Using Knowledge-poor Coreference Resolution for Text Summarization

Sabine Bergler and **René Witte** and **Michelle Khalife** and **Zhuoyan Li** and **Frank Rudzicz**
Concordia University
bergler@cs.concordia.ca

Abstract

We present a system that produces 10-word summaries based on the single summarization strategy of outputting noun phrases representing the most important text entities (as represented by noun phrase coreference chains). The coreference chains were computed using fuzzy set theory combined with knowledge-poor corefernce heuristics.

1 Introduction

Imagine the following task: of a set of texts on a particular topic you need to select which one(s) to read based on 10-word indicative summaries of the texts. Summaries can be of any form.

This describes Task 1 of the NIST sponsored DUC 2003 competition. Our approach to this task is simple: we order the entities[1] in the text by importance to the text and output representative NPs until we reach the limit.

We approximate the importance of an entity to a text by the number of times it is referred to in that text, that is by the length of its corresponding noun phrase coreference chain.

In addition, we prefixed our summaries with a text category, generated using the classification tool *Bow* (McCallum, 1996) to supply some contextual information:

Source	**Summary**
ERSS	People: construction project, Schulz's work, voices, a repository, his "Peanuts" strip
Target	Charles Schultz to build museum to house his work.

While the idea of using the length of coreference chains is not novel to the summarization community (see (Brunn et al., 2001; Lal and Rüger, 2002) just for the last two DUC competitions), our approach is distinguished by its purity: no other technique is used to identify material for the summary. DUC evaluations show a surprising success of this single summarization principle: In a set of 15 systems manually evaluated for "usefulness" by external evaluators, our system placed above average.

The core engine behind the summarizer is a knowledge-poor noun phrase coreference system called Fuzzy-ERS,[2] based on ERS (Bergler, 1997), which is similar in spirit, but simpler than (Baldwin, 1997). Knowledge-poor heuristics by nature are less reliable and we chose to model the certainty of their results explicitly, using *fuzzy set theory* (Zadeh, 1987; Witte, 2002a).

Using fuzzy theory allows Fuzzy-ERS to simultaneously consider all coreference possibilities, even if this temporarily assigns a NP to more than one coreference chain (albeit with different coreference certainties). This means greater flexibility, because the same coreference heuristics can lead to a strict or lenient system based simply on the choice of cut-off threshold, which can vary for different uses.

We describe ERSS in detail below and evaluate its usefulness on the summarization task outlined above.

2 ERSS

Input to ERSS is a tagged text (using Mark Hepple's Brill-style POS tagger (Hepple, 2000)). The major components used are:

NPE, a noun phrase chunker that performs above 85%

Fuzzy-ERS, a coreference resolution system using fuzzy logic

[1] Events are part of the output if they are referred to by NPs, but since they frequently corefer to predicates, they do not usually achieve their proper place in this system.

[2] ERS stands for Experimental Resolution System.

Precision	min.	max.	average
strict	52.23%	72.15%	62.85%
average	64.33%	83.12%	75.00%
lenient	75.95%	94.09%	87.15%
Recall	min.	max.	average
strict	56.00%	80.00%	71.40%
average	74.00%	90.00%	85.20%
lenient	92.00%	100.0%	99.00%
F-measure	min.	max.	average
strict	57.44%	73.71%	66.85%
average	71.89%	84.91%	79.78%
lenient	85.41%	96.12%	92.70%

Table 1: Performance of the noun phrase extractor

Classifier, a naive Bayes classifier for multi-dimensional text categorization

ERSS, the summarization system

ERSS is implemented in the GATE architecture (Cunningham, 2002) and uses some of the ANNIE components and resources provided with GATE, as well as a classifier built with the *Bow* toolkit (McCallum, 1996) and WordNet (Fellbaum, 1998).

Noun Phrase Extractor. NPE uses a context-free NP grammar and an Earley-type chart parser to extract minimal noun phrases. Minimal noun phrases do not carry attachments, relative clauses, appositions, etc. Thus in our system *the president of the United States of America* generates three NPs, namely *the president, the United States,* and *America*.[3] The obvious setback of losing the semantics of this NP is offset by the fact that we avoid dealing with the ambiguity of PP attachment and have not compiled word lists for NPE.

