Catch What You Can

Terry COPECK, Diana INKPEN, Anna KAZANTSEVA,
Alistair KENNEDY, Darren KIPP, Stan SZPAKOWICZ
School of Information Technology and Engineering
University of Ottawa
800 King Edward Avenue
Ottawa, Ontario, Canada K1N 6N5
{terry,diana,ankazant,akennedy,dkipp,szpak}@site.uottawa.ca

Abstract

Work on text summarization for DUC 2007 at the University of Ottawa continued along the lines established in recent years. We participated in the associated Pyramid evaluation effort now entrusted to the broader research community, expanding our corpus of SCU-marked documents by the 23 topics annotated this year to a total of 70. For the first time our summarization system made direct use of the information in this corpus as part of its mechanism to select sentences, and we were gratified to see an improvement in the scores that human judges gave to the summaries it produced.

1 Introduction

Participation in the Document Understanding Conference each year helps focus and stimulate the summarization research of our team. DUC's schedule establishes a certain tempo for system development, and the annual workshop provides an occasion to compare notes with others and to learn about alternative ways of dealing with common issues in summarization. Most important to us however is what occurs between delivery of our submission and the workshop post-mortem—evaluation of participants' summaries by the staff at NIST. We inspect these results with interest: they help guide the direction of future work.

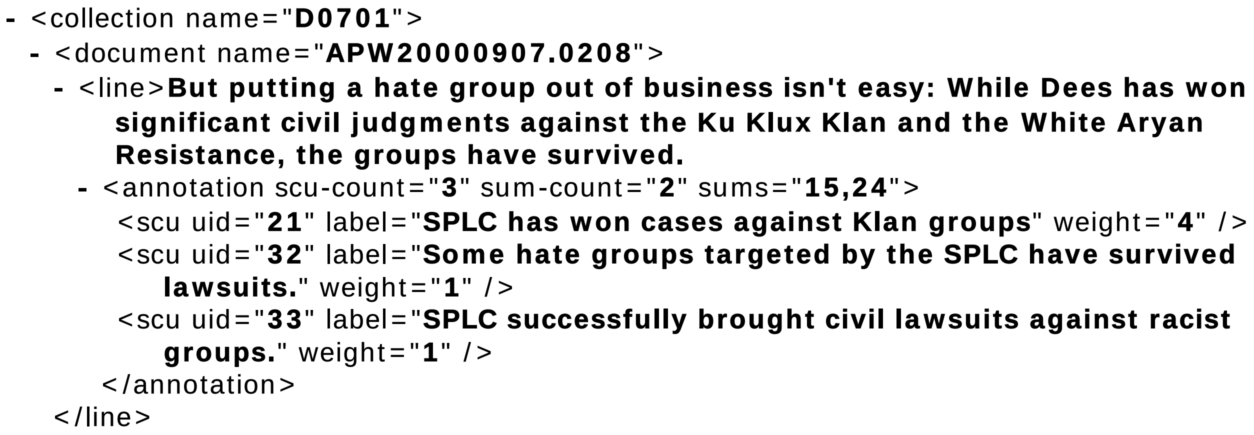
If we draw benefits from DUC, we do in return try to give back to the summarization community. Beyond participating in most of the optional tracks and tasks offered in each year's conference, the University of Ottawa uses the information in the Pyramid .pan files to annotate original document sentences with their Summary Content Unit (SCU) information (Nenkova & Passonneau 2004). We have developed and maintain a corpus of topics marked with these data and make it available to DUC participants on request.

This SCU-marked corpus may be of most interest to readers, and recent developments concerning it are reported first in the next section. We then proceed in subsequent sections to describe the design of our summarization system used in DUC 2007 and discuss its performance on the conference test data.

