Performance of a Three-Stage System for Multi-Document Summarization

Daniel M. Dunlavy
University of Maryland
ddunlavy@cs.umd.edu

John M. Conroy, Judith D. Schlesinger
IDA/Center for Computing Sciences
{conroy, judith}@super.org

Sarah A. Goodman, Mary Ellen Okurowski
Department of Defense
{abbygood@hotmail.com, meokuro@nsa.gov}

Dianne P. O'Leary
University of Maryland
oleary@cs.umd.edu

Hans van Halteren
University of Nijmegen
hvh@let.kun.nl

1 Introduction

Our participation in DUC 2003 was limited to Tasks 2, 3, and 4. Although the tasks differed slightly in their goals, we applied the same approach in each case: preprocess the data for input to our system, apply our single-document and multi-document summarization algorithms, post-process the data for DUC evaluation. We did not use the topic descriptions for Task 2 or the viewpoint descriptions for Task 3, and used only the novel sentences for Task 4.

The preprocessing of the data for our needs consisted of term identification, part-of-speech (POS) tagging, sentence boundary detection and SGML DTD processing. With the exception of sentence boundary detection for Task 4 (the test data was sentence-delimited using SGML tags), each of these preprocessing tasks was performed on all of the documents. Details of each of these tasks are presented in Section 2.

The summarization algorithms were enhanced versions of those presented by members of our group in the past DUC evaluations (Conroy et al., 2001; Schlesinger et al., 2002). The enhancements to the previous system are detailed in Section 3.

Previous post-processing consisted of removing lead adverbs such as "And" or "But" to make our summaries flow more easily. For DUC 2003, we added more extensive editing, eliminating part or all of selected sentences. This post-processing is described in Section 4.

2 Preprocessing

Many of the preprocessing tasks were performed using tools created by the Edinburgh Language Technology Group (`http://www.ltg.ed.ac.uk/`). Specifically, various components of that group's LT TTT (v1.0) parsing system were used. These tools were chosen due to their flexibility in handling both SGML and ASCII text documents, as well as their capability in handling most of the preprocessing tasks required by our summarization tools. The remaining tasks were performed using tools created in Perl.

The main tool used for identifying the terms or tokens of a sentence, tagging each term with its part of speech, and detecting and tagging the sentence boundaries was LT POS (Mikheev, 2000), a tool in the LT TTT suite. LT POS is a probabilistic part-of-speech tagger and sentence splitter based on a combination of hidden Markov and maximum entropy models. The default models, trained on the Brown corpus (Francis and Kucera, 1982), were used in our system.

Prior to this year, we used the SRA NetOwl software with Named-Entity recognition and aliasing to identify terms. NetOwl also does stemming. We moved away from NetOwl largely to experiment with a simplified preprocessing system. Our goal is to rebuild the preprocessing starting with a simple process and to add "enhancements" only when they improve summarization.

To benchmark the change in the way we identified terms, we used 82 DUC 2001 documents for which we had tagged sentences that could serve as sources for the NIST human generated abstracts. Two HMMs were trained, the first to utilize the NetOwl preprocessing and the second to use with the new simpler preprocessor. Then a single-document extract summary of approximately 100 words was generated for each document, using the terms generated by each of the two methods. The outcome of a ten-fold cross-validation was that the new simple method gave an average precision of 60% while the more complicated NetOwl gave an average precision of 58%.

2.1 Parsing Files using DTDs

Using the SGML document type definition (DTD) for a document allowed us to determine the set of all possible

Task	DTD	Filename	SGML Tag	$stype$
2,3	ACQUAINT	acquaint.dtd	<TEXT>	1
			<HEADLINE>	0
4	FBIS	fbis.dtd	<TEXT>	1
			<TI>	0
			<H1>, . . ., <H8>	0
	Federal Register	fr.dtd	<TEXT>	1
			<SUMMARY>	1
			<SUPPLEM>	1
			<FOOTNOTE>	1
			<DOCTITLE>	0
	Financial Times	ft.dtd	<TEXT>	1
			<HEADLINE>	0
	LA Times	latimes.dtd	<TEXT>	1
			<HEADLINE>	0
			<SUBJECT>	0
			<GRAPHIC>	0

Table 1: Mapping SGML tags to $stype$ values.