The performance of NPE, evaluated with the GATE *Corpus Annotation Diff Tool* against a set of manually annotated texts, is shown in Table 1. Here, the *strict* measure considers all partially correct responses as incorrect, *lenient* regards all partially correct (overlapping) responses as correct, and the third column gives an *average* of both. The F-measure is computed with $\beta = 0.5$.

Parsing errors are mostly due to one of the following four anomalies (in order of importance): (i) wrong tagging, which leads to malformed noun phrases, (ii) selection of the wrong parse tree, (iii) parse tree explosion,[4] or (iv) insufficiency of the context-free grammar.

Fuzzy Coreferencer. Fuzzy-ERS groups the NPs extracted by NPE into *coreference chains*, ordered sets of NPs that refer to the same entity. ERS was initially conceived as a baseline system, operating with almost no knowledge sources. It considers definite and indefinite NPs, dates, amounts, and third person pronouns[5]. It is based on a few shallow heuristics which operate on the ordered set of NPs produced by NPE. The different heuristics are distinguished by their likelihood to produce a valid result: string equality is more likely to indicate correct coreference than matching only by head noun. In (Bergler, 1997) this was addressed implicitly by a specific ordering of the heuristics. Using fuzzy values now allows us an explicit representation of the certainty of each stipulated coreference: a NP is assigned to a coreference chain with a certain likelihood. To determine the final coreference chains, the system can now be biased: setting a threshold of 1 for chain membership essentially removes the fuzzy component from the system and results in very short, accurate coreference chains. Setting a more lenient threshold allows more NPs into the chain, risking false positives.

We describe the design and influence of the fuzzy values below.

Classifier. The classifier is a naive Bayes model trained on a number of small, focused ontologies (which we call *Micro-Ontologies*), implemented with the *Bow* toolkit (McCallum, 1996). Each of these ontologies focuses on a particular topical categorization (e.g., disasters and their subtypes); together, they give a multi-dimensional categorization of a text. For example, using three of these ontologies, a news article could be classified as {*Politics, People, Single-Event*} within a three-dimensional space.

The classification output does not correspond to the DUC requirements and had no correlates in the target summaries. It pushed us over the 10 word limit and thus penalized the system on the length-adjusted coverage count. This is why we only consider coverage here, because leaving off the text category is trivial. Yet we feel strongly that the text classification does add to the picture of the text analyzed. We cannot, however, determine whether it had any influence on the usefulness score.

Summarizer. The summarizer is based on the simple idea that a 10-word summary should mention the most important entities of the text. We stipulate that the most important entities of a newspaper text are usually the ones corresponding to the longest noun phrase coreference chains. Thus, for the summarization, all chains are *ranked*. The longest chain usually receives the highest rank, but the ordering is additionally influenced by a

[3] We repair some of this by using the named entity (NE) recognition component from ANNIE, which resolves *the United States of America* to a single named entity before it is fed to NPE.

[4] We abort parsing when a certain number of parse trees are exceeded.

[5] Pronoun resolution is inspired by (Hobbs, 1978; Lappin and Leass, 1994) but since we do not parse the entire sentence our algorithm is much cruder.

boosting factor that promotes chains with NPs that also occur in the first two sentences. Currently, we choose the longest NPs as representatives for the longest chains.

Thus, our summarization strategy can be summarized as follows:

1. output the most salient text classification with a simple decision-tree algorithm to provide come context
2. sort the coreference chains according to their ranking
3. select the longest noun phrase from each chain
4. output NPs as long as the length limit (10 words for the DUC 2003 Task 1) has not yet been reached.

3 Fuzzy Noun Phrase Coreference Resolution

The core idea for using a fuzzy-theory based resolution algorithm is the realization that coreference between noun phrases can neither be established nor excluded with absolute certainty. While statistical methods employed in natural language processing already model this *uncertainty* through probabilities, non-statistical methods that have been used so far had no systematic, formal representation for such imperfections. Instead, weights or biases are derived experimentally or through learning algorithms (Cardie and Wagstaff, 1999). Here, uncertainty is implicitly and opaquely dealt with in the system and changing it requires rebuilding the system or training set.

