Sentence Trimming and Selection: Mixing and Matching

David M. Zajic
University of Maryland
dmzajic@umiacs.umd.edu
John M. Conroy
IDA/Center for Computing Sciences
conroy@super.org
Bonnie J. Dorr
University of Maryland
bonnie@cs.umd.edu
Dianne P. O'Leary
University of Maryland
oleary@cs.umd.edu
Jimmy Lin
University of Maryland
jimmylin@umiacs.umd.edu
Judith D. Schlesinger
IDA/Center for Computing Sciences
judith@super.org

Abstract

We describe how components from two distinct multi-document summarization systems were combined. Twenty four possible combinations of components were considered. We observed some contrasts between conservative and aggressive sentence compression (i.e., *trimming*) in the context of multi-document summarization.

1 Introduction

The University of Maryland and IDA/CCS collaborated on a submission to DUC2006. Both sites' systems included tools for sentence compression (i.e., *trimming*), sentence selection (i.e., scoring) and summary generation. Please see our individual discussions (Zajic et al., 2006; Conroy et al., 2006b) for details.

We merged our systems in a variety of ways. Some tools could be used in combination; others were treated as alternatives. For example, the sentence compression tools could be used separately, in combination, or omitted entirely. Likewise, sentence scoring could be done by a combination of tools or by each site separately. However, only one or the other of the summary generation tools was used in any given configuration.

For DUC2006, we submitted the configuration that had the highest ROUGE-2 score on the DUC2005 test data.

Earlier work has shown that sentence compression improves performance of *single*-document summarization systems. (Zajic et al., 2004) (Dorr et al., 2003) In this paper, we examine the effect of two different sentence compression approaches on *multi*-document summarization. The sentence compression approaches differ in their level of risk-taking. One is conservative with respect to grammaticality; the other over-generates possible compressions and takes more risks with respect to content and structure.

We also experimented with two different sentence selection schemes.

2 Component and Collaboration Descriptions

In this section we describe, at a high level, the components of our summarization systems, and the ways in which they were combined to create collaborative systems.

2.1 Sentence Compression

The CCS and UMD sentence compression systems both use syntactic trimming, but differ in the depth of parsing information used and the level of risk assumed. The CCS trimmer is conservative, using shallow parsing. The UMD trimmer is aggressive and uses full parses.

The CCS trimmer aims to remove parts of sentences that are less likely to contain information that would be important to have in a summary *without* having an impact on the grammaticality of the sentence. This is achieved by matching each sentence to established patterns that key off of specific words and/or punctuation to locate phrases or clauses that can be deleted.

Removals include:

- lead adverbs, conjunctions, and semantically light multi-word phrases (such as "As an example," or "At one point.";
- medial adverbs, such as "also," "however";
- age references, as in ", 51" or ", aged 24";
- gerund phrases;
- relative clause appositives; and
- attributions, such as "police said."

See (Conroy et al., 2006b) and prior DUC papers for more detail on the workings of the CCS trimmer.

The UMD trimmer uses a linguistically motivated heuristic to trim constituents from a syntactic parse tree until a length constraint is met. The trimming rules are designed to preserve grammaticality after each operation while the heuristic is designed to remove semantically light constituents before removing semantically vital constituents. In the context of multi-document summarization, the global length of the summary is constrained, but not the length of any given sentence. We adapted the trimmer to this application by proposing each intermediate stage of the trimming as a trimmed candidate of the original sentence. Relevance scores (described in Section 2.2) are used to determine which trimmed candidates (including the original sentence) provide the best balance of semantic coverage and brevity. For more details on the UMD Trimmer, see (Zajic et al., 2006) and earlier DUC workshop papers.

2.2 Candidate Scoring

The CCS method of scoring candidates uses an approximate oracle score. This score uses query terms, extracted from the topic descriptions, as well as signature terms (Lin and Hovy, 2002), to approximate the probability that a term will be chosen by a human abstractor. Psuedo-relevance feedback is used to improve the probability distribution. The score, for a sentence x, is denoted by $\omega_{qs\rho}(x)$ and an estimate of the fraction of abstract terms in a sentence. See (Conroy et al., 2006a) and (Conroy et al., 2006b) for details.