SGML tags that exist in documents of that type. Using these tag sets, we distinguished which sentences 1) were candidates for extract summaries, 2) contained key terms or phrases that would aid in creating a summary, and 3) contained no useful information for the task of summarization. We created a new attribute, $stype$, for the SGML tag denoting a sentence boundary, <s>, in order to denote each of these three types of sentences. The possible values for this new attribute are 1, 0, and -1, respectively. Table 1 presents the values of $stype$ used for sentences embedded into the SGML tags encountered in the several types of documents used in the evaluation. Tags not shown are assigned $stype = -1$.

Choosing to embed information into the document itself instead of creating a processing module in our summarization algorithm allows us flexibility in using the information throughout the various stages of our system. Furthermore, it will allow us to expand the types of sentence classification without affecting the summarization system.

3 Sentence Extraction

Our summarization algorithm uses a hidden Markov model (HMM) to select sentences for a single-document extract summary. A pivoted QR algorithm is added to select from those to generate multi-document extracts. Details of both algorithms and how they are used for sentence scoring and selection are given in (Conroy and O'Leary, March 2001), (Conroy et al., 2001), (Schlesinger et al., 2002), and (Schlesinger et al., 2003). Improvements we made to our algorithm for this year are included here.

The HMM uses features based upon terms, which we define as a delimited string consisting of the letters a-z, minus a stop list, i.e., everything but Roman letters is considered to be a delimiter. (All text is first converted to lower case.) The preprocessing tools (2) identify the terms for the HMM.

The features we used for the HMM for DUC 2003 are different from prior years. While we previously used the number of terms in a sentence, we now use "signature" and "subject" terms:

- the number of signature terms, n_{sig}, in the sentence—value is $o_2(i) = \log(n_{sig} + 1)$;
- the number of subject terms, n_{subj}, in the sentence—value is $o_1(i) = \log(n_{subj} + 1)$;
- the position of the sentence in the document—built into the state-structure of the HMM.

The signature terms are the terms that are more likely to occur in the document (or document set) than in the corpus at large. To identify these terms, we use the log-likelihood statistic suggested by Dunning (1993) and first used in summarization by Lin and Hovy (2002). The statistic is equivalent to a mutual information statistic and is based on a 2-by-2 contingency table of counts for each term.

The subject terms are a special subset of the signature terms. These are terms that occur in sentences with $stype = 0$, for example, headline and subject heading sentences.

The features are normalized component-wise to have mean zero and variance one. In addition, the features for sentences with $stype$ 0 and -1 are coerced to be -1, which forces these sentences to have an extremely low probability of being selected as summary sentences.

The above process of extract generation was used for both Tasks 2 and 3 of DUC 2003. For Task 4, the novelty task, we made the design decision that our extracts would be taken from only the novelty sentence set.[1] We achieved this by overriding the sentence type of all sentences that were not marked as novel to be type -1. Thus the HMM would only give high scores to sentences that were labeled as novel.

The model was trained using the help of the novelty data given by NIST. We focused on only the novel sentences in this set. To strengthen the model further, we sorted the novel sentences by hand for 24 of the document sets. This process removed many sentences which were no longer relevant in isolation. These data were then used to train the HMM to score the sentences and determine which features should be included.

In particular, the training data helped determine the number of states for the HMM. The upshot was that a small state space, consisting of five states, two summary states and three non-summary states, was optimal. Empirically, the number of summary states roughly corresponds to the median length in sentences of human summaries.