Our approach is to examine *explicit* representation and processing models for uncertainty based on fuzzy set theory (Zadeh, 1987; Klir and Folger, 1988; Cox, 1999). There are several advantages in explicitly modelling uncertainty: we do not have to choose arbitrary cut-off points when deciding between "corefering" and "not corefering," like for the semantic distance between words. Instead of such an a priori decision to be lenient or restrictive, we can dynamically decide on certainty thresholds to suit different processing contexts and this value itself can become part of the system deliberations.

As a consequence, we have more information available when building coreference chains, improving overall performance. Moreover, it is now possible to use the same result in different contexts by requesting a specific coreference certainty: a summarizer, for example, can decide to select only coreferences with a high certainty, while a full-text search engine might allow a user to retrieve information based on a more lenient certainty degree.

Our fuzzy noun phrase coreference resolution algorithm is based on the system described in (Bergler, 1997), but has been completely rewritten with the fuzzy-theory based representation model presented in (Witte, 2002a; Witte, 2002b). We now describe the fuzzy resolution algorithm in detail; we start with the representation model for fuzzy coreference chains, then describe the fuzzy resolution algorithm and its resources, and finally show how the computed fuzzy coreference chains can be converted into classical, crisp chains.

3.1 Modeling Fuzzy Coreferences

Fuzzy coreference chains are the basic representational unit within our fuzzy resolution algorithm. A single *fuzzy chain* $\mathscr{C}$ is represented by a fuzzy set $\mu_{\mathscr{C}}$, which maps the domain of all noun phrases in a text to the $[0,1]$-interval. Thus, each noun phrase np_i has a membership degree $\mu_{\mathscr{C}}(\mathrm{np}_i)$, indicating how certain this NP is a member of chain $\mathscr{C}$. The membership degree is interpreted in a possibilistic fashion: a value of 0.0 *("impossible")* indicates that the NP cannot be a member of the chain, a value of 1.0 *("certain")* means that none of the available information opposes the NP from being a member of the chain (*not* that it must be a member!), and values in between indicate varying degrees of compatibility of a noun phrase with the chain.

Example (Fuzzy Coreference Chain) Figure 1 shows an example for a fuzzy coreference chain. Here, the noun phrases np_3 and np_6 have a very high certainty for belonging to the chain, np_1 only a medium certainty, and the remaining NPs are most likely not chain members.

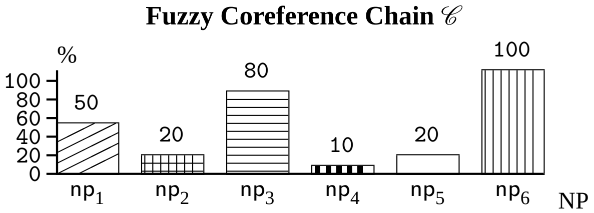


Figure 1: Example for a fuzzy chain showing the membership grades for each noun phrase

The output of our coreference algorithm is a set of fuzzy coreference chains, similar to classical resolution systems. Each chain holds all noun phrases that refer to the same conceptual entity. However, unlike for classical, crisp chains, we do not have to reject inconsistent information out of hand, so we can admit a noun phrase as a member of more than one chain, with a varying degree of certainty for each. This will be discussed later in more detail. We first show how fuzzy chains are constructed through *fuzzy heuristics*.

3.2 Fuzzy Heuristics

The fuzzy resolution system contains a number of heuristics[6] for establishing coreference, each focus-

[6] The heuristics used are a subset of the ones motivated in (Bergler, 1997).

ing on a particular linguistic phenomenon. Examples for fuzzy heuristics are pronominal coreference, synonym/hypernym-coreference, or substring coreference.

Formally, a fuzzy heuristic $\mathscr{H}_i$ takes as input a noun phrase pair $(\mathrm{np}_j, \mathrm{np}_k)$ and returns a fuzzy set $\mu^{\mathscr{H}_i}_{(\mathrm{np}_j,\mathrm{np}_k)}$ that indicates the certainty of coreference for the noun phrase arguments.

Such a certainty degree can be intuitively determined for almost all heuristics: an example is the synonym/hypernym heuristic, which has been implemented with WordNet (Fellbaum, 1998). Here, we assume two NPs that are synonyms corefer *certainly*, hence they are assigned a degree of 1.0. For hypernyms, our certainty decreases linearly with increasing semantic distance (we are currently evaluating different measures for semantic distance).