The UMD system uses Universal Retrieval Architecture (URA) to calculate relevance and centrality scores for each trimmed candidate. The relevance score is broken down into two separate components: the matching score between a trimmed candidate and the query, and a similarity score between the document containing the trimmed candidate in question and the entire cluster of relevant documents. We assume that candidates having higher term overlap with the query and candidates originating from documents that are more "central" to the topic cluster are preferred for inclusion in the final summary.

2.3 Summary Generation

The CCS system forms the summary by taking the top scoring candidates among those candidates with at least 8 distinct terms. (The length of 8 was empirically determined to be optimal using the DUC05 data.) To minimize redundancy, enough sentences to give a summary of length 500 are first selected, i.e., twice the target length. A pivoted-QR is then used to select the subset of these top scoring sentences (Conroy and O'Leary, March 2001).

The lead sentence of the summary is the highest scoring candidate as given by the score $\omega_{qs\rho}$. The order for the subsequent sentences is determined using a Traveling Salesperson (TSP) formulation (Conroy et al., 2006b) that is seeded with the identified lead sentence and the set of subsequent sentences selected by the pivoted QR. A *distance* between each pair of sentences was defined and an ordering was then determined that minimized the sum of the distances between adjacent sentences. Since the summaries were limited to 250 words, it is not too difficult to solve the TSP. For example, there are only 3,628,800 ways to order 10 sentences plus a lead sentence, so exhaustive search is feasible. Rather than do an exhaustive search, however, the best of large sample of orderings, some random and some determined by single-swap changes on a previous candidate ordering was chosen.

The UMD summary generator implements a Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998) (Goldstein et al., 2000) algorithm. All the candidates are ranked by features

which reward relevance and penalize redundancy. The highest scoring sentence is added to the summary. All the remaining candidates are re-scored for dynamic features, such as redundancy to the current summary state. These steps are iterated until a summary of the desired length is reached.

2.4 System Combinations

We used a common pre-processing base developed by CCS from which to proceed with various combinations of our system components. This base consisted of sentence splitting, "junk removal", and the initial sentence selection. The "junk removal" deals with removal of datelines, editor's instructions, and other strings of characters that should not appear in a summary and are not true parts of sentences. The initial sentence selection used omega to select twice as much material as would fit in the allowed space, in this case 500 words. The collaborative test systems all used this pre-processed base as the source from which a 250-word summary was generated.

It is important to note that all tests began with the small set of sentences selected by the omega score in order to minimize the time required for the UMD trimmer, which uses a full parse, to process.

Our collaboration enabled us to experiment with a variety of options for sentence trimming and sentence selection:

- We have two sentence compression systems, CCS trimming and UMD trimming, referred to as (no_)ccs_trim and (no_)umd_trim, respectively, in the tables below. We can use them together, independently, or we can omit trimming, giving four options. When both compression systems are used CCS trimming is applied first, then UMD trimming generates multiple trimmed candidates. [1]

- We have two summary generation systems, CCS generation and UMD generation. These systems which operate independently, giving two options, referred to as ccs_gen and umd_gen, respectively, in the tables below.

- The summary generation systems make use of features to select high-relevance, low-redundancy sentences. Each system can use URA, omega or a combination of URA and omega, giving three options, referred to as ura, omega, and ura_omega, respectively, in the tables below. When omega is used on UMD trimmed candidates, omega is recalculated for each candidate.

For the CCS sentence selection, the trimming method determined how the URA and omega scores were used. If the UMD trimmer was used the "best candidate trimming" of a sentence was selected by using a weighted combination of the URA and omega scores. For combined systems that used only omega, the weight of URA was set to zero. For combined systems that used only URA, the weight of omega was set to zero. If the omega weight was non-zero, the columns of the term sentence matrix were normalized to the omega score. If the omega weight was zero, the URA score was used to weight the columns.

If the UMD trimmer was not used then there is no need to select a "best candidate trimming," and all sentences were sent to the pivoted QR. The weighting of columns of the term sentence matrix in the QR is handled in the same manner.