Another feature we considered for our system was using query terms derived from the topic descriptions. We attempted to use this information in two ways. The first was to simply add an additional feature to the HMM. This approach actually decreased the precision[2] of the system! The second method we considered was using the derived query terms in conjunction with a retrieval system to rank each document. We hoped to use these document scores in conjunction with HMM sentence scores to generate the extract sentences. Unfortunately, the IR scores did not correlate strongly with the likelihood that a document's sentence would be chosen for the summary. We hypothesize that since the document collection only contains documents relevant to the query, the topic description terms do not add any additional information. Clearly, more analysis is required to determine why the topic descriptions did not help in the generation of the summaries.

4 From Extract to Abstract

The output from the Sentence Extraction component is a ranked set of sentences selected by the QR algorithm. The HMM tends to select longer sentences. This means that for a 100-word summary, needed for tasks 2, 3, and 4, the QR algorithm would usually select only 2 or 3 sentences from all those first selected by the HMM. We felt that so few sentences would not supply enough of the information we would like to see in a summary.

In order to include more sentences in the summary, we decided to eliminate parts of the top selected sentences that do not usually convey the most important information. Occasionally we lose something we should have kept but, in general, we gain. Shortening the selected sentences permits the inclusion of additional sentences, potentially gaining additional information. To accommodate this, the QR algorithm ranks sentences with about 300 words rather than the needed 100 words.

Full parsing and comprehension are too costly to pursue. We have done some initial investigation into using elementary discourse units (EDUs) (Carlson et al., 2002) to determine sentence structure, component parts, and the importance and relevance of those parts, and would like to use EDUs for the purpose of creating an abstract. Unfortunately, automatic parsing of EDUs is still not strong enough to meet our needs.

Instead, we chose to develop patterns using "shallow parsing" techniques, keying off of lexical cues. The sentences passed by the Sentence Extraction component were run through a part-of-speech (POS) tagger. Each sentence, in order of its ranking by the Sentence Extraction component, was matched against the various patterns. The following eliminations were made, when appropriate:

- sentences that begin with an imperative;
- sentences that contain a personal pronoun at or near the start;
- gerund clauses;
- restricted relative-clause appositives;
- intra-sentential attribution;
- lead adverbs.

4.1 Sentence Elimination

We eliminate two kinds of sentences in our summaries: imperatives and those "beginning" with pronominals. We determined that imperatives rarely contain novel information; in order for them to be effective and understood, they must reference information the reader already knows.

Sentences that have a personal pronoun close to the start of the sentence seem to fall into two categories: 1) they are preceded by a proper noun, in which case, they are fine to use; or 2) they do not have their reference

[1] While this strategy is defensible and perhaps prudent it prevented us from generating a summary for document set 323, which did not have any novel sentences. We conferred with Paul Over, who indicated this was an error in the TREC data.

[2] In these experiments we assume the extract summary length is known and, therefore, precision and recall are identical.

within the sentence in which they appear. In the latter case, the antecedent generally is in a preceding sentence.

Ideally, when a sentence begins with a pronoun, we should do analysis and identify its antecedent. This is a very difficult task. We have instead taken the approach of eliminating any sentence that begins with a pronoun unless its preceding sentence in the original document has already been included in the summary. Because this is a one-pass process, even if we later include the sentence with the needed reference, the pronominal sentence will *not* be re-included[3]. Elimination of these sentences *may* cause the loss of some important information, but *definitely* improves the readability of generated summaries.

The sentence eliminating rules are:

1. imperatives: if the first word is tagged as a verb "base form" (VB in our POS tagger), the sentence is considered an imperative and eliminated from use in the summary.

2. pronominals: if a personal pronoun (PRP in our POS tagger) appears within the first eight (8) words[4] of a sentence and it is not preceded by a proper noun, we check to see that the sentence immediately preceding it in the original document has been selected. If not, the sentence containing the pronominal is eliminated.