The *design* of fuzzy heuristics brings new challenges to the system developer, however, since the uncertainty of a coreference must now be modeled explicitly. Our experiences show that this requires an additional initial effort, as uncertainty management and fuzzy set theory are not commonly used tools in computational linguistics. The start-up effort is worthwhile, though, since fuzzy heuristics turned out to be easier to design (no mismatch between uncertain reality and computer model) and more powerful (retaining more information) than their classical, non-fuzzy counterparts.

3.3 Building Fuzzy Chains

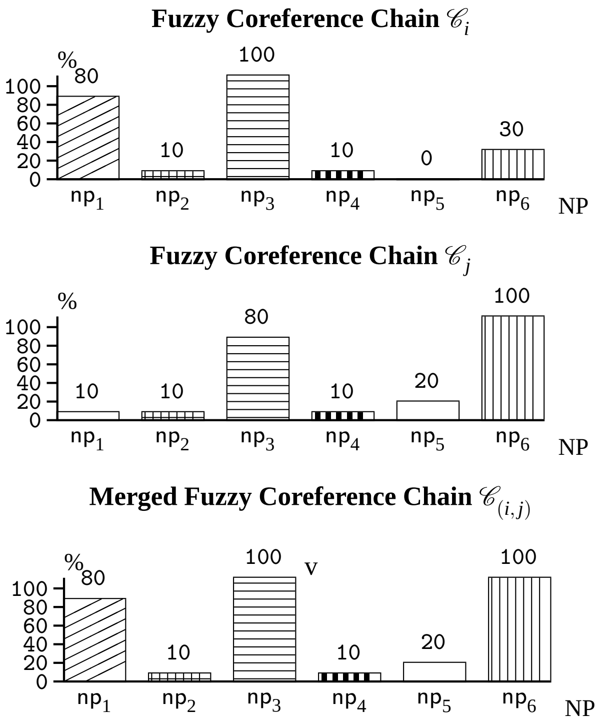
The first step in the fuzzy coreference algorithm is the construction of *fuzzy chains*, holding the possibilities of coreference represented by certainty degrees as described above. This is achieved by applying all fuzzy heuristics to each noun phrase pair and computing the logical fuzzy-or function over all individual results.

In a first step we build as many fuzzy chains as there are noun phrases in a text. Each noun phrase is a member of each chain, but usually with varying degrees of certainty.

For the final result, however, we are interested in compiling all possible coreferences concerning a given NP into a single coreference chain. This is achieved through a merging algorithm assuming that coreference is symmetric and transitive.

3.3.1 Merging Fuzzy Chains

All coreference possibilities concerning a noun phrase np_i are described in the fuzzy set $\mu_{\mathscr{C}_i}$, which constitutes an incomplete fuzzy coreference chain. Since the coreference relation is symmetric and transitive, if $\mathscr{C}_1$ establishes a coreference of e.g. np_1 and np_3 (with some certainty) and likewise $\mathscr{C}_2$ for np_3 and np_5, we expect the final result to also show a coreference for np_1 and np_5 in the same chain.


Fuzzy Coreference Chain $\mathscr{C}_i$

Fuzzy Coreference Chain $\mathscr{C}_j$

Merged Fuzzy Coreference Chain $\mathscr{C}_{(i,j)}$

Figure 2: Merging two fuzzy coreference chains with $\gamma = 0.75$

This is achieved by the process of *merging* the incomplete fuzzy chains into a set of complete chains where each chain holds all references to a single entity with a given certainty, prescribed by a *consistency* parameter γ, which is a threshold value for inclusion of a coreference possibility into the merged chain. Here, the consistency of a fuzzy coreference chain $\mathscr{C}$ is defined as the consistency (maximum value) of its corresponding fuzzy set $\mu_{\mathscr{C}}$. In order for a reference chain $\mathscr{C}_i$ to reach a consistency degree of at least γ, there has to be at least one noun phrase np_j in this chain with $\mu_{\mathscr{C}_i}(\mathrm{np}_j) \geq \gamma$ (note that every noun phrase corefers with itself to a degree of 1.0, so all initial chains $\mu_{\mathscr{C}_i}$ created by the algorithm above also have a consistency degree of 1.0). Thus, two chains are merged if their fuzzy set intersection[7] reaches at least the requested consistency degree γ.

