# Context-based Multi-Document Summarization using Fuzzy Coreference Cluster Graphs

**René Witte** and **Ralf Krestel**
Faculty of Informatics
Institute for Program Structures
and Data Organization (IPD)
Universität Karlsruhe, Germany
witte|krestel@ipd.uka.de

**Sabine Bergler**
The CLaC Laboratory
Department of Computer Science
and Software Engineering
Concordia University, Montréal, Canada
bergler@cs.concordia.ca

## Abstract

Constructing focused, context-based multi-document summaries requires an analysis of the context questions, as well as their corresponding document sets. We present a fuzzy cluster graph algorithm that finds entities and their connections between context and documents based on fuzzy coreference chains and describe the design and implementation of the ERSS summarizer implementing these ideas.

## 1 Introduction

These days, *finding* information is a much less challenging task than *processing* it, thanks to the success of online text databases, information retrieval, and web services. The original idea of automatic summarization—reducing the amount of information a user has to process by compressing it—can help to alleviate this problem, but in the worst case just adds even more information to the original content.

A better solution perhaps is to turn the problem around: nobody really wants to spent hours on *Google* searching for potentially relevant information. What users need is useful information pertaining to their task at hand, like writing a report, an email, or a research paper. Shouldn't a system be able to sense a user's current *context*, search for relevant information by itself, and present a summary thereof? Coupled with current information retrieval techniques and intelligent information system architectures (Witte, 2004), a new generation of language-aware information systems could proactively deliver the information users need, instead of requiring them to spend their limited time searching for them.

It is this vision that motivates our work in context-based summarization in general and the context task within the Document Understanding Conference (DUC) competition in particular. Although DUC covers only a part of the outlined vision—namely, summarization of a prescribed document set based on an explicitly stated context—this is nevertheless a core component covering one of the central enabling technologies for a language-aware information system.

**DUC 2006: Context-based Multi-Document Summarization.** Like in 2005 (NIST, 2005), the DUC 2006 competition included a single task: the generation of a focused 250-word summary based on a context, which typically comprised a set of questions that *"model real-world complex question answering, in which a question cannot be answered by simply stating a name, date, quantity, etc."* (NIST, 2006). From the task description: *"Successful performance on the task will benefit from a combination of IR and NLP capabilities, including passage retrieval, compression, and generation of fluent text."*

## 2 Building Focused Summaries

Our summarization system's main resources are intra- and inter-document coreference chains, which are computed using fuzzy set theory as the underlying representational formalism, hence *fuzzy coreference chains*. These fuzzy chains are then clustered, generating a cluster graph data structure based on which summaries are generated.

We originally presented the ERSS system implementing these ideas for very short (10 word) single-document summaries (Bergler et al., 2003), followed by an enhanced version in 2004 for multi-document summaries (Bergler et al., 2004), which in turn was further improved for the context task in DUC 2005 (Witte et al., 2005). In this paper, we provide a complete description of our clustering strategy as applied to focused summarization for DUC 2005–2006.

### 2.1 Summarization Strategy Overview

Our summarization system is based on generating and clustering coreference chains using fuzzy set theory. We compute both inter- and intra-document coreference chains, which together allow us to find important entities within a document and across documents.

For focused summaries, based on a set of questions, we consider the context as yet another document within a cluster when computing cross-document coreference chains. This allows us to identify information within and across documents that are semantically connected with one or multiple question(s) from the context.

Sentences are then extracted based on a scoring and ranking scheme and assembled into a multi-document summary, with only light postprocessing performed on each sentence. The main components of our system are:

**Preprocessing:** A number of preprocessing components perform tokenization, gazetteering (marking tokens with semantic labels based on lists like person names, locations, or companies), abbreviation detection, quote recognition, and sentence splitting. For these tasks, we use slightly modified versions of the tools that come with the ANNIE system, which is part of GATE (Cunningham et al., 2002).

**POS Tagger:** Part-of-speech tagging is performed by the Hepple tagger (Hepple, 2000) included in the GATE distribution.

**Named Entity (NE) Transducer:** A multi-stage JAPE[1] transducer, which is also based on the ANNIE system that comes with GATE, identifies several named entity types, like Person, Organization, Location, Number, and Date information.

**NP/VG Chunker:** Noun phrase (NP) chunking is performed in two steps; firstly, base NPs are generated using the MuNPEx chunker (Witte, 2006). And secondly, "long NPs" are generated based on some prepositional and conjunctional patterns. Verb groups are computed using the VG chunker module that comes with the GATE distribution (Cunningham et al., 2002).

**Fuzzy Coreferencer:** This component builds intra- and inter-document fuzzy coreference chains, as described in Section 2.2 below.

**Summarizer:** This is our summarization framework, which allows for pluggable summarization *strategies*, described in more detail below. Coreference cluster graphs are computed and summaries generated based on the results of the upstream components.