4.2 Clause Elimination

Three different kinds of clauses were eliminated: gerund clauses, restricted relative-clause appositives, and intra-sentential attribution. In addition to the patterns identified to locate the clauses to be removed, we utilized a simple heuristic that if the number of tokens to be deleted was greater than or equal to the number of tokens to be retained, the elimination was not performed.

Gerunds often comment on, rather than advance, a narration and therefore tend to be incidental. To eliminate a gerund clause, it must 1) be at the start of the sentence or immediately follow a comma, and 2) have the gerund (VBG) as the lead word or as the second word following a preposition (IN) or "while" or "during". The end of the clause is identified by a comma or a period. The following is a sentence from the training data with a gerund phrase that would be removed:

> More than 800 lives were lost when the 21,794 tonne ferry**, sailing from the Estonian capital Tallinn to Stockholm,** sank within minutes early yesterday morning in the Baltic Sea 40 km south west of the Finnish island of Uto.

[3]We need to do more evaluation to determine if this was a correct decision.

[4]Eight was chosen heuristically; we know of no evidence that proves it to be the correct number of words.

Restricted relative-clause appositives usually provide background information. Because they are always delimited by punctuation, they can be removed relatively easily. The patterns for these clauses look for specific words playing specific part-of-speech roles in the sentence: "who", "when", "where", "which", and "in which", and require the clause to follow a comma and end with a comma or period. A sentence from the training data with a restricted relative-clause appositive to be removed is:

> The Menendez family lived in the Princeton Area until 1986**, when they moved to California**.

While attributions can be informative, we decided that they could be sacrificed in order to include other, hopefully more important, information in the summary. Identifying intra-sentential attributions is not always easy. Our rules find some, but not all, cases. We developed a list of about 50 verbs (and their various forms) that are used in attributions. A verb must be found in this list for the clause to be considered for removal.

When an attribution occurs at the start of a sentence, we require it to terminate with "that", without any preceding punctuation. (We have not yet determined how to find the proper end of the many attributions that occur without this word.) For an attribution that occurs at the end of a sentence, it must follow the last comma of the sentence. The last word of the sentence must then be one of our specified attribution verbs. A sentence from the training data with an attribution to be removed is:

> **The federal Government's highway safety watchdog said Wednesday that** the Ford Bronco II appears to be involved in more fatal roll-over accidents than other vehicles in its class and that it will seek to determine if the vehicle itself contributes to the accidents.

4.3 Lead Adverb Elimination

For DUC 2002, we eliminated sentence lead words such as "And" and "But" since they did not add substantial information and often hindered the flow of the summary. For the same reason, in DUC 2003 we expanded this and eliminated any adverb (RB) that began a sentence.

4.4 Impact

We performed a "post mortem" analysis of the summaries we generated to see what problems we had created when we eliminated parts of sentences. (We did not evaluate the impact of eliminating entire sentences.) Both the number of bad sentences (fragments, missing words, etc.), identified as "problems" in Table 2, and the number of clauses that should have been removed but weren't, identified as "misses" in the table, were relatively small. In some cases, the problems were due to poor sentence splitting and not to any of the heuristics we applied.

DUC task	Number of Problems	Number of Misses	Total Number of Sentences
2	1	6	91
3	5	6	102
4	5	2	113

Table 2: Heuristic Errors

We also evaluated changes in the summaries after performing these heuristics. On average, summaries were about .5 sentence longer after the heuristics were applied. For each of Tasks 2, 3, and 4, Table 3 shows the average number of sentences per summary, the average number of words per summary, the number of summaries that were not changed by the heuristics, and the number and percentage of the total of each of the different phrases that were removed.

5 Results

Overall, our system, system 16, was comparable to the top systems as rated by mean coverage, as given by NIST. Figure 1 shows the rankings of the humans, machine (peer) systems, and baseline systems for each of tasks 2, 3, and 4. The systems are sorted by average mean coverage. Our system ranked third and second among the peer systems for tasks 2 and 3, respectively. For task 4, it was the top scoring peer system but was inched out by baseline 5!

