSA in (sentence-extraction based) text summarization we followed in this paper was proposed by Gong and Liu (2002). Gong and Liu propose to start by creating a term by sentences matrix  $A = [A_1, A_2, \dots, A_n]$ , where each column vector  $A_i$  represents the weighted term-frequency vector of sentence  $i$  in the document under consideration. If there are a total of  $m$  terms and  $n$  sentences in the document, then we will have an  $m \times n$  matrix  $A$  for the document. The next step is to apply Singular Value Decomposition (SVD) to matrix  $A$ . Given an  $m \times n$  matrix  $A$ , the SVD of  $A$  is defined as:

$$(2) \quad A = U\Sigma V^T$$

where  $U = [u_{ij}]$  is an  $m \times n$  column-orthonormal matrix whose columns are called left singular vectors,  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$  is an  $n \times n$  diagonal matrix, whose diagonal elements are non-negative singular values sorted in descending order, and  $V = [v_{ij}]$  is an  $n \times n$  orthonormal matrix, whose columns are called right singular vectors. From a NLP perspective, what the SVD does is best explained as deriving the *latent semantic structure* of the document represented by matrix  $A$ : it identifies  $r$  linearly-independent base vectors (‘topics’) which specify the best joint index of terms and sentences contained in the original document.

A unique feature of SVD is that it is capable of capturing and modelling interrelationships among terms so that it can semantically cluster terms and sentences. 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. As each particular word combination pattern describes a certain topic in the document, each singular vector can be viewed as representing a salient topic of the document, and the magnitude of its corresponding singular value represents the degree of importance of the salient topic.

The summarization method proposed by Gong and Liu (2002) is a fairly straightforward application of SVD. The matrix  $V^T$  describes the importance degree of each ‘implicit topic’ in each sentence: the summarization process simply chooses the most informative sentence for each term. The  $k$ th sentence chosen is the one with the largest index value in the  $k$ th right singular vector in matrix  $V^T$ .

The original summarization method proposed by Gong and Liu has some disadvantages, the main of which is that it is necessary to use the same number of dimensions as is the number of sentences we want to choose for a summary. In order to remedy this problem, we proposed several modifications to Gong and Liu’s summarization method (Steinberger and Jezek, 2004). We showed in previous work (Steinberger and Jezek, 2004) that the modified algorithm results in significant improvements over Gong and Liu’s method; we will therefore use this algorithm as a baseline, called ‘Pure LSA’.

## 4 Using An Anaphoric Resolver to Help LSA-Based Summarization

### 4.1 GUITAR

The system we used in these experiments, GUITAR (Poesio and Kabadjov, 2004; Poesio et al., 2005), is an anaphora resolution system designed to be high precision, modular, and usable as an off-the-shelf component of a NL processing pipeline. GUITAR has been under development for about two years, during which period three versions of the system were developed, and tested with the summarizer dis-

cussed in the previous section, 1.1, 1.2, and 2.1.<sup>2</sup> We discuss each version in turn.

#### 4.1.1 GUITAR 1.1

The goal of the first version of GUITAR 1.1 was to have an anaphoric resolver implemented in Java which (i) would work in an incremental fashion taking XML as input and producing XML as output (ii) would resolve both pronouns and definite descriptions (iii) would make it very easy both to use different preprocessors and to replace some of the anaphora resolution modules. This latter goal was achieved by taking full advantage of Java's method inheritance structure, and by developing a DISCOURSE MODEL API specifying the methods to be used to access and modify the discourse model built by GUITAR. The first resolution modules included in the system were an implementation of the MARS pronoun resolution algorithm (Mitkov, 1998) and a partial implementation of the algorithm for resolving definite descriptions proposed by Vieira and Poesio (2000). This first version of the system included a preprocessor able to extract a representation in GUITAR's input format, MAS-XML, from the output of the LT-CHUNK partial parser.