2 Work on Pyramid Data

Each year we update a corpus of topic document collections with new topics used in the year's Pyramid evaluation activity. The corpus is composed of one XML file (`.scu`) per topic in which sentences identified by our locally written sentence boundary detector are concatenated on a document-by-document basis. Because most summarization systems compose their output using sentences extracted from original documents, any assessment of these sentences in these summaries in terms of SCUs can, in the majority of cases, successfully be propagated back to the source document (Copeck & Szpakowicz 2005). This provides the useful resource of a document marked with a measure of the degree to which some number of its individual sentences address the information request on which the Pyramid was based.

One caveat: there is no generally-accepted standard for recognizing sentence boundaries, and any such annotation is particular to the sentences recognized; on this issue people can disagree. A further flaw occurs when a system incorrectly recognizes, or fails to recognize, a break that incontestably does not or does end a sentence. Our sentence break detector is certainly not perfect. Notwithstanding these limitations, coming



```
- <collection name="D0701">
  - <document name="APW20000907.0208">
    - <line>But putting a hate group out of business isn't easy: While Dees has won
        significant civil judgments against the Ku Klux Klan and the White Aryan
        Resistance, the groups have survived.
      - <annotation scu-count="3" sum-count="2" sums="15,24">
          <scu uid="21" label="SPLC has won cases against Klan groups" weight="4" />
          <scu uid="32" label="Some hate groups targeted by the SPLC have survived
              lawsuits." weight="1" />
          <scu uid="33" label="SPLC successfully brought civil lawsuits against racist
              groups." weight="1" />
        </annotation>
    </line>
```

Figure 1: Annotation of a Sentence with Multiple SCUs

to agreement on the set of sentences that constitute a document has never appeared to cause a problem for participants in DUC.

Sentences in the topic XML files which compose the corpus are stored as `<line>` elements under the document in which they appear. Those which realize SCUs are marked with an `<annotation>` element composed of three attributes with simple values and one or more `<scu>` elements, each of which describes a SCU realized by the sentence. The fields appearing within the annotation structure are as follows:

`scu-count:`	count of SCUs realized by the sentence – *integer; agrees with count of SCU elements*
`sum-count:`	count of summaries using the sentence – *integer*
`sums:`	anonymized identifiers of participants – *comma-delimited list; agrees with sum-count*
`uid:`	SCU identifier – *integer*
`label:`	content of SCU – *string*
`weight:`	number of manually written summary passages the SCU expresses – *integer*

Figure 1 provides an example taken from topic D0701. A sentence in document APW2000907.0208, *But putting a hate group ...* , has been deemed by annotators to realize the three SCUs #21, #32 and #33. Two of these content units each reflect a single passage in the manually-written summaries on which the D0701 Pyramid was based, while the substance of the third, *SPLC has won cases against Klan groups*, appears four times in manual summaries. Two participants in the conference, peers #15 and #24, used this sentence in their summary. It is because of their use that we are able to associate the three SCUs listed in the annotation with this source document sentence.

3.1 SCU Results

Two questions may suggest themselves in connection with the effort to annotate source documents with SCU data. How useful are the Pyramid data? And if it is meaningful, how accurate is our process to record it? Let's take the second, simpler, question first.

We began to 'reverse engineer' summaries to identify matching source sentences after the 2005 pyramid data were made available. Initially we used the public domain `amatch` 'approximate match' Perl module and achieved about 85% success when one-in-four token difference was allowed (Copeck & Szpakowicz 2005). In 2006 we added a secondary partial match facility which increased the hit rate to 95% (Copeck, Inkpen, Kazantseva, Kennedy, Kipp, Nastase & Szpakowicz 2006). Table 1 shows that

	2007		Prior	
Source Sentences	12832		33204	
Summary Sentences	2846	*100%*	9315	*100%*
linked to SCUs	1692	*59%*	4683	*50%*
linked to source texts	2715	*95%*	8868	*95%*
not linked to source texts	131	*4.6%*	447	*4.8%*

Table 1: Counts and Percentages of Summary Sentence Linkages, 2007 and prior

results this year were on a par with 2006 (statistics for prior years are based on the original data rerun with the augmented matcher). Further, inspection of the sentences which are not matched again shows that these are generally fragments, often produced by the truncation of a summary at the 250-word mark. With three years' experience and detailed runtime logs, we feel confident that we are accurately linking SCU annotations to source document sentences.