We now briefly review fuzzy coreference resolution (Section 2.2) and then describe our main algorithm for generating the data structure needed for building summaries, fuzzy coreference cluster graphs (Section 2.3). Finally, the generation of focused summaries from this data structure is covered in Section 2.4.

[1] JAPE is a regular-expression based language for writing grammars over annotations, from which (non-deterministic) transducers can be generated by a GATE component.

## 2.2 Fuzzy Coreferences

As mentioned above, the main resource of our system are *fuzzy coreference chains*. "Fuzzy" here refers to fuzzy set theory (Klir and Folger, 1988), which forms the formal basis for coreference resolution algorithms based on *fuzzy clustering*. Fuzzy coreference resolution is a rather new approach that differs from the classical rule-based or statistical algorithms.

For the purpose of this paper, we will only give a brief outline of fuzzy coreference resolution. Our approach is described in more detail in (Witte and Bergler, 2003; Witte, 2002).

Fuzzy coreference resolution algorithms work on entities and build fuzzy coreference chains using clustering algorithms.[2] The central idea is to use fuzzy set theory as the formal representation model for entity resolution. This immediately allows to apply research results and algorithms from the well-understood area of *fuzzy clustering* to computational linguistics. A common property of all fuzzy clustering algorithms is the use of soft thresholds within the clustering process, allowing a run-time tradeoff between precision (fewer connections between entities, smaller clusters) and recall (larger clusters, extraneous links).

### 2.2.1 Fuzzy Coreference Chains

Fuzzy coreference chains link entities, which are typically represented by noun phrases (NPs). In this paper, we denote the set of all noun phrases within a text with the (crisp) set $\text{ALLNP} = \{\text{np}_1, \ldots, \text{np}_m\}$, i.e., there are $m$ noun phrases within a document. A single fuzzy coreference chain $c$ is then represented by a fuzzy set $\mu_c$, which maps the domain of all noun phrases ALLNP to the $[0,1]$-interval: $\mu_c : \text{ALLNP} \rightarrow [0,1]$. Thus, each noun phrase $\text{np}_i \in \text{ALLNP}$ has a membership degree $\mu_c(\text{np}_i)$, indicating how certain this NP is a member of chain $c$. The membership degree for a single noun phrase $\mu_c(\text{np}_i) \in [0,1]$ is interpreted in a possibilistic fashion: a value of 0.0 *("impossible")* indicates that the NP cannot be a member of chain $c$, a value of 1.0 *("100% possibility" or "certain")* means that none of the available information indicates that the NP is not in the chain, and intermediate values represent different degrees of compatibility of a noun phrase with the chain.

Note that we can apply the same basic idea to cluster verb groups (VGs) within and across documents, which we will discuss in the summarization section below.

**Example (fuzzy coreference chain).** Figure 1 shows an example for a fuzzy coreference chain $c$. Here, the

[2] In this paper, we denote coreference chains computed by fuzzy coreference clustering algorithms as *fuzzy coreference chains* (and not fuzzy clusters as commonly done in the literature), to avoid confusion with the cluster graphs containing these chains presented in the next section.

noun phrases $np_3$ and $np_6$ have a very high possibility for belonging to the chain, $np_1$ only a medium possibility, and the remaining NPs are most likely not chain members.

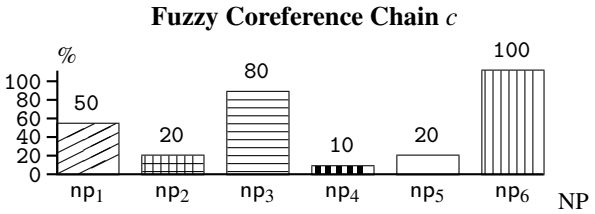


Figure 1: Fuzzy chain $c$ with membership grades for each noun phrase $np_1, \ldots, np_6$

The output of a fuzzy coreference algorithm is a set of fuzzy coreference chains, similarly to classical coreference 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 different degrees of certainty for each. This provides an explicit representation of the uncertainty that is so common in natural language analysis.

Fuzzy chains can be converted to crisp chains using a *defuzzification* function, which allows downstream language analysis components that are not fuzzy-aware to use results of a fuzzy algorithm.

#### 2.2.2 Fuzzy Coreference Resolution

Fuzzy chains are constructed through (usually knowledge-poor) *fuzzy heuristics*. Some features used by our algorithm are *head noun, gender, string similarity,* and the *WordNet distance*.

Our fuzzy coreference algorithm is essentially a single-link hierarchical clustering strategy. It initially creates one fuzzy chain for each NP, which forms its medoid (for example, in Figure 1, $np_6$ is the chain's medoid). We then compute the degree of coreference between all NP pairs within a text, each degree normalized to a fuzzy value in the $[0,1]$-interval. Each fuzzy degree can be interpreted as a distance between the medoid and the co-refering NP; it is added to every chain using standard fuzzy set operators. For example, in Figure 1, at least one fuzzy heuristic must have determined a fuzzy coreference degree between $(np_6, np_3)$ of 0.8.