A simple chain merging algorithm examines all possible chain combinations given a degree γ and returns a list of merged fuzzy chains.

Example (Chain Merging) An example for the merging of two chains is shown in Figure 2. Here, a single new chain $\mathscr{C}_{(i,j)}$ (bottom) has been formed out of the two

[7] We use the standard functions for possibilistic fuzzy sets, that is *min* for intersection, *max* for union, and $1 - \mu$ for computing the complement.

chains $\mathscr{C}_i$ and $\mathscr{C}_j$ (top) given a degree of $\gamma = 0.75$. If we had asked for a consistency degree of $\gamma = 1.0$, however, the chains would not have been merged since the consistency degree of both fuzzy sets' intersection is only 0.8.

With this algorithm, we can directly influence the result by changing the required consistency degree for an output chain; a degree of 1.0 corefers only 100% certain[8] NP pairs, a degree of 0.0 would corefer all NPs into a single chain, and degrees in between result in chains of varying NP clusters according to their coreference certainty. The cut-off value γ influences the results of ERSS directly (for the DUC 2003 ten word summary, we used the empirically chosen consistency degree of 0.6).

3.3.2 Defuzzification

Most of our existing processing resources have not yet been "fuzzified," hence, they still expect classical, crisp coreference chains. For these components we have to *defuzzify* our fuzzy chains.

We chose a simple defuzzification function: a crisp reference chain contains exactly the noun phrases having a membership degree of at least γ.

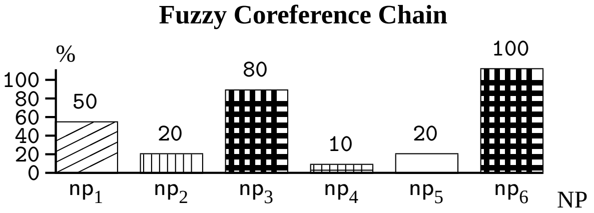


Figure 3: Defuzzification Example

Example (Defuzzification) An example is shown in Figure 3. With a certainty degree of $\gamma = 0.8$ we get the crisp result set $c = \{\text{np}_3, \text{np}_6\}$.

3.4 Performance of the fuzzy coreference resolution algorithm

The performance of the fuzzy coreference algorithm depends largely on two factors: the quality of the implemented heuristics (and their available resources) and the properties and settings of the fuzzy algorithm itself. We only analyze the second component here, assuming a given set of fuzzy heuristics.[9]

The fuzzy coreference algorithm described above produces a similar result to its non-fuzzy counterpart when run with a consistency degree of 1.0.[10] However, with this algorithm we now gained the ability to explicitly request coreference results with different degrees of certainty.