**Personal Pronoun Resolution** Mitkov (1998) developed a robust approach to pronoun resolution which only requires input text to be part-of-speech tagged and noun phrases to be identified. Mitkov's algorithm operates on the basis of antecedent-tracking preferences (referred to hereafter as "antecedent indicators"). The system identifies the noun phrases which precede the anaphor within a distance of 2 sentences, checks them for gender and number agreement with the anaphor, and then applies genre-specific antecedent indicators to the remaining candidates (Mitkov, 1998). The noun phrase with the highest aggregate score is proposed as antecedent. As LT-CHUNK does not extract agreement features, the preprocessor developed for GUITAR 1.1 included a very basic agreement feature guesser.

**Definite Description Resolution** The Vieira / Poesio algorithm attempts to classify each definite description as either DIRECT ANAPHORA, DISCOURSE-NEW, or BRIDGING DESCRIPTION

(Vieira and Poesio, 2000). The Vieira / Poesio algorithm also attempts to identify the antecedents of anaphoric descriptions and the anchors of bridging ones. GUITAR 1.1 only incorporated an algorithm for resolving direct anaphora derived quite directly from Vieira / Poesio, without including any discourse-new detectors or any bridging reference resolution component.

#### 4.1.2 GUITAR 1.2

The first tests using GUITAR 1.1 as a preprocessor for our summarizer immediately revealed a number of problems with that version, resulting in two main changes. The first was the development of a new preprocessor able to use Charniak's full parser (Charniak, 2000) to produce the system's input format, MAS-XML. The second was a thorough revision of the agreement feature guesser included in the preprocessor, to obtain better defaults. This version was made available on the Web as GUITAR 1.2.

#### 4.1.3 GUITAR 2.1

A more substantial revision was the development of statistical methods for detecting discourse new descriptions (Poesio et al., 2005). In addition, further experiences with the summarizer identified two systematic problems with Charniak's parser's output: that it doesn't treat possessive pronouns as separate NPs (e.g., *his car* is treated as analyzed as a single NP, instead of as a possessive NP containing a separate pronominal NP) and that it analyzes NPs postmodified by PPs, such as *the expansion of the jeep plant*, as two distinct NPs separated by a preposition, as in:

[NP [NP *the expansion*] of [NP *the jeep plant*]]

rather than as NP postmodified by a PP, as in:

[NP *the expansion* [PP of [NP *the jeep plant*]]]

In order to correct these two problems, we included a correction component in the post-processor; this makes it possible for the system to resolve possessive pronouns as well. The new version of the system was called GUITAR 2.1.

#### 4.2 Using Anaphoric Information in Combination with SVD

SVD can be used to identify the 'implicit topics' or main terms of a document not only when on the basis

<sup>2</sup>Version 1.1 and 1.2 have both been made publically available. Version 2.1 will become available in the Summer.

of words, but also of coreference chains, or a mixture of both. We tested two ways of combining these two types of information.

#### 4.2.1 The Substitution Method

The simplest way of integrating anaphoric information with the methods used in our earlier work is to use anaphora resolution simply as a pre-processing stage of the SVD input matrix creation. Firstly, all anaphoric relations are identified by the anaphoric resolver, and anaphoric chains are identified. Then a second document is produced, in which all anaphoric nominal expressions are replaced by the first element of their anaphoric chain.

#### 4.2.2 The Addition Method

An alternative approach is to use SVD to identify ‘topics’ on the basis of two types of ‘terms’: terms in the lexical sense (i.e., words) and terms in the sense of objects, which can be represented by anaphoric chains. In other words, our representation of sentences would specify not only if they contain a certain word, but also if they contain a mention of a certain chain. This matrix is then used as input to SVD.

The chain ‘terms’ tie together sentences that contain the same anaphoric chain. If the terms are lexically the same (direct anaphors - like *deficit* and *the deficit*) the basic summarizer works sufficiently. However, Gong and Liu showed (and we made similar experiments) that the best weighting scheme is boolean. In this case all terms have the same weight. The advantage of the addition method is the opportunity to give higher weights to anaphors.

## 5 Evaluation

### 5.1 The CAST Corpus

To evaluate our system, we used the corpus of manually produced summaries created by the CAST project (Orasan et al., 2003). Most of the texts included in the CAST corpus are news articles taken from the Reuters Corpus; the rest are popular science texts from the British National Corpus. The annotated corpus contains information about the importance of the sentences (Hasler et al., 2003). Sentences are marked as **essential** or **important**. 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. 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.