In our opinion the assessments which human judges make of summary *content* (previously *responsiveness*) and *fluency* are the best measures of summary quality. Of these two, fluency is not pertinent to measures of summary content. Therefore, to answer the first question posed at the opening of this section, we annually calculate correlation coefficients between the Pyramid evaluation participants' Modified SCU Scores and their content scores averaged across all topics for which Pyramids were constructed. In 2005 and 2006 these ρ values were 0.79 and 0.84 respectively. This year however correlation fell off significantly—to 0.53.

Figure 2 may help explain why this occurred. On two scatter plots we show the participants' Modified SCU and Content scores for 2007 and the two prior years on the same axes. The charts show that in 2007 SCU scores increased markedly, but without a commensurate improvement in summary responsiveness, in the eyes of human judges. We have become better at picking sentences which score well on Pyramid grounds, but not necessarily better at summarizing.

3.2 Recent Developments

In 2007 the administration and coordination of Pyramid evaluation moved from Columbia University to NIST and Microsoft Research. Participation in the peer annotation effort fell by half this year from 21 to 11 despite overall participation in the conference remaining on a par with previous years (2007: 30; 2006: 34, 2005: 31). Annotators had 23 unique topics to mark up and to check in 2007, a number comparable to prior years (2006: 20; 2005: 19).

In our 2005 workshop paper, we noted that the limited number of summaries furnishing sentences to be sought in source documents made it highly likely that other sentences, equally well-suited for use in a summary, would not be annotated in our corpus simply because they had not been singled out by any system. With 2007's markedly reduced participation in SCU evaluation, that caution is even more pertinent this year.

For the second consecutive year document collections contained fewer sentences on average than the year before. The 45 topics summarized this year averaged 553 sentences, 77% of 2006's 721 sentences and 58% of the 940 sentences in 2005 topics. 25 documents were provided for a topic in each of the last two years, while 2005 topics had varying numbers of sentences which averaged to 32.

Although the phenomena cannot be viewed in any way as related, this situation was paralleled by a similar second-year drop in the number of SCUs identified by creators in topic Pyramids. Pyramid SCUs averaged 69 in 2007, down from 80 in 2006 and 119 in 2005. The

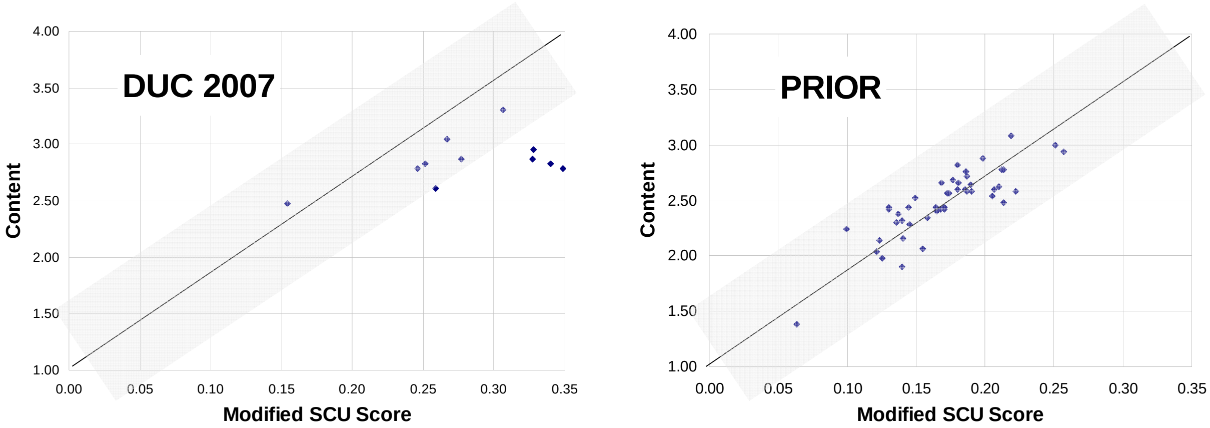


Figure 2: Content versus Modified SCU Score, 2007 and Prior

markedly higher number in 2005 can be accounted for in part by the fact that Pyramids that year were based on seven manual summaries, while subsequent years constructors have used four. More source material, more SCUs found.