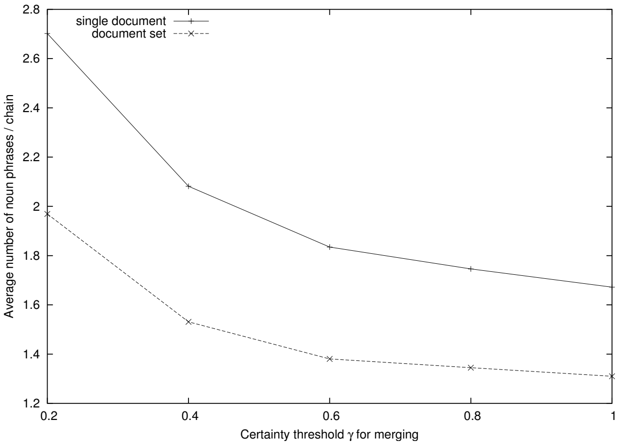


Figure 4: Different fuzzy values γ result in chains of different lengths

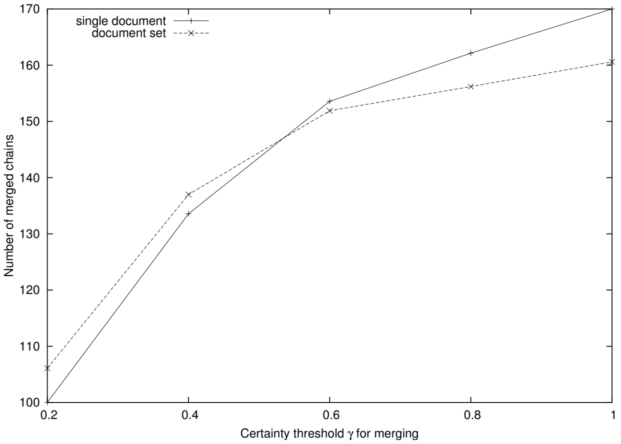


Figure 5: Number of resulting (merged) chains depends on the fuzzy value γ

The decisive parameter here is the consistency parameter γ used for merging, effectively determining how certain a coreference must be to be admitted in a chain. Higher γ-values lead to a greater number of shorter chains that have a higher certainty of coreference between its NPs at the expense of completeness. Lower γ-values, in turn, result in fewer and longer chains, but might contain wrongly coreferred NPs.

This intuitive understanding of the fuzzy algorithm's behaviour has been experimentally confirmed during our evaluations for DUC 2003. Figure 4 shows how different settings for the certainty threshold γ used in the merging phase of the algorithm influence the resulting chains: the lower the requested certainty, the more chains are

[8] Under a closed world assumption the degree of consistency corresponds to a degree of certainty.

[9] For alternative sets of heuristics see (Baldwin, 1997; Kameyama, 1997; Harabagiu and Maiorano, 1999).

[10] Of course, if we didn't want to exploit fuzzy theory we would have written the algorithm differently and thus the comparison is only illustrative.

> power transmission project, *the Three Gorges*, the power transmission project, the Three Gorges Project, *the Three Gorges Project Office*, the entire power transmission project, the Three Gorges Project, the Three Gorges Dam-Wangxian-Changshou-Chongqing power transmission project, the Three Gorges power transmission project, *contracting projects*, the Three Gorges Project

Figure 6: Example coreference chain from ERSS: NPs which do not belong in the chain are highlighted in italics.

merged, resulting in longer output chains (shown here are values for a single document containing 433 recognized noun phrases and values that were averaged over a 10-document set). Likewise, Figure 5 shows how the number of resulting chains decreases with a decreasing certainty threshold.

As can be seen, a fuzzy value of 0.2 results in comparatively long chains containing a higher average number of NPs. An empirical evaluation showed that these chains are not very useful, however, since they contain many wrong coreferences (after all, a certainty of 20% is not very high). Likewise, coreference chains with a certainty of 1.0 tended to be too fragmented for our intended application, automatic summarization. Intermediate fuzzy values lead to good coreference chains that produce useful results, as we will show below.

4 Evaluation for Summarization

Fuzzy-ERS works with very knowledge-poor techniques depending solely on isolated minimal NPs. It is thus much less sophisticated than other NP coreference systems. Because of the direct influence of γ on precision and its inverse relationship with recall, we chose to evaluate the usefulness of fuzzy theory for coreference resolution based summarization.

We evaluate Fuzzy-ERS on 10 word summaries. With $\gamma = 0.6$ we include some very inaccurate NPs in chains, especially the WordNet derived distance measure is very permissive at that value. Yet the benefit of overcoming the chain fragmentation of higher thresholds still outweighs the imprecision of some chains.

NIST assessors evaluated ERSS summaries against manually constructed target summaries of different styles: Some were single sentences, some multiple sentences, some resembled our output very closely and some mixed the other styles. This was a feature of this year's target summaries: not to penalize a system too much for stylistic differences, NIST had four summaries prepared for each text and selected one at random for the target summary.

	Documents			Directories		
	min	max.	avg.	min.	max.	avg.
Recall	0	100	44.7	26	71	48.5
Precision	0	100	44.5	26	69	47.5
F-measure	0	100	42	26	62	44

Table 2: Performance of ERSS over 264 Documents in 60 Directories

ERSS was judged to give relevant summaries in 83% of the cases (its summary being a marked unit). Usefulness was judged average at 1.82 over a scale from 0 (bad) to 4 (excellent) (the overall average for Task 1 was 1.84). We share the feeling that our output is almost "so-so," as it often misses the most salient terms, but gives often a surprisingly coherent overview over the article.