### 5.2 Evaluation Measures

Evaluating summarization is a notoriously hard problem, for which standard measures like Precision and Recall are not very appropriate. The main problem with P&R is that human judges often disagree what are the top  $n\%$  most important sentences in a document or cluster and yet, there appears to be an implicit salience value for all sentences which is judge-independent. Using P&R creates the possibility that two equally good extracts are judged very differently. Because of these problems with precision and recall, we used a number of alternative evaluation measures. The first of these, relative utility (RU) (Radev et al., 2000) allows model summaries to consist of sentences with variable membership. With RU, the model summary represents all sentences of the input document with confidence values for their inclusion in the summary. To compute relative utility, a number of judges, ( $N \geq 1$ ) are asked to assign utility scores to all  $n$  sentences in a document. The top  $e$  sentences according to utility score<sup>3</sup> are then called a sentence extract of size  $e$ . We can then define the following system performance metric:

$$(3) \quad RU = \frac{\sum_{j=1}^n \delta_j \sum_{i=1}^N u_{ij}}{\sum_{j=1}^n \epsilon_j \sum_{i=1}^N u_{ij}},$$

where  $u_{ij}$  is a utility score of sentence  $j$  from annotator  $i$ ,  $\epsilon_j$  is 1 for the top  $e$  sentences according to the sum of utility scores from all judges and  $\delta_j$  is equal to 1 for the top  $e$  sentences extracted by the system. For details see (Radev et al., 2000).

The second measure we used is Cosine Similarity, according to the standard formula:

$$(4) \quad \cos(X, Y) = \frac{\sum_i x_i \cdot y_i}{\sqrt{\sum_i (x_i)^2} \cdot \sqrt{\sum_i (y_i)^2}},$$

<sup>3</sup>In the case of ties, some arbitrary but consistent mechanism is used to decide which sentences should be included in the summary.

Evaluation Method	Pure LSA	Manual Substitution	Manual Addition
F-score	0.420	0.410	<b>0.489</b>
Relative Utility	0.595	0.573	<b>0.662</b>
Cosine Similarity	0.774	0.806	<b>0.823</b>

Table 1: Evaluation of the improvement with manual annotation - summarization ratio: 15%.

Evaluation Method	Pure LSA	Manual Substitution	Manual Addition
F-score	0.557	0.549	<b>0.583</b>
Relative Utility	0.645	0.662	<b>0.688</b>
Cosine Similarity	0.863	0.878	<b>0.886</b>

Table 2: Evaluation of the improvement with manual annotation - summarization ratio: 30%.

where  $X$  and  $Y$  are representations of a system summary and its reference summary based on the vector space model. Finally, we measured ROUGE scores, with the same settings as in the Document Understanding Conference (DUC) 2004, and we did observe performance improvements with those as well, but we will not report them here for lack of space—see (Steinberger et al, submitted).

### 5.3 How Much May Anaphora Resolution Help? An Upper Bound

In order to establish an upper bound on the performance improvements that could be obtained by adding an anaphoric resolver to our summarizer, and to measure the performance of GUITAR over the 37 documents, we annotated all the anaphoric relations in the 37 documents by hand using the annotation tool MMAX (Mueller and Strube, 2003), and we tested both methods of adding anaphoric knowledge to the summarizer discussed above with this manual annotation. Results for the 15% and 30% ratios<sup>4</sup> are presented in Tables 1 and 2. The baseline is our own previously developed LSA-based summarizer without anaphoric knowledge. The result is that the substitution method did not lead to significant improvement, but the addition method did. For the 15% ratio, we found that addition led to improvements in 15 / 37 documents (40%), no changes with 18/37 (48.7%), and worse performance with 4/37 documents (10.8%), for an improvement in Relative Utility score from .595 to .662. For the 30% ratio, addition led to improvements in 46% of documents, no changes in 27%, and worse results for

24.3%, for a change in Relative Utility from .645 to .688. (Both improvements are significant.) Intuitively speaking, anaphoric information leads to improvements whenever the most important sentences are also those containing the most references to the 'main entities'; but it may also lead to worse results when those 'highly entity-cohesive' sentences are not considered particularly important by the summarizers.