However, the top scoring peer systems are extremely close. More precisely, using a one-way ANOVA test on the top six ranked systems gives a p value of 0.82, 0.87, 0.92 for tasks 2, 3, and 4, respectively. Under the null hypothesis, i.e., that the top scoring systems have the same mean and variance with regards to mean coverage, p gives the probability that these average mean coverages would be produced. Thus, mean coverage does not separate the top scoring systems.

Note that in tasks 3 and 4, one of the baselines was among the top six scoring systems. By comparison, the top scoring baseline for task 2 ranked twelfth and the corresponding p value was 0.07 meaning we can be 93% certain that the top 12 systems do *not* have the same mean coverage score.

Table 4 shows how our system fared on the twelve questions which measure readability on a scale from 0 to 4. We present the average score on each of the questions for system 16, the average peer system (excluding ours), and the average of the 10 humans. Overall, our system performed fairly well.

Two observations should be made. On question 8, noun resolution, we performed comparably to *humans* on task 2 yet worse than the average *peer* system on tasks 3 and 4. We will need to investigate this anomaly. On question 12, which measures the number of sentences which appear to be misplaced, *all systems* seem to perform at about the same level, which is still significantly below human performance.

6 Conclusion and Future Efforts

In conclusion, our system performed better than last year, but there is still a wide gap between human and machine performance. The remaining question is whether the improvements we would like to make to our system will help narrow this gap or not.

We will continue to improve our preprocessing by testing state-of-the-art POS taggers and sentence splitters as they become available.

The utility of query terms for sentence selection was disappointing. We will seek better ways to use these in our HMM/QR framework.

Many more heuristics should be developed to eliminate less useful parts of selected sentences. As mentioned in Section 4.2, many attributions are still not identified and we would like to find a way to do this more thoroughly. We would also like to eliminate most adverbs that occur but it is difficult to determine when they are needed and when they can be removed.

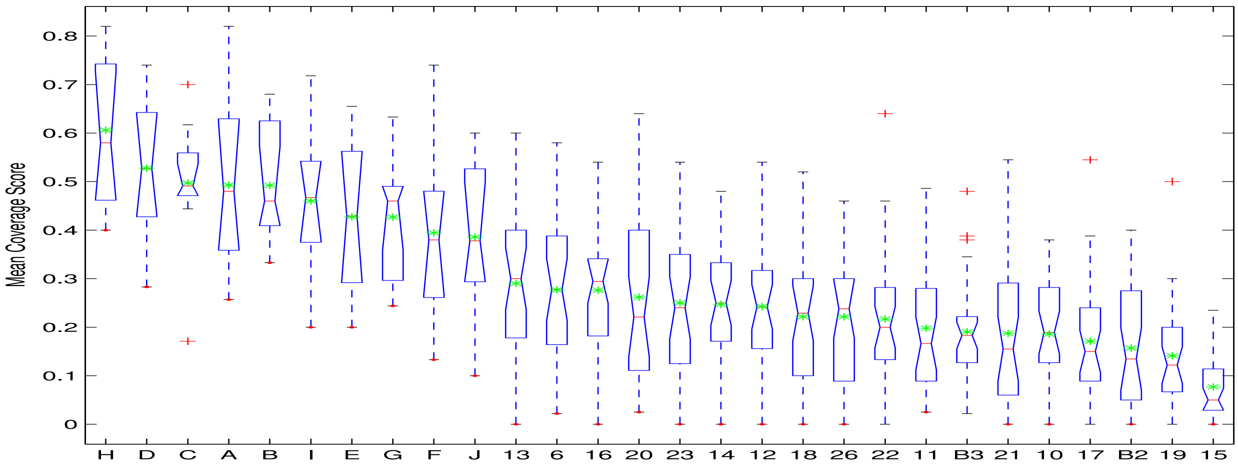
Additionally, we looked at eliminating all sentences with passive construction but found that we were not yet able to identify only the sentences we want to eliminate. Finally, we would like to eliminate the various types of parenthetic phrases—(...), [...], –...–, etc.—but have so far found it difficult to identify those which contain information that should be included in a summary.