Insofar as this statistic can be taken as an oblique check on the quality of Pyramid construction in the changed circumstances in which the Pyramid evaluation occurred in 2007, the verdict is mildly positive: 69 SCUs per topic is just somewhat lower than 80.

3 Work on the DUC System

Our team members are mostly faculty and graduate students with a variety of interests. This dictates our *modus operandi*. While some of our system's functional components remain largely unchanged from one version to the next, each new implementation also incorporates design elements reflecting the current varied research pursuits of individuals, insofar as these can be brought to bear on the summarization task. This is in marked contrast with systems developed in those organizations where, for various reasons, development and refinement of a single overarching system design can be pursued for a number of years. One consequence of this system architecture is that the results our system produces are quite variable from one year to the next as different components are swapped in and out for evaluation in the conference.

Components in our system which remain the same from year to year tend to be those whose role in the summarization process is generally well understood and uncontroversial. The sentence boundary detection code mentioned in the introduction is a prime example. Another is the family of subroutines that gather and manage data which characterize the sentences in a collection of documents on a topic—the topic *data model*. Development of these components tends to involve either improving the code, as additional processing experience highlights flaws and neglected cases (the first example), or increasing the range or functionality of the module in question (the second example). A third class of unchanging components are those which improve the summary in some way: the fruit of past labour. One instance is the subroutine to replace pronouns with their referents when these can be inferred with high confidence; another is the filtering-out of redundant sentences from the queue of candidates just prior to summary output. Taken together these modules make up the framework of the program.

Components in our system which do change substantially from one version to the next are those which are central to summarization based on sentence extraction—ones which use the resources provided by the system to determine which sentences will ultimately appear in the summary. While the extent of available resources ultimately establishes a ceiling for their efficacy, to our way of thinking these subroutines lie at the heart of the summarization process.

3.1 The Main Task

This dichotomy was in evidence in the 2007 design of our system. Changes were made both to the program framework and to the core program which computes sentence ratings and thereby determines the summary contents. We will summarize the framework changes first, before turning to the more interesting issue of sentence rating.

3.1.1 The Program Framework

We have referred to the sentence boundary detector code in the discussion of this year's experience in developing the SCU-marked corpus. Small changes were in fact made to that module this year; and additional instances of wire service formatting were identified and stripped from sentences in the *normalization* process (Copeck & Szpakowicz 2003). A number of filters used to improve fluency of the summary output were moved from the output stage to the normalization process, a more suitable place in the processing sequence. An alternative version of any sentences with pronouns in which these were replaced with their likely references was stored in the data model rather than computed on the fly. While most of these alternative strings are not used, augmenting the sentence data record with an edited alternative will facilitate further and more aggressive sentence editing in the future. Finally, the summarization program was reorganized into a series of Perl program libraries based on function, making its subroutines more accessible and more easily managed. Code which addressed obsolete tasks defined in past conferences was simultaneously removed and archived. Previous versions of the program had supplemented the topic data model with auxiliary data stores: a *lexicon*, containing counts and labels of all content words in the topic; and a *phraseology*, listing the same information for stopword-delimited phrases rather than for single tokens. To these was added a model of the topic containing data at the document rather than sentence level—the number of words and sentences in each. Intended to facilitate computation, this *document* store is yet another example of the burgeoning record of data being gathered about each topic.

The most interesting development in the program framework for DUC 2007 was work done to improve the order in which sentences are put in the summary. Hitherto these appeared in the summary in the same order in which they are stored in the data model. While this is defensible when the data model is based on a single document, as was the case with tasks in early conferences, the scheme is questionable when the topic data model is based on a number of documents. Sentences at the end of one document then precede in the count those beginning the next document, and the resulting sequence is incoherent.