### 5.4 Results with GUITAR

The performance results of the two versions of GUITAR we evaluated are presented next. GUITAR 1.1 achieved over the 37 documents  $P=50.4\%$  and  $R=33.8\%$ , for an  $F=40.5\%$ . By contrast, GUITAR 2.1 achieved  $P=50.6\%$ ,  $R=47.3\%$ , and  $F=48.9\%$ , over the 37 documents. The breakdown of figures for this last version is as follows: for definite description resolution, we found  $P=64.4\%$  and  $R=52.1\%$ ; for personal pronouns,  $P=44\%$  and  $R=46.3\%$ ; for possessive pronouns,  $P=39.3\%$  and  $R=39.7\%$ .

The results obtained by adding versions 1.1 and 2.1 of GUITAR to the summarizer are presented in Tables 3 and 4 (relative utility, f-score, and cosine). These results can be summarized as follows. First of all, addition works much better than substitution; in fact, with some metrics, substitution works worse than pure LSA. Secondly, anaphoric information, when used in the best way, does help summarization: except for 30% summarization and using cosine similarity, all versions of the summarizer using GUITAR with addition give better results than the version using SVD without coreference chains. However, only version 2.1 of GUITAR led to signifi-

<sup>4</sup>We used the same summarization ratios as in CAST.

Evaluation Method	CAST	Pure LSA	GUITAR1.1 Substitution	GUITAR1.1 Addition	GUITAR2.1 Substitution	GUITAR2.1 Addition
F-score	0.348	0.420	0.401	0.440	0.413	<b>0.445</b>
Relative Utility	0.527	0.595	0.575	0.620	0.591	<b>0.624</b>
Cosine Similarity	0.726	0.774	0.724	0.779	0.774	<b>0.789</b>

Table 3: Results with GUITAR - summarization ratio: 15%.

Evaluation Method	CAST	Pure LSA	GUITAR1.1 Substitution	GUITAR1.1 Addition	GUITAR2.1 Substitution	GUITAR2.1 Addition
F-score	0.522	0.557	0.532	0.551	0.540	<b>0.560</b>
Relative Utility	0.618	0.645	0.627	0.661	0.648	<b>0.663</b>
Cosine Similarity	0.855	<b>0.863</b>	0.839	0.851	0.858	0.857

Table 4: Results with GUITAR - summarization ratio: 30%.

cant improvements over Pure LSA according to what in our view is the best measure, Relative Utility (t-test,  $p < 0.05$ ). Using GUITAR 1.1 led to an improvement, but this was only significant at the 10% level.

Further analyses of our results (to be discussed at the workshop) showed, first of all, that GUITAR 2.1 manages to achieve an improvement in 9/17 (52.3%) of the documents for which we found an improvement using manual addition (30%), meaning that there is some room for improvement; and that what would help the most is adding to GUITAR named entity resolution. Secondly, we also saw that although many aspect of GUITAR's performance could be greatly improved, starting with pronoun resolution, this performance is still enough to lead to improvements; disabling pronoun resolution leads to worse results for the summarizer.

## 6 Conclusion and Further Research

One of the main results of this work is to show that using anaphora resolution in summarization can lead to significant improvements, not only when 'perfect' anaphora information is available, but also when an automatic resolver is used, and even if the performance of the anaphoric resolver is far from perfect. As far as we are aware, this is the first time that such a result has been obtained for summarization using standard evaluation measures over a reference corpus. We also showed however that improving the performance of GUITAR was essential in achieving significant improvement; and that the way in which anaphoric information is used matters. With our set of documents at least, substitution would not result in significant improvements even with perfect

anaphoric knowledge.

Further work will include, in addition to extending the set of documents and testing the system with other collections, evaluating the improvement to be achieved by adding a named entity resolution algorithm to GUITAR.