References

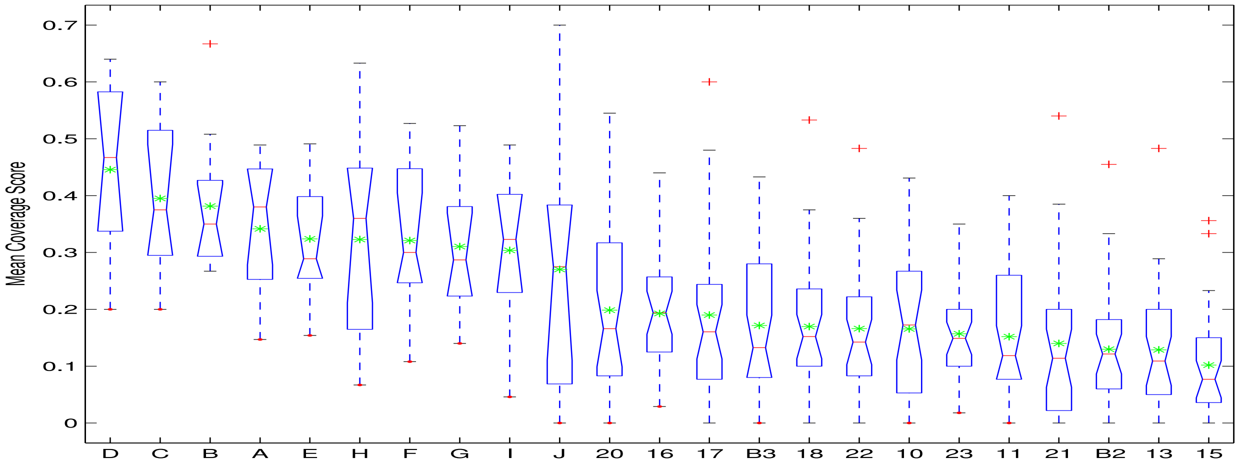
L. Carlson, D. Marcu, and M.E. Okurowski. 2002. “Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory”. In *Discourse and Dialogues*. Kluwer Academic Press.

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.

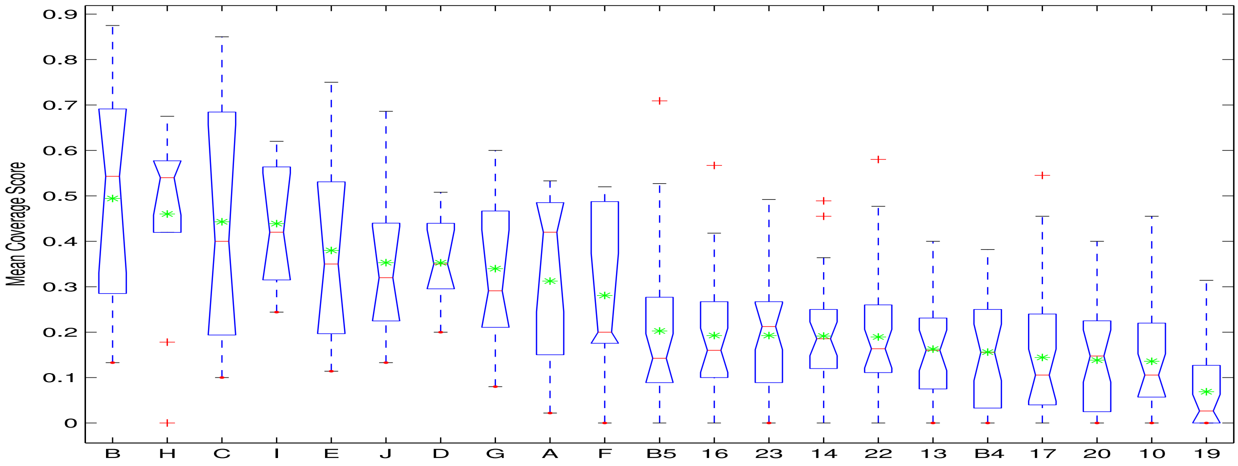
J.M. Conroy, J.D. Schlesinger, D̃.P. O’Leary, and M.E. Okurowski. 2001. “Using HMM and Lo-