A more thoughtful organization would bring together sentences on the basis of their semantic similarity; such sentences appear to talk about the same matters. At a minimum, such an ordering would eliminate situations when a sentence mentioning other concepts is interjected between two which do address a single topic. People are confused when this happens, since they expect to read a coherent narrative.

The approach we followed first constructs a diagonal matrix recording a measure of agreement between pairs of sentences in the summary. Agreement is computed as the number of matching tokens in the two sentences normalized over their length in tokens. The current first sentence in the sequence is taken as the starting point for organizing the summary on the grounds that it has the absolute highest rating among sentences in the topic document collection. The sequence of summary sentences is then extended by traversing the matrix and adding to the sequence that sentence which has the highest measure of agreement with the current last sentence in the sequence, until the set of sentences is exhausted. This produces a sub-optimal organization, but has the benefit of running in polynomial time.

3.1.2 The Program Core

The key process in our summarization program uses whatever salient information we have been able to collect on sentences in the topic document collection, to rank them on their suitability for use in a summary to meet a specific information request. Once ranking has occurred, summarization reduces to assembling the highest-ranked sentences into fluent text.

As noted earlier in this section, each year the sentence-ranking scheme in our system changes as we try alternative approaches; and such was again the case in 2007. Sentences this year were scored by a vote of three different ranking mechanisms, with instances of each scheme's values normalized over its output range to ensure that it received equal weight in the vote. The three approaches used were 1) a slightly-improved version of the *graph-matching* algorithm used to rank sentences the previous year (Copeck *et al* 2006); 2) a scheme which added the topic title to the information request to produce a single *conflated query*; and 3) the use of a suite of machine learners to *predict the likelihood* of a sentence realizing a SCU.

The graph-matching approach was discussed in detail in last year's workshop paper. The reader is directed there and to another paper at the NAACL 2006 TextGraph workshop (Nastase & Szpakowicz 2006) for more information.

3.1.2.1 Query Conflation

To test the hypothesis that a topic's title would meaningfully supplement the specification provided in its task information request, we considered the two together in one rating scheme. A subordinate objective was to try an alternative approach to matching sentences to the query.

The process consisted of measuring the similarity between each sentence and a query consisting of a title such as *World-wide chronic potable water shortages*, and an information request like *What countries are having chronic potable water shortages and why?*. We computed the similarity of a sentence to each part, rewarding similarity to the title more on the grounds that it is less likely to contain irrelevant tokens. The similarity score S was computed according to the following formula:

$$S(\text{sent}, \text{query}) = S(\text{sent}, \text{InfoReq}) + 2 * S(\text{sent}, \text{title})$$

In order to calculate how similar two sentences are, we compute the overlap of content bigrams and unigrams in each, rewarding overlap between parts of speech that are likely to be more salient. Each sentence is lemmatized and part-of-speech tagged using the *MontyLingua* software package (Liu 2004). All function words are removed. The open classes (nouns, verbs, adjectives and adverbs) are assigned an experimentally determined factor meant to reward matches between word-pairs in more salient categories. Nouns and verbs have the highest factor (2), followed by adjectives (1.5) and adverbs (1).

When processing a sentence, we collect a list of all content unigrams and bigrams. The similarity score between a pair of sentences is computed by counting the number of overlapping unigrams and bigrams. When the POS-tags of two candidate lemmas also match, the score is multiplied by the corresponding factor. Bigram matches are rewarded more highly than unigram matches by a factor of 2. The final score is normalized over the joint sentence length.

3.1.2.2 SCU Likelihood Prediction

In 2007 we sought to put the information present in the SCU-marked corpus to use in predicting the likelihood that a previously unknown sentence would realize a SCU. We used version 3.5 of the open-source Weka machine learning environment (Witten & Frank 2005) to train twelve of the more than 100 machine learners it provides on the 47 topics in the corpus from 2005 and 2006, saving each model thus produced for future use. Learners were chosen that 1) accept numerical values, 2) output class predictions, and 3) run in reasonable time on the datasets involved.