Coverage overall was judged at 29% by the NIST assessors. Here our system had two consistent penalties: one, that it was consistently above the threshold of 10 words, because we added the classifier output to our 10-word summaries rather than include it. This is why we do not consider the length adjusted scores here (we consistently overstep the threshold by only two to three words which do not get any consideration in the coverage score.) The second penalty arises from the different format of the target summaries. The structuring information that a headline or full sentence can give in the predicate and the relating of two terms eludes ERSS, but are of course part of the coverage scoring process.

We also evaluated ERSS manually on the same target summaries. To compare our output with the target summary, we choose to split the target into *concept-tokens* (CTs), where tokens could be single nouns, noun phrases and possibly verbs. CTs are thus similar to and comparable with ERSS's output.

Any CT that matches against an output NP counts as one hit. We do not count or compare with the output of the classifier, since the document type information given by our classifier is not present in the target summaries.

The match can be partial, 'Asian Games' and 'Second Asian Games' count as a hit, as does 'drug trade' and 'China's major drug problem,' where we have a common "drug-problem" concept. Since we assume that the general subject of the document collection is known, this is justifyable, while in general it is not.

Once concept-tokens are matched against ERSS's NPs, recall and precision are measured, and consequently the F-measure. Table 2 shows the average values for recall, precision, and the F-measure for the summary comparison. We provide maximal and minimal values to indicate the spread of values.

No hits happen when either ERSS returns the general event such as 'International Human Rights Treaty,' while the manual summary goes more into details and is about an ‘arrest,’ or ERSS and the manual summary each cover a distinct idea in the text, and we get ‘Bad weather’ vs. ‘No Satellite Damage,’ or for the “SwissAir Flight 111” example, ‘the dead’ vs. ‘the plane’s wreckage’. On the other hand, in 4% of texts we have optimal recall (100%) spread over 14 different directories. The manual summaries in this case are short and headline-like. When it’s the other way round, i.e. ERSS returns 2 to 3 NPs, precision is at its best. This occurs 3.5% of the time over 7 directories.

This clustering of excellent recall in only 14 directories, and best precision in 7, leads us to the intuition that our approach is more suitable to certain text styles than others. Stipulating that texts with the same subject area are more alike, we investigated (unsuccessfully, so far) possible correlations between usefulness, coverage, and our own recall and precision measures. At this point we can illustrate this idea by listing the texts that received a usefulness rating of 4 from at least one assessor:

(D30028 APW199810030646 T 25 C 4 F 4)
(D30028 APW199810040175 T 25 C 3 F 4)
(D30040 APW199812300983 T 25 B 2 C 4)
(D30050 NYT199810070352 T 25 C 4 J 1)
(D31002 APW199810030170 T 25 B 4 I 3)
(D31002 APW199810030180 T 25 B 4 I 2)
(D31002 APW199810030470 T 25 B 4 I 2)
(D31002 APW199810030473 T 25 B 4 I 2)
(D31002 APW199810030492 T 25 B 4 I 2)
(D31027 APW199810180638 T 25 A 1 C 4)
(D31050 APW199812030338 T 25 C 4 E 2)

We see that one directory has over 50% of these files. We will investigate this matter further.

4.1 Automatic Evaluation

Manual evaluation being notoriosly laborious and error-prone, we started to experiment with several automatic measures for determining the performance of our system. Under the assumption that the *“usefulness”* manually determined by the DUC assessors gives an accurate description of a system’s performance, we are especially interested in finding an automatic measure that has a high correlation with the usefulness score. This is especially interesting for our fuzzy system, as it would enable us to compare the output for different certainty settings in a less empirical fashion.

Our first approach was to use the NAMS scoring function proposed by (Lin and Hovy, 2002). In their evaluation, it showed a performance exceeding 97% when compared to the manual system ranking using the Spearman rank-order correlation coefficient. Table 3 shows the results of applying the NAMS method for ranking the summarizer output compared with the NIST manually-determined usefulness ranking, our own manual ranking as described above, as well as the automatically determined coverage and length-adjusted coverage provided by NIST. Unfortunately, we were not able to find a correlation of more than 50%, which obviously is not good enough for the intended automatic evaluation. This is probably due to the much more restricted content of the 10-word summaries examined here, while the summaries examined by Lin and Hovy ranged from 50–400 words. Additionally, target summary style varied greatly, skewing the performance evaluation outcome depending on the (randomly) chosen target style, while not necessarily indicating a real difference in the performance of our system. And finally, the opinion of the manual assessor varied greatly, differing by as much as 3 points on the 4-point scale used for measuring usefulness.