Finally, all chains are *merged* using a prescribed consistency degree $\gamma$. Merging combines compatible chains into merged chains (or NP clusters) using the coreference properties of symmetry and transitivity. The merge degree $\gamma$ influences the size of the chains, and in effect, their precision and recall. A degree of 0 would merge all NPs into a single (yet useless) chain, while a value of 1 would lead to chains of the best possible precision, leaving out uncertain links and thereby resulting in more singletons (and lower recall). The result is a set of coreference chains $C^\gamma = \{c_1, \ldots, c_o\}$.

The process of merging is now repeated for each possible value of $\gamma \in \{\gamma_1, \ldots, \gamma_n\}$,[3] leading to a *family* of coreference chains, a set of sets of chains: $\mathscr{C} = \{C_1^{\gamma_1}, \ldots, C_n^{\gamma_n}\}$. Note that a similar result can be obtained with a non-fuzzy coreference clustering strategy, however, for the purpose of our algorithm described in the next section it is important that the individual chains exhibit *monotonicity*, that is, if two entities are linked within a chain of a specific certainty $\gamma_i$, they must also be linked in all chains of lower certainty $\gamma_j \leq \gamma_i$.

We use the same algorithm to create both inter- and intra-document coreference chains, only the number of enabled heuristics and various parameters differ for each. The end results are two families of coreference chains, one for intra- and one for inter-document coreferences.[4]

For more details on fuzzy set theory, fuzzy clustering, and fuzzy coreference resolution, we refer the reader to the cited literature.

### 2.3 Fuzzy Coreference Cluster Graphs

We can now describe our fuzzy coreference cluster graph algorithm that builds the data structure needed for constructing context-based summaries. This algorithm takes as input the intra- and inter-document coreference chain families computed by a coreference algorithm under different (fuzzy) clustering thresholds as described in the previous section.

The first step is the construction of an initial fuzzy coreference cluster graph, as described in Section 2.3.1 below. Our clustering algorithm, described in Section 2.3.2, then works on this data structure, computing clusters that can be used to create several kinds of multi-document summaries, including focused summaries (Section 2.4).

#### 2.3.1 Cluster Graph Initialization

A *fuzzy coreference cluster graph* is an undirected, weighted graph with entities (here NPs or VGs) as nodes and weighted coreferences between these entities as edges. Essentially, it folds both inter- and intra-document coreference chains into one data structure that can then be traversed by the clustering algorithm. Thus, the algorithm's input are the intra- and inter-document coreference families described above:

[3] Since fuzzy sets are stored in horizontal representation through a set of $\alpha$-*cuts*, with $[\mu]_\alpha = \{\omega \in \Omega | \mu(\omega) \geq \alpha\}$, the merge degree $\gamma$ can only assume a finite number of different values, which typically correspond to the $\alpha$-cut levels (e.g., $\alpha \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$).

[4] Cross-document chains do not contain links between NPs of the same document, since these links have already been computed by the intra-document step.

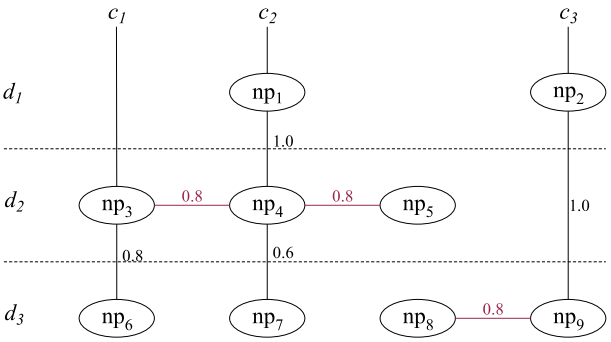


Figure 2: Initialized fuzzy cluster graph

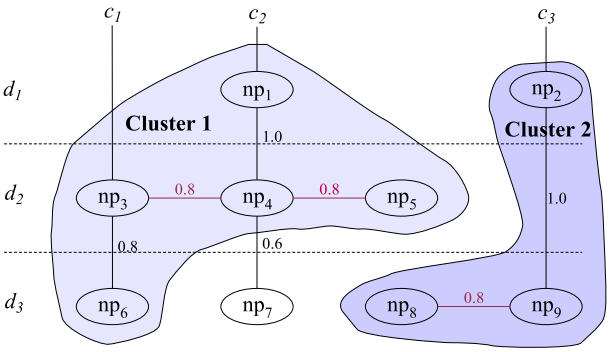


Figure 3: Resulting graph after running the clustering algorithm with $\theta = 0.8$

**Input (cluster graph initialization).** Input to the cluster initialization step is a set of sets of coreference chains $\mathscr{C} = \mathscr{C}_{\text{inter}} \cup \mathscr{C}_{\text{intra}}$, with the inter-document coreference chains $\mathscr{C}_{\text{inter}} = \{C_1^{\gamma_1}, \ldots, C_n^{\gamma_n}\}$ and the intra-document chains $\mathscr{C}_{\text{intra}} = \{C_{n+1}^{\gamma_1}, \ldots, C_{2n}^{\gamma_n}\}$.