During the submission run the summarizer system generated an ARFF file from each 2007 topic data model and submitted it to each stored Weka learner model. Nineteen input features were provided in this data file. While most were direct transcriptions of the values of surface syntactic features, two were replaced with computed counterparts when it was clear that features in the data model would have little predictive value taken as-is. Thus a sentence's count of topical words was normalized by dividing it by the sentence length in tokens. Similarly a value identifying a sentence's overall position in its document was computed to convert the information in two of the data model's features, sentence-position-in-paragraph and paragraph-position-in-document, into a usable format.

Weka was then called to apply each stored learner model to classify sentences in the topic ARFF file. The learner assigned each sentence to one of two classes: those that realize / do not realize a SCU. These predictions were accumulated to produce an overall score for the likelihood of each sentence in the topic realizing a SCU.

Although multiple learners were employed for the simplest of all reasons—they were readily available—such a technique is supported by research into model ensembles (Caruana, Niculescu-Mizil, Crew & Ksikes 2004). A plausible hypothetical model also exists which envisions the style of individual authors, smoothed by editors and constrained by the conventions of the news report genre though they may be, still remaining sufficiently idiosyncratic to foil any single learner. In such heterogeneous domains the use of multiple learners, each capable of recognizing a particular style or family of styles, is hypothesized to be most effective.

3.2 The Update Task

DUC in 2007 introduced as a pilot the task of producing summaries which update a reader's existing knowledge regarding an information request with whatever pertinent new information is provided in additional documents. Our efforts focussed primarily on extending the architecture of the existing system to accommodate this pilot task in a way which was faithful to its specification. The approach we took was to produce the first summary in the usual manner—the initial summary is not an update. The data model describing the initial documents is the basis for the *master* data model for the topic. On each successive update, a separate data model is constructed describing the documents in the update collection (we called this collection a *group*). The update summary is then selected from sentences in the update group based on a comparison of its data model with the topic master data model. After the summary has been produced the group data model is merged with the topic master data model, updating the latter and ensuring that subsequent updates will compare the correct two models. This process can be continued indefinitely.

The difficult part lies in finding an appropriate basis on which to compare the update data model with the topic master. Our approach was to extend the data model by adding a *novelty* feature to measure the extent to which content words in the phrases composing each update sentence do not appear in the set of those in sentences in the master document collection. In both cases run-together phrases were disambiguated using a simple grammar to recognize conjoined NP, VP pairs. We assumed that such novel phrases would likely express information of interest to the update summary reader.

The novelty metric was employed in a formula to compute rankings for sentences in the update group. The formula adjusted the ranking up or down to a maximum of 50% based on 1) the sentence's location in the document, 2) its likelihood of being SCU-ranked (see Section 2), and 3) the presence of pronouns apt to have external referents. The update summary was then selected from the top-rated sentences in the group.

4 Results

The changes discussed in the previous section had an effect on the ranking of our summaries in 2007. The greatest improvement was in the content rating, an outcome on which we place importance because we believe content is the most significant measure of a summary. Linguistic quality also improved slightly. On the three computed measures of BE (Basic Elements) and ROUGE (SU4 and 2 submeasures) we moved to the middle of the group, which is also where we stood in the Pyramid evaluation.

As noted in Section 3, any improvement in our results in a given year is as likely as not to be reversed the year following, because each year our academic setting leads us to explore new and different approaches to sentence selection.

Our results on the Update task were uniformly deemed undistinguished by both human and automatic evaluators, who demonstrated a depressing degree of unanimity in this opinion. Clearly the novelty measure we used this year must either be rethought and reworked, or abandoned.