## References

- S. Azzam, K. Humphreys and R. Gaizauskas. 1999. Using coreference chains for text summarization. In *Proceedings of the ACL Workshop on Coreference*. Maryland.
- B. Baldwin and T. S. Morton. 1998. Dynamic coreference-based summarization. In *Proceedings of EMNLP*.
- R. Barzilay and M. Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL97/EACL97 Workshop on Intelligent Scalable Text Summarization*.
- S. Bergler, R. Witte, M. Khalife, Z. Li, and F. Rudzicz. 2003. Using Knowledge-poor Coreference Resolution for Text Summarization. In *Proc. of DUC*. Edmonton.
- M. W. Berry, S. T. Dumais and G. W. O'Brien. 1995. Using Linear Algebra for Intelligent IR. In *SIAM Review*, 37(4):573-595.
- B. Boguraev and C. Kennedy. 1999. Saliency-based content characterization of text documents. In I. Mani and M. T. Maybury (eds), *Advances in Automatic Text Summarization*, MIT Press, Cambridge, MA.
- E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of NAACL 2000*. Philadelphia, United States.
- F. Y. Y. Choi, P. Wiemer-Hastings and J. D. Moore 2001. Latent Semantic Analysis for Text Segmentation. In *Proceedings of EMNLP*. Pittsburgh.
- Y. Gong and X. Liu. 2002. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *Proc. of 24th ACM SIGIR*. New Orleans.

- L. Hasler, C. Orasan and R. Mitkov 2003. Building better corpora for summarization. In *Proceedings of Corpus Linguistics*. Lancaster, United Kingdom.
- J. Hobbs. 1978. Resolving Pronoun References. In *Lingua*, 44:311–338.
- Eduard Hovy and Chin-Yew Lin 1997. Automated text summarization in SUMMARIST. In *ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization*. Madrid, Spain.
- A. Knott, J. Oberlander, M. O’Donnell, and C. Mellish. 2001. Beyond elaboration: The interaction of relations and focus in coherent text. In Sanders, T., Schilperoord, J., and Spooren, W., editors, *Text representation: linguistic and psycholinguistic aspects*. John Benjamins.
- T. K. Landauer and S.T. Dumais. 1997. A solution to Plato’s problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. In *Psychological Review*, 104, 211-240.
- S. Lappin and H. J. Leass. 1994. An Algorithm for Pronominal Anaphora Resolution. In *Computational Linguistics*, 20(4):535–562.
- R. Mitkov. 1998. Robust pronoun resolution with limited knowledge. In *Proceedings of the 18th International Conference on Computational Linguistics* Montreal, Canada.
- C. Mueller and M. Strube 2001. 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.
- V. Ng and C. Cardie. 2002. Improving Machine Learning Approaches to Coreference Resolution. In *Proc. of the ACL*.
- C. Orasan, R. Mitkov and L. Hasler 2003. CAST: a Computer-Aided Summarization Tool. In *Proceedings of EACL2003 (Research Notes Session)*. Budapest, Hungary.
- M. Poesio and M. A. Kabadjov 2004. A General-Purpose, off-the-shelf Anaphora Resolution Module: Implementation and Pre-liminary Evaluation. In *Proceedings of LREC*. Lisbon.
- M. Poesio, R. Stevenson, B. Di Eugenio, and J. M. Hitzeman. 2004. Centering: A parametric theory and its instantiations. *Computational Linguistics*, 30(3).
- M Poesio, M. A. Kabadjov, R. Vieira, R. Goulart, and O. Uryupina. 2005. Do discourse-new detectors help definite description resolution? In *Proceedings of IWCS*. Tilburg.
- D. R. Radev, H. Jing, and M. Budzikowska. 2000. Centroid-based summarization of multiple documents. In *ANLP/NAACL 2000 Workshop*. Seattle, Washington.
- Josef Steinberger and Karel Jezek. 2004. Text Summarization and Singular Value Decomposition. In *Proceedings of the 3rd ADVIS conference*. Izmir, Turkey.
- R. Stuckardt. 2003. Coreference-Based Summarization and Question Answering: a Case for High Precision Anaphor Resolution. In *International Symposium on Reference Resolution*. Venice, Italy.
- R. Vieira and M. Poesio. 2000. An empirically-based system for processing definite descriptions. In *Computational Linguistics*, 26(4).