Note that each coreference chain $c \in C_i^{\gamma}$ contains again a set of sets of NPs, where all NPs within a subset $c \in C$ corefer with a certainty degree of $\gamma$. We can now create the initial cluster graph:

**Definition (initial cluster graph).** An initial cluster graph $\mathscr{G} = (V, E)$ is constructed from the intra- and inter-document coreference families as follows. The set of graph nodes $V$ is given by the set containing all NPs from all documents. The set of edges is derived from the set $\mathscr{C}$ containing both intra- and inter-document coreference families by iterating through all coreferences $C \in \mathscr{C}$. For each chain $c \in C$, we then iterate through all the entities $(\text{np}_i, \text{np}_j)$ within that chain and create an edge of weight $\gamma$ between them.

Note that we treat coreferences as links, that is, for a coreference chain $c_i^{\gamma} = \{\text{np}_1, \text{np}_2, \text{np}_3\}$ we add two edges with weight $\gamma$ to the graph, one between $\text{np}_1$ and $\text{np}_2$ and one between $\text{np}_2$ and $\text{np}_3$.

**Example (initial cluster graph).** Figure 2 shows an example for an initial cluster graph. There are three documents $d_1, d_2, d_3$ and two coreference families (inter- and intra-document), containing three coreference sets each for $\gamma \in \{0.6, 0.8, 1.0\}$:[5]

$$\mathscr{C}_{\text{inter}} = \{C_1, C_2, C_3\}, \mathscr{C}_{\text{intra}} = \{C_4, C_5, C_6\}$$

With the inter-document chains $C_1, C_2, C_3$:

$$C_1 = \{\{\text{np}_3, \text{np}_6\}, \{\text{np}_1, \text{np}_4, \text{np}_7\}, \{\text{np}_2, \text{np}_9\}\}$$
$$C_2 = \{\{\text{np}_3, \text{np}_6\}, \{\text{np}_1, \text{np}_4\}, \{\text{np}_2, \text{np}_9\}\}$$
$$C_3 = \{\{\text{np}_1, \text{np}_4\}, \{\text{np}_2, \text{np}_9\}\}$$

[5] Singletons are omitted for brevity

and the intra-document chains $C_4, C_5, C_6$:

$$C_4 = \{\{\text{np}_3, \text{np}_4, \text{np}_5\}, \{\text{np}_8, \text{np}_9\}\}$$
$$C_5 = \{\{\text{np}_3, \text{np}_4, \text{np}_5\}, \{\text{np}_8, \text{np}_9\}\}$$
$$C_6 = \{\}$$

Intra-document coreference chains are drawn horizontally (in red), while cross-document chains (in black) are displayed from top to bottom. Each edge in the graph is labeled with the fuzzy *certainty* value of the coreference; in the example, $\text{np}_4$ and $\text{np}_5$ corefer with a certainty of 0.8, while $\text{np}_4$ and $\text{np}_7$ corefer with a certainty of 0.6.

### 2.3.2 The Clustering Algorithm

We can now describe the main clustering algorithm that works on the initial data structure described above. Similarly to chain merging, graph clustering is controlled by a threshold $\theta$. In general, the lower the clustering threshold, the more entities are clustered together, resulting in fewer, but larger, clusters.

The key idea is to use the degree of coreference between entities, represented by an edge's weight, as the *inverse distance* between those entities: entities linked by an edge of weight 1.0 are closest, whereas entities with an edge of weight 0.0 (i.e., no edge) are infinitely far apart. We can now apply an agglomerative hierarchical clustering strategy, creating a dendogram data structure:

**Definition (coreference graph clustering).** The clustering process starts with clusters containing individual entities, i.e., each node $v \in V$ in the initialized graph $\mathscr{G}$ represents a cluster by itself. We now apply a hierarchical clustering strategy, where we progressively merge clusters until the algorithm terminates. Two clusters are merged if a direct edge exists between them of weight $\gamma \geq \theta$. If multiple edges exist between two clusters, we evaluate the one with the highest weight, i.e., we use a single-linkage clustering strategy. The algorithm terminates when no more edges $e \geq \theta$ exist between clusters.

| **"What caused the crash of EgyptAir Flight 990? Include evidence, theories and speculation."** |
|---|
| After an examination of the flight data recorder, the cockpit voice recorder, radar data and small amounts of wreckage, Hall said in that there was no sign of mechanical failure that could have caused the crash. For much of the past week, investigators and mourning families have waited as foul weather and winds caused delays in the search for the recorders. Hall said the retrieval of the first recorder _ another, the cockpit voice recorder, remains missing on the sea floor _ could provide key insights into the crash. An electrical or computer problem would cause those screens to go blank, or could even result in erroneous flight data to be displayed, Gellert said. That would explain why it kept working longer, assuming an electrical problem caused the autopilot to disconnect. Primarily because authorities have introduced the possibility that a human _ not some mechanical failure _ caused the crash. Under that theoretical line of reasoning, if co-pilot Gamil El Batouty really did cause the crash, he could have done so for political rather than personal reasons. Hall acknowledged there have been "many rumors, theories and stories" circulating about whether the crash was caused by mechanical failure or a criminal act such as a hijacking, crew fight or pilot suicide. Investigators doubt that the plane crash, which resulted in the death of all 217 aboard, may have been caused by a criminal act. The Egyptian side has strongly opposed the speculations, stressing technical failure might cause the tragedy. |