(a) Task 2. Humans: A–J, Baselines: B2–B3, Peer Systems: 6–26



(b) Task 3. Humans: A–J, Baselines: B2–B3, Peer Systems: 10–23



(c) Task 4. Humans: A–J, Baselines: B4–B5, Peer Systems: 10–23

Figure 1: Mean Coverage

task #	before heuristics		after heuristics		unchanged	gerunds		rel-cl appos.		attributions	
	# sents.	# words	# sents.	# words		# words	%	# words	%	# words	%
Task 2	3.0	94.2	3.5	97.2	11	6.1	9.3	11.2	33.3	7.2	57.3
Task 3	3.1	102.4	3.6	98.8	11	10.7	4.8	16.4	32.3	9.6	62.9
Task 4	3.6	101.4	4.0	101.4	5	8.7	15.3	14.6	23.7	8.0	61.0

Table 3: Change in Sentence Length and Number of Words After Heuristics

	Task 2			Task 3			Task 4		
Question	System 16	Peers (mean)	Humans (mean)	System 16	Peers (mean)	Humans (mean)	System 16	Peers (mean)	Humans (mean)
1	0.0	0.5	0.0	0.3	0.5	0.0	0.0	0.2	0.0
2	0.0	0.1	0.0	0.1	0.2	0.0	0.1	0.0	0.0
3	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0
4	0.1	0.2	0.1	0.2	0.2	0.0	0.2	0.2	0.0
5	0.1	0.2	0.0	0.3	0.2	0.0	0.2	0.2	0.1
6	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
7	0.0	0.2	0.1	0.1	0.2	0.0	0.1	0.1	0.0
8	0.1	0.3	0.1	0.3	0.3	0.0	0.5	0.4	0.0
9	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.3	0.3	0.0	0.1	0.1	0.0	0.0	0.1	0.0
12	0.6	0.6	0.2	0.6	0.6	0.1	0.7	0.6	0.1

Table 4: Average Error Rates on Readability Questions (Scale: 0–4)

gistic Regression to Generate Extract Summaries for DUC". In *DUC 01 Conference Proceedings*. http://duc.nist.gov/.

T. Dunning. 1993. "Accurate Methods for Statistics of Surprise and Coincidence". *Computational Linguistics*, 19:61–74.

W. N. Francis and H. Kucera. 1982. *Frequency Analysis of English Usage: Lexicon and Grammar*. Houghton Mifflin Company, Boston, MA.

C.Y. Lin and E. Hovy. 2002. "The Automatic Acquisition of Topic Signatures for Text Summarization". In *DUC 02 Conference Proceedings*. http://duc.nist.gov/.

A. Mikheev. 2000. "Tagging Sentence Boundaries". In *Proceedings of the First Meeting of the North American Chapter of the Computational Linguistics (NAACL)*, pages 264–271, Seattle, WA. Morgan Kaufmann.

J.D. Schlesinger, M.E. Okurowski, J̃.M. Conroy, D.P. O'Leary, A.Taylor J. Hobbs, and H.T. Wilson. 2002. "Understanding Machine Performance in the Context of Human Performance for Multi-document Summarization". In *DUC 02 Conference Proceedings*. http://duc.nist.gov/.

J.D. Schlesinger, J.M. Conroy, M.E. Okurowski and D.P. O'Leary. 2003. "Machine and Human Performance for Single and Multidocument Summarization". *IEEE Intelligent Systems*, 18(1):46–54, Jan/Feb.