Columbia at the Document Understanding Conference 2003

Ani Nenkova, **Barry Schiffman**, **Andrew Schlaiker**, **Sasha Blair-Goldensohn**,
Regina Barzilay, **Sergey Sigelman**, **Vasileios Hatzivassiloglou**, and **Kathleen McKeown**

Department of Computer Science
Columbia University
1214 Amsterdam Avenue, New York, N.Y. 10027

Contact email: kathy@cs.columbia.edu

1 Introduction

The Columbia Summarizer for DUC 2003, Task 2, is based on the multi-document summarization system that we developed for DUC 2002 (McKeown et al., 2002). It uses different summarization strategies depending on the type of documents in the input set. Four different strategies are used, one for single events, one for multiple related events, one for biographies and one for discussion of an issue with related events. The summarization strategy encoded in MULTIGEN is used for single event document sets. All other strategies are encoded in DEMS, which uses different parameter settings for different document input types.

For Task 4, we adapted an open-ended question answering system that we have been developing as part of our AQUAINT project. The version used in DUC contains only a subset of the different techniques contained in the