Figure 4: ERSS-generated focused summary for D0617H (context shown on top)

**Example (final cluster graph).** Figure 3 shows the result after running the clustering algorithm on the graph in Figure 2 with $\theta = 0.8$. This results in two large NP clusters and the singleton cluster $\{\text{np}_7\}$ in document $d_3$. For $\theta = 0.6$, however, $\text{np}_7$ would have been added to cluster 1, whereas a larger $\theta$ value would have created smaller clusters and more singletons.

Note that we can repeat the clustering process for each fuzzy value of $\theta$, which results in a cluster family (or one multi-dimensional cluster). However, within this paper, we will only discuss the application of single clusters.

### 2.4 Generating Focused Summaries

Focused multi-document summaries are based on a user *context*, which in DUC corresponds to a set of questions. Thus, a focused summary needs to collect information from the documents that pertains to the context (answer the questions), which can result in a summary that is very different from standard, unfocused summaries that typically include the most salient information.

We generate focused summaries from cluster graphs by including the focus questions (context information) as another, distinct document $d_0$ when creating and clustering the fuzzy coreference chains. Then, all clusters that contain entities (NPs or VGs) from document $d_0$ also contain information relevant to the focus question. All other clusters, even if they are bigger, are discarded for this kind of summary, i.e., we slice the cluster graph with the context document. Within our system, this is done twice, for the NP cluster graph and the VG cluster graph.

Sentences containing at least one element in the remaining clusters are ranked according to a number of features:

**Position:** This feature ranks a sentence according to its position within the original newspaper article. A linear function is used to award early occurrence in case the sentence starts before a certain threshold set to 250 characters. Sentences later in the text do not receive any scores.

**Context NPs:** To rank sentences higher that contain several noun phrases from the question, we score a sentence $s$ according to a formula that normalizes the number of entities $e$ within a sentence, depending on the sentence length in words $w$:

$$\text{Score}(s) = \frac{(\alpha^2+1)\cdot(1-\frac{1}{1+e})\cdot\frac{e}{w}}{\alpha^2} + \left(1-\frac{1}{1+e}\right)$$

An alternative scoring strategy is the harmonic mean, which we evaluate together with short, long, and no normalization in Section 3.3.

$$\text{HamonicScore}(s) = \frac{(\alpha^2+1)\cdot(1-\frac{1}{1+e})\cdot\frac{e}{w}}{\alpha^2\cdot(1-\frac{1}{1+e})+\frac{e}{w}}$$

Both use an empirically determined value of $\alpha = 4$.

**Context VGs:** We apply the same strategy we use for NPs to VGs.

**Tf∗idf:** For evaluation purposes, we also introduced a tf∗idf-based feature. This computes a tf∗idf value for each word in a sentence based on the corpus of the DUC 2006 data. The score for each sentence is then computed as the sum of the single tf∗idf scores divided by the number of words in the sentence.

For each feature, all sentences are ranked based on their individual scores. Each feature is then assigned a certain *weight*, which we empirically determined through experiments (see Section 3.3). And the final sentence ranks are subsequently computed by summing up their ranks for each feature with the prescribed weight.

**Sentence Extraction.** We can now extract sentences from the documents based on their rank. The basic idea is to generate the summary by choosing the highest ranked sentences until the word limit has been reached. To prevent exceeding the length limit (250 words in DUC 2006), the summarizer can replace sentences which would cause

| Measure | ERSS | ERSS' | mean | best / worst | rank | rank' |
|---|---|---|---|---|---|---|
| ROUGE-1 | 0.369060 | 0.383591 | 0.371414 | 0.409779 / 0.223513 | 23/35 | 14/35 |
| ROUGE-2 | 0.064839 | 0.076589 | 0.073627 | 0.095097 / 0.028351 | 29/35 | 18/35 |
| ROUGE-SU4 | 0.123896 | 0.134975 | 0.128826 | 0.154662 / 0.063982 | 24/35 | 17/35 |
| Basic Elements | 0.034072 | 0.034830 | 0.36179 | 0.050786 / 0.004565 | 25/35 | 23/35 |
| Linguistic quality | 3.22 | — | 3.38 | 4.39 / 2.32 | 23/35 | — |
| Responsiveness content | 2.52 | — | 2.54 | 3.08 / 1.68 | 21/35 | — |
| Responsiveness overall | 2.28 | — | 2.18 | 2.84 / 1.34 | 11/35 | — |

Table 1: Evaluation results overview for ERSS 2006 (System ID #20) and post-DUC experiment ERSS'

the complete summary to exceed the limit with lower ranked sentences that still fit in the 250 words limit.