	NAMS	combined NAMS
Coverage	0.50	0.60
avg. Usefulness	0.34	0.34
avg. Precision	0.46	0.41
avg. Recall	0.52	0.46
avg. F-Measure	0.52	0.44

Table 3: Spearman correlation of the automatically computed NAMS score compared to different automatic and manual rankings

5 Conclusions and further work

ERSS’s performance validates our approach: coreference resolution is part of the known toolkit for summarization. Yet a system that uses as its single summarization strategy the length of NP coreference chains performs average. This is a strong endorsement for Fuzzy-ERS and the idea that most cited discourse entities give an ok summarization of a text. We will improve Fuzzy-ERS to achieve better coreference resolution and we will embed it in a more sophisticated environment.

Another line of further investigations will study the features of the texts with good scores and compare and contrast them with those getting bad scores.

References

Breck Baldwin. 1997. Cogniac: High precision coreference with limited knowledge and linguistic resources. In *Proceedings of the ACL’97/EACL’97 Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 38–45, Madrid, Spain, July.

Sabine Bergler. 1997. Towards reliable partial anaphora resolution. In *Proceedings of the ACL’97/EACL’97 Workshop on Operational Factors in Practical, Robust*

Anaphora Resolution for Unrestricted Texts, Madrid, Spain, July.

M. Brunn, Y. Chali, and C.J. Pincha. 2001. Text summarization using lexical chains. In *Document Understanding Conference (DUC)*, New Orleans, Louisiana USA, September 13-14, 2001.

Claire Cardie and Kiri Wagstaff. 1999. Noun phrase coreference as clustering. In *Proceedings of the Joint Conference on Empirical Methods in NLP and Very Large Corpora*, Maryland.

Earl Cox. 1999. *The Fuzzy Systems Handbook*. AP Professional, 2nd edition.

H. Cunningham. 2002. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36:223–254. `http://gate.ac.uk`.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Sanda Harabagiu and Steven Maiorano. 1999. Knowledge-lean coreference resolution and its relation to textual cohesion and coherence. In *Proceedings of the ACL'99 Workshop on the Relation of Discourse/Dialogue Structure and Reference*, pages 29–38, University of Maryland, June.

Mark Hepple. 2000. Independence and commitment: Assumptions for rapid training and execution of rule-based pos taggers. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, Hong Kong, October.

Jerry R. Hobbs. 1978. Resolving pronoun references. *Lingua*, 44:311–338.

Megumi Kameyama. 1997. Recognizing referential links: An information extraction perspective. In *Proceedings of the ACL'97/EACL'97 Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, Madrid, Spain, July.

George J. Klir and Tina A. Folger. 1988. *Fuzzy Sets, Uncertainty, and Information*. Prentice-Hall.

P. Lal and S. Rüger. 2002. Extract-based summarization with simplification. In NIST, 2002 (NIS, 2002).

Shalom Lappin and Herbert Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–562.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In NIST, 2002 (NIS, 2002).

Andrew Kachites McCallum. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. `http://www.cs.cmu.edu/~mccallum/bow`.

NIST. 2002. *DUC 2002 Workshop on Text Summarization*, Philadelphia, Pennsylvania, USA, July 11-12.

René Witte. 2002a. *Architektur von Fuzzy-Informationssystemen*. BoD. ISBN 3-8311-4149-5.

René Witte. 2002b. Fuzzy belief revision. In *9th Intl. Workshop on Non-Monotonic Reasoning (NMR'02)*, pages 311–320, Toulouse, France, April 19–21. `http://rene-witte.net`.

L.A. Zadeh. 1987. Fuzzy sets. In R.R. Yager, S. Ovchinnikov, R.M. Tong, and H.T. Nguyen, editors, *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, pages 29–44. Wiley&Sons. Originally published in *Information and Control*, Vol. 8, New York: Academic Press, 1965, pages 338–353.