The UMD summary generator combined the URA and omega scores as part of the linear combination of features that are used to rescore the candidates at each iteration. When URA features and omega were used together, each of the four URA features was given weight 1.0 and the omega score was given weight 4.0. When only omega was used, the URA features were given weights 0; and when only URA was used omega was given weight 0.

This gives a total of 24 combination systems that were tested on the DUC-05 data, computing average ROUGE-1 and ROUGE-2 scores for each combination scored against the human abstracts. The results are described next.

[1] CCS trimming is applied first for two pragmatic reasons. First, it generates only one trimmed candidate which can serve as a candidate or as the input to UMD trimming. Second, CCS trimming is generally subsumed by UMD trimming, so applying CCS trimming after UMD trimming would be difficult to detect.

CCS Trimmer	Y	Y	Y	Y	Y	Y
UMD Trimmer	Y	Y	Y	Y	Y	Y
Sentence Sel	UMD	UMD	UMD	CCS	CCS	CCS
Omega	Y	Y	N	Y	Y	N
URA	Y	N	Y	Y	N	Y
Rouge 1	0.3776	0.3762	0.3819	0.3837	0.3847	0.3846
Rouge 2	0.0770	0.0772	0.0785	0.0776	0.0792	0.0775

Table 1: Using Both UMD and CCS Sentence Compression

CCS Trimmer	Y	Y	Y	Y	Y	Y
UMD Trimmer	N	N	N	N	N	N
Sentence Sel	UMD	UMD	UMD	CCS	CCS	CCS
Omega	Y	Y	N	Y	Y	N
URA	Y	N	Y	Y	N	Y
Rouge 1	0.3870	0.3877	0.3882	0.3879	0.3881	0.3864
Rouge 2	0.0788	0.0796	0.0793	0.0784	0.0790	0.0780

Table 2: Using Only CCS Sentence Compression

CCS Trimmer	N	N	N	N	N	N
UMD Trimmer	Y	Y	Y	Y	Y	Y
Sentence Sel	UMD	UMD	UMD	CCS	CCS	CCS
Omega	Y	Y	N	Y	Y	N
URA	Y	N	Y	Y	N	Y
Rouge 1	0.3773	0.3751	0.3817	0.3847	0.3822	0.3863
Rouge 2	0.0765	0.0769	0.0781	0.0769	0.0776	0.0776

Table 3: Using Only UMD Sentence Compression

CCS Trimmer	N	N	N	N	N	N
UMD Trimmer	N	N	N	N	N	N
Sentence Sel	UMD	UMD	UMD	CCS	CCS	CCS
Omega	Y	Y	N	Y	Y	N
URA	Y	N	Y	Y	N	Y
Rouge 1	0.3860	0.3881	0.3886	0.3894	0.3872	0.3882
Rouge 2	0.0778	0.0773	0.0783	0.0785	0.0794	0.0781

Table 4: Using No Sentence Compression

1	umd_gen/no_ccs_trim/no_umd_trim/ura	0.38865
2	umd_gen/ccs_trim/no_umd_trim/ura	0.38818
3	ccs_gen/ccs_trim/no_umd_trim/omega	0.38813
4	ccs_gen/ccs_trim/no_umd_trim/ura_omega	0.38813
5	ccs_gen/ccs_trim/no_umd_trim/ura	0.38813
6	umd_gen/no_ccs_trim/no_umd_trim/omega	0.38812
7	umd_gen/ccs_trim/no_umd_trim/omega	0.38775
8	ccs_gen/no_ccs_trim/no_umd_trim/ura	0.38720
9	ccs_gen/no_ccs_trim/no_umd_trim/omega	0.38720
10	ccs_gen/no_ccs_trim/no_umd_trim/ura_omega	0.38720
11	umd_gen/ccs_trim/no_umd_trim/ura_omega	0.38697
12	ccs_gen/no_ccs_trim/umd_trim/ura	0.38627
13	umd_gen/no_ccs_trim/no_umd_trim/ura_omega	0.38601
14	ccs_gen/ccs_trim/umd_trim/ura_omega	0.38570
15	ccs_gen/no_ccs_trim/umd_trim/ura_omega	0.38563
16	ccs_gen/ccs_trim/umd_trim/omega	0.38474
17	ccs_gen/ccs_trim/umd_trim/ura	0.38465
18	ccs_gen/no_ccs_trim/umd_trim/omega	0.38225
19	umd_gen/ccs_trim/umd_trim/ura	0.38193
20	umd_gen/no_ccs_trim/umd_trim/ura	0.38167
21	umd_gen/ccs_trim/umd_trim/ura_omega	0.37759
22	umd_gen/no_ccs_trim/umd_trim/ura_omega	0.37729
23	umd_gen/ccs_trim/umd_trim/omega	0.37619
24	umd_gen/no_ccs_trim/umd_trim/omega	0.37507

Table 5: Rouge 1 Average Recall ranking. The /-separated fields indicate which summary generation system was used; whether ccs trimming was used; whether umd trimming was used; and which combination of URA and omega was used.

1	ccs_gen/ccs_trim/umd_trim/ura_omega	0.07988
2	umd_gen/ccs_trim/no_umd_trim/omega	0.07964
3	ccs_gen/no_ccs_trim/no_umd_trim/ura	0.07935
4	ccs_gen/no_ccs_trim/no_umd_trim/omega	0.07935
5	ccs_gen/no_ccs_trim/no_umd_trim/ura_omega	0.07935
6	umd_gen/ccs_trim/no_umd_trim/ura	0.07930
7	ccs_gen/ccs_trim/umd_trim/omega	0.07924
8	ccs_gen/no_ccs_trim/umd_trim/ura_omega	0.07913
9	ccs_gen/ccs_trim/no_umd_trim/omega	0.07897
10	ccs_gen/ccs_trim/no_umd_trim/ura_omega	0.07897
11	ccs_gen/ccs_trim/no_umd_trim/ura	0.07897
12	umd_gen/ccs_trim/no_umd_trim/ura_omega	0.07880
13	umd_gen/ccs_trim/umd_trim/ura	0.07847
14	umd_gen/no_ccs_trim/no_umd_trim/ura	0.07830
15	umd_gen/no_ccs_trim/umd_trim/ura	0.07808
16	umd_gen/no_ccs_trim/no_umd_trim/ura_omega	0.07783
17	ccs_gen/no_ccs_trim/umd_trim/omega	0.07763
18	ccs_gen/no_ccs_trim/umd_trim/ura	0.07757
19	ccs_gen/ccs_trim/umd_trim/ura	0.07747
20	umd_gen/no_ccs_trim/no_umd_trim/omega	0.07730
21	umd_gen/ccs_trim/umd_trim/omega	0.07722
22	umd_gen/ccs_trim/umd_trim/ura_omega	0.07705
23	umd_gen/no_ccs_trim/umd_trim/omega	0.07685
24	umd_gen/no_ccs_trim/umd_trim/ura_omega	0.07649

Table 6: Rouge 2 Average Recall ranking

Rouge 2	System 8	0.08954 (0.08540 - 0.09338) 4th of 35
Rouge 2	System 15	0.09097 (0.08671 - 0.09478) 2nd of 35
Rouge 2	System 32	0.08051 (0.07679 - 0.08411) 13th of 35
Rouge SU4	System 8	0.14607 (0.14252 - 0.14943) 4th of 35
Rouge SU4	System 15	0.14733 (0.14373 - 0.15069) 3rd of 35
Rouge SU4	System 32	0.13600 (0.13212 - 0.13955) 13th of 35

Table 7: Official DUC2006 ROUGE Scores, 95% Confidence Intervals and Ranks for Systems 8, 15 and 32

3 Results

The ROUGE results for each combination are given in Tables 1 through 4. The ranks of the combination systems on DUC2005 test data are shown in Tables 5 and 6. The 95% confidence interval of the highest scoring combination systems are (0.38289 - 0.39422) for Rouge 1 Recall and (0.07683 - 0.08298) for Rouge 2 Recall. Comparing the remaining 23 combination systems to these 95% confidence intervals shows that there are some significant differences among the combination systems. We selected the system with the highest Rouge 2 Recall score for submission to DUC2006. This system used both CCS and UMD trimming, and the CCS summary generator with both URA and omega to select amongst the UMD trimmer sentence variations and omega only to make the final sentence selections.

The CCS submission described in (Conroy et al., 2006b) used the CCS summary generator with omega only, and used only CCS trimming. The UMD/BBN submission described in (Zajic et al., 2006) is not among these combination systems because it did not make use of the common preprocessing base. It used URA and UMD trimming only, however.

4 Evaluation and Analysis

In the DUC2006 evaluation, the UMD/CCS combination system was System 8. The CCS submission was System 15 and the UMD/BBN submission was System 32. Table 7 shows the scores and ranks of the three systems.

Due to its conservative approach, the CCS trimmer is not introducing any grammatical errors other than those due to code bugs that have since been corrected (or will be if not yet identified). The CCS trimmer permits the inclusion of at least 2–3 additional sentences in a summary. It's important to note, however, that the change from an HMM, which was used until this year, to the omega score, impacted the number of sentences in a summary at least as much as trimming since omega tends to select shorter sentences than the HMM.

In combination with the UMD summary generator, use of the UMD trimmer adds on average 2.73 sentences to a summary. This is a net gain, i.e. on average UMD trimmer introduces 3.13 new sentences to a summary but drops 0.40 existing summaries from the untrimmed summary. This average is not affected by the use of the CMU trimmer, however it is affected by the features used in sentence selection. When only URA is used, 1.72 sentences are added by UMD trimming, but when only omega is used, 3.95 sentences are added. One might guess that this effect is largely due to omega's bias for shorter sentences, with or without UMD trimming. However, this appears not to be the case. The average summary generated without UMD trimming using only URA contained 11.2 sentences, while the average summary generated without UMD trimming using only omega contained 12.0 sentences. The difference appears to be in how many original source sentences are replaced by a trimmed candidate. With UMD trimming and URA, 51.3% of sentences are replaced by a trimmed version of that sentence. Under omega, 65.4% are replaced by a trimmed version. When both URA and omega are used, the figures fall in between: 54.8% of sentences are replaced by a trimmed candidate, resulting in an average net increase of 2.51 sentences.

5 Conclusion

We have combined components of two distinct multi-document summarization systems, both of which make use of sentence compression. We explored 24 possible ways of combining these components, and selected the combination with the highest Rouge 2 Recall score on the DUC2005

test data. We observed that conservative trimming does not lead to the loss of important information, but that aggressive trimming can sometimes cause the loss of important information. However, aggressive trimming is capable of freeing up space to include more sentences in the summary which can often improve the summary content.

Acknowledgments

This work has been supported, in part, under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-2-0001, the TIDES program of the Defense Advanced Research Projects Agency, BBNT Contract No. 9500006806, and the University of Maryland Joint Institute for Knowledge Discovery. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA.

References

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Research and Development in Information Retrieval*, pages 335–336.

J.M. Conroy and D.P. O'Leary. March, 2001. "Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition". Technical report, University of Maryland, College Park, Maryland.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. 2006a. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the ACL'06/COLING'06*.

John M. Conroy, Judith D. Schlesinger, Dianne P. O'Leary, and Jade Goldstein. 2006b. Back to basics: Classy 2006. In *Proceedings of the 2006 Document Understanding Workshop, New York*.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 2003 Text Summarization Workshop, Edmonton, Alberta, Canada*, pages 1–8.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, , and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of ANLP/NAACL Workshop on Automatic Summarization*, pages 40–48.

C.Y. Lin and E. Hovy. 2002. The automatic acquisition of topic signatures for text summarization. In *DUC 02 Conference Proceedings*.

David Zajic, Bonnie Dorr, and Richard Schwartz. 2004. BBN/UMD at DUC2004: Topiary. In *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop, Boston*, pages 112–119.

David Zajic, Bonnie Dorr, Richard Schwartz, and Jimmy Lin. 2006. Sentence Compression as a Component of a Multi-Document Summarization System. In *Proceedings of the 2006 Document Understanding Workshop, New York*.