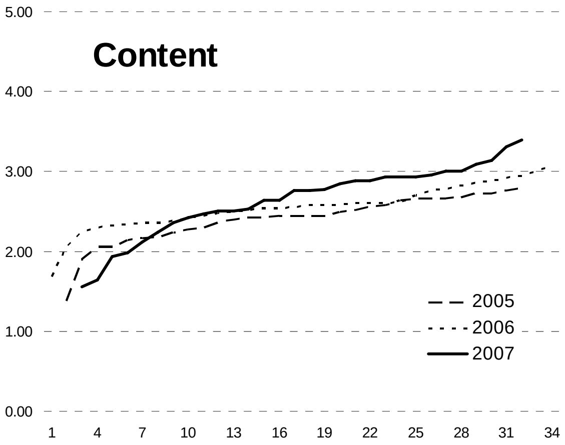


Figure 3: Content Scores, 2005 through 2007

This concludes the discussion of our particular performance in DUC 2007. However 2007 is the third year in which the conference has set participants a similar task. That consistency allows the inter-year comparisons shown in Figure 3 to be made and conclusions to be drawn about the performance of the conference participants as a whole.

Figure 3 charts the ordered average content ratings for peer participants over the three years in question. It suggests that as a group DUC participants are making small but measurable progress in producing summaries that successfully satisfy the conference's specified information request. That's good news.

The two other classes of evaluated summaries, those produced by human authors and baseline summaries, have not improved similarly. Nine or ten human-written summaries were assessed each year. Their average scores over the three years are 4.64, 4.75 and 4.71, while the single baseline study was rated 1.98, 2.04 and 1.87 (the second baseline introduced in 2007 has no earlier counterpart and is ignored). The absence of any evident trend in these data tends to rule out the possibility of grade inflation in peer performance over the period. That makes the good news even better.

5 Future Work

A goal for the next year would be to conduct more experiments in which our team judges summary responsiveness and fluency internally with the objective of improving the sentence selection process through trial and error or by iterative refinement. Such a labour-intensive activity must however be balanced against the competing demands and timelines of the other constituents of the research.

We will continue to update the corpus of SCU-marked topics with new material as it becomes available, and to use it to guide future development of our summarization system.

Acknowledgements

This year we would like to venture outside our local situation to acknowledge two people whose efforts have benefited us. Lucy Vanderwende merits a round of applause from all DUC participants in the summarization community for taking over management and coordination of the Pyramid rating activity. We would also like to thank Vivi Nastase, now at EML Research in Heidelberg, who repeatedly brought to our notice the concepts underlying the summary sentence reorganization code implemented in this year's system until we finally paid attention.

Partial support for this work comes from the Natural Sciences and Engineering Research Council of Canada.

References

Caruana, Richard, Alexandru Niculescu-Mizil, Geoff Crew and Alex Ksikes. 2004. Ensemble Selection from Libraries of Models. *Proceedings of the 21st International Conference on Machine Learning* (ICML'04).

Copeck, Terry, Diana Inkpen, Anna Kazantseva, Alistair Kennedy, Darren Kipp, Vivi Nastase and Stan Szpakowicz. 2006. Leveraging DUC. *Proceedings of the Workshop on Automatic Summarization* (DUC 2006), HLT/NAACL-2006.

Copeck, Terry and Stan Szpakowicz. 2005. Leveraging Pyramids. *Proceedings of the Workshop on Automatic Summarization* (DUC 2005), HLT/EMNLP-2005.

Copeck, Terry and Stan Szpakowicz. 2003. Picking Phrases, Picking Sentences. *Proceedings of the Workshop on Automatic Summarization* (DUC 2003), HLT/NAACL-2003.

Hugo Liu. 2004. MontyLingua: An end-to-end natural language processor with common sense. web.media.mit.edu/~hugo/montylingua.

Nastase, Vivi and Stan Szpakowicz. 2006. A Study of Two Graph Algorithms in Topic-driven Summarization. *Proceedings of the Workshop on Graph-based Algorithms for Natural Language* Processing (TextGraphs2006), HLT/NAACL-2006.

Nenkova, Ani and Rebecca Passonneau. 2004. Evaluating content selection in summarization: the pyramid method. *Proceedings of the Workshop on Automatic Summarization* (DUC 2004), HLT/NAACL-2004.

Witten, Ian H. and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques, 2nd ed.* Morgan Kaufmann, San Francisco, 2005.