An alternative sentence selection strategy allows to include lower ranked sentences in the summary if they contain entities of sentences in the context that the the higher ranked sentences do not address. With this strategy, we can ensure that at least one sentence referring to every part of the question is included, presuming such candidate sentences exist.

**Postprocessing.** After the relevant sentences have been ranked and extracted, we perform a few postprocessing steps:

1. Filter out sentences starting with a quote or a bracket.
2. Remove temporal expressions like "this week," "last Saturday," or "on Monday," as well as some other phrases like "nevertheless" or "however."
3. Sort the sentences according to the position of the entities in the question, i.e., sentences answering the first question come first in the summary and so on, independent of their ordering in the original documents.
4. Another optional formatting feature, which was added to improve the human-measured grammaticality score, is to use one paragraph in the output summary for each sentence of the question.

An example for an ERSS-generated summary can be seen in Figure 4.

## 3 Evaluation

Like in previous years, NIST evaluated all participating systems with several automatic ROUGE measures (Lin and Hovy, 2003; Lin, 2004) and additionally the Basic Elements (BE) measure (Hovy et al., 2005), as well as a number of manual measures, including the (pseudo-) extrinsic *Responsiveness* score. In addition, we performed evaluations on several of our system's parameters using the automatic measures in order to determine their influence on the final score (Section 3.3).

Table 1 gives an overview of the results using the different evaluation measures. Most notable is the striking

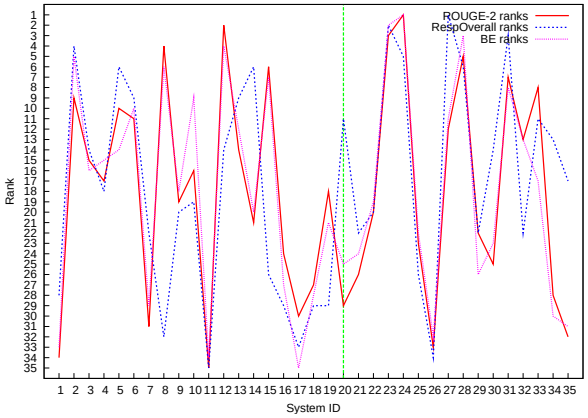


Figure 5: System ranks based on ROUGE, BE and overall responsiveness scores for all systems (not including human summaries)

difference in our system's rank when evaluated with the ROUGE and BE measures and the overall responsiveness. Figure 5 in particular shows the divergence of the results: While most systems score very similar under the different measures, they significantly disagree for ERSS (system no. 20). A similar effect can only be observed for systems no. 8 and 15, which are highly regarded by the automated metrics, but score poorly under the manual responsiveness, and system no. 35, which behaves similarly to our own.

Unlike last year, we did not participate in this year's additional Pyramid (Nenkova and Passonneau, 2004) evaluation. This is due to our analysis of last year's results (Witte et al., 2005), where we could not find any correlation between the Pyramid score and the other automatic or the manual responsiveness measure when regarding our system by itself, i.e., not the averaged correlation over all systems.

### 3.1 Automatic Evaluation

In addition to the automatic evaluation performed by NIST, we examined the correlation between the BE score and several ROUGE scores for our system only, i.e., not the correlation over all systems. This has been computed using the Spearman rank coefficient, as we previously described (Witte et al., 2005); and like last year, their

| Measure | average correlation with | | | | | |
|---|---|---|---|---|---|---|
| | BE | ROUGE-1 | ROUGE-2 | ROUGE-SU4 | Resp. Content | Resp. Overall |
| Basic Elements | — | 0.920768 | 0.958511 | 0.941801 | 0.791405 | 0.604850 |
| ROUGE-1 | | — | 0.937191 | 0.972629 | 0.846146 | 0.614934 |
| ROUGE-2 | | | — | 0.978391 | 0.764322 | 0.556735 |
| ROUGE-SU4 | | | | — | 0.781897 | 0.578343 |
| Responsiveness Content | | | | | — | 0.708283 |
| Responsiveness Overall | | | | | | — |

Table 2: Spearman correlations between the BE, ROUGE-2, ROUGE-SU4, responsiveness content, and responsiveness overall scores, for all systems

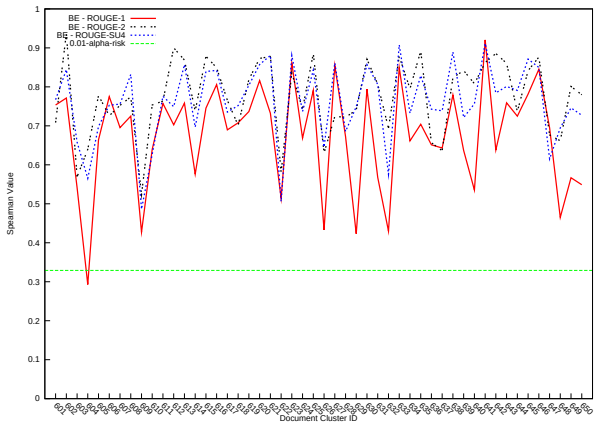


Figure 6: Correlation of different ROUGE and BE system ranks per document cluster over all systems

correlation is copacetic: 0.852 for BE/ROUGE-1, 0.894 for BE/ROUGE-2, and 0.895 for BE/ROUGE-SU4.

Figure 6 shows the correlation for each document set, over all systems. With the exception of ROUGE-1, the agreement of the other two ROUGE measures with BE is for every cluster well above the 0.01 $\alpha$-risk.

The correlations between the automatic (ROUGE, BE) and manual (responsiveness, responsiveness overall) measures are shown in Table 2. The most important result here is that there is no significant correlation between the pseudo-extrinsic *overall responsiveness* score and any automatic metric.

### 3.2 Manual Evaluation

Table 3 shows the manual evaluation results for ERSS. Ranks for ERSS are overall lower compared to last year (Witte et al., 2005), with the exception of *Grammatical quality*, where we implemented most of our improvements for this year's version. This appears to be the main reason for the drastically higher *overall responsiveness* score, which also measures the readability of the summary, not just information content (as this year's and last year's *responsiveness* measures do).

| Linguistic Feature | ERSS | mean | best / worst | rank |
|---|---|---|---|---|
| Grammaticality | 3.96 | 3.58 | 4.52 / 1.38 | 9/35 |
| Non-redundancy | 4.08 | 4.23 | 4.66 / 3.76 | 24/35 |
| Referential clarity | 2.46 | 3.12 | 4.70 / 1.90 | 32/35 |
| Focus | 3.32 | 3.60 | 4.56 / 2.50 | 31/35 |
| Structure/Coherence | 2.28 | 2.39 | 4.22 / 1.16 | 19/35 |

Table 3: Manual evaluation results for ERSS 2006

### 3.3 Post-DUC experiments

In order to determine how our ranking features (see Section 2.4) influence the resulting scores, we performed additional experiments with various parameter settings. Some of these are shown in Table 4. The tf∗idf-based rank was introduced to serve as a baseline in order to determine how much the fuzzy clustering algorithm contributes to ERSS' performance.

The first row shows the settings as we ran them for DUC 2006; the second row gives the results from the ERSS 2005 system running on the 2006 data. Turning off the sentence filtering (postprocessing) decreased performance slightly, as does the alternative sentence selection strategy aiming to include an answer for each question ("AllQs" column). When looking at individual weights, we can see that NPs contribute most to the summaries, which is to be expected. Looking only at verbs gives the worst performance, even lower than ranking based on tf∗idf. Thus, our newly added verb clustering strategy hurt the performance—whether this holds in general or is due to our very crude verb coreference strategy is still under investigation.

In fact, the best result was obtained using only noun phrase clusters (Table 4, last row), without using any verb clusters, early occurrence, or tf∗idf-boost.

We also experimented with different normalization strategies for the number of NPs or VGs within a sentence. The achieved results can be seen in Table 5. The formula we used can be found in Section 2.4.

## 4 Discussion and Conclusions

While ERSS ranks lower this year in the DUC competition, our analysis shows that the system itself improved slightly when compared with last year's version running on the

| filter | early | np | vg | tfidf | AllQs | $\theta$ | ROUGE2 | BE |
|---|---|---|---|---|---|---|---|---|
| ERSS 2006 vs. 2005 | | | | | | | | |
| true | 3 | 3 | 3 | 0 | false | 0.6 | 0.065 | 0.034 |
| false | 0 | 3 | 0 | 0 | false | 0.6 | 0.063 | 0.024 |
| Clustering Threshold $\theta$ | | | | | | | | |
| true | 3 | 3 | 3 | 0 | false | 0.6 | 0.065 | 0.034 |
| true | 3 | 3 | 3 | 0 | false | 0.8 | 0.065 | 0.029 |
| true | 0 | 3 | 0 | 0 | true | 0.6 | 0.066 | 0.026 |
| true | 0 | 3 | 0 | 0 | true | 0.8 | 0.077 | 0.035 |
| true | 3 | 3 | 0 | 0 | true | 0.6 | 0.071 | 0.035 |
| true | 3 | 3 | 0 | 0 | true | 0.8 | 0.074 | 0.033 |
| Filter Parameter | | | | | | | | |
| true | 3 | 3 | 3 | 0 | false | 0.6 | 0.065 | 0.034 |
| false | 3 | 3 | 3 | 0 | false | 0.6 | 0.063 | 0.029 |
| Include all Questions Strategy | | | | | | | | |
| true | 3 | 3 | 3 | 0 | false | 0.6 | 0.065 | 0.034 |
| true | 3 | 3 | 3 | 0 | true | 0.6 | 0.064 | 0.028 |
| Singleton Scoring Weights | | | | | | | | |
| true | 3 | 0 | 0 | 0 | false | 0.6 | 0.065 | 0.026 |
| true | 0 | 3 | 0 | 0 | false | 0.6 | 0.070 | 0.034 |
| true | 0 | 0 | 3 | 0 | false | 0.6 | 0.050 | 0.018 |
| true | 0 | 0 | 0 | 3 | false | 0.6 | 0.057 | 0.021 |
| Best Result | | | | | | | | |
| true | 0 | 3 | 0 | 0 | true | 0.8 | 0.077 | 0.035 |

Table 4: ROUGE-2 and BE scores for different parameter configurations, first row shows DUC 2006 settings

same data (both 2005 and 2006).

Since ERSS was largely unchanged, we conclude that other systems have improved markedly since DUC 2005: Especially stochastic systems gain from the availability of a larger training set, since the task was almost unchanged from last year. This underscores the importance of a mostly rule-based system like ERSS to explore, develop, and test features for new domains and tasks, where statistical systems suffer from lack of training data."

**Acknowledgments.** We would like to thank Thomas Moschny from Universität Karlsruhe, IPD, for his support in running ERSS on the institute's Itanium cluster.

| Formulas | ERSS | | ERSS' | |
|---|---|---|---|---|
| | ROUGE-2 | BE | ROUGE-2 | BE |
| No normalization | 0.062 | 0.025 | 0.077 | 0.035 |
| #Entities/#Words | 0.060 | 0.027 | 0.070 | 0.031 |
| #Words/#Entities | 0.062 | 0.024 | 0.062 | 0.025 |
| ERSS formula | 0.065 | 0.034 | 0.077 | 0.035 |
| 1/ERSS formula | 0.063 | 0.024 | 0.062 | 0.025 |
| Harmonic | 0.064 | 0.026 | 0.068 | 0.029 |
| 1/Harmonic | 0.065 | 0.026 | 0.068 | 0.029 |

Table 5: Effects of different sentence normalization strategies on the results

## References

Sabine Bergler, René Witte, Michelle Khalife, Zhuoyan Li, and Frank Rudzicz. 2003. Using Knowledge-poor Coreference Resolution for Text Summarization. In *Proceedings of the HLT/NAACL Workshop on Text Summarization (DUC 2003)*. Document Understanding Conference. http://www-nlpir.nist.gov/projects/duc/pubs/2003final.papers/concordia.final.pdf.

Sabine Bergler, René Witte, Zhuoyan Li, Michelle Khalife, Yunyu Chen, Monia Doandes, and Alina Andreevskaia. 2004. Multi-ERSS and ERSS 2004. In *Proceedings of the HLT/NAACL Workshop on Text Summarization (DUC 2004)*. Document Understanding Conference. http://www-nlpir.nist.gov/projects/duc/pubs/2004papers/concordia.witte.pdf.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*. http://gate.ac.uk.

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.

E. Hovy, C. Lin, and L. Zhou. 2005. Evaluating DUC 2005 using Basic Elements. In NIST (NIST, 2005). http://duc.nist.gov/pubs.html#2005.

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

Chin-Yew Lin and E. H. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proc. of the 2003 Human Language Technology Conference HLT/NAACL 2003*, Edmonton, Canada, May 27–June 1.

Chin-Yew Lin. 2004. ROUGE: a Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, Barcelona, Spain, July 25–26. http://www.isi.edu/~cyl/ROUGE/.

Ani Nenkova and Rebecca J. Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. In *Proc. HLT/NAACL*, pages 145–152.

NIST, editor. 2005. *DUC 2005*, Vancouver, BC, Canada, October 9–10. http://duc.nist.gov/pubs.html#2005.

NIST, editor. 2006. *DUC 2006*, New York City, NY, USA, June 8–9. http://duc.nist.gov.

René Witte and Sabine Bergler. 2003. Fuzzy Coreference Resolution for Summarization. In *Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, pages 43–50, Venice, Italy, June 23–24. Università Ca' Foscari. http://www.rene-witte.net/downloads/wittebergler_fuzzycoref.pdf.

René Witte, Ralf Krestel, and Sabine Bergler. 2005. ERSS 2005: Coreference-Based Summarization Reloaded. In *Proceedings of Document Understanding Workshop (DUC)*, Vancouver, B.C., Canada, October 9–10. http://duc.nist.gov/pubs/2005papers/ukarlsruhe.witte.pdf.

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

René Witte. 2004. An Integration Architecture for User-Centric Document Creation, Retrieval, and Analysis. In *Proceedings of the VLDB Workshop on Information Integration on the Web (IIWeb'04)*, pages 141–144, Toronto, Canada, August 30. http://rene-witte.net/downloads/witte_iiweb04.pdf.

René Witte. 2006. Multi-lingual Noun Phrase Extractor (MuNPEx). http://www.ipd.uka.de/~durm/tm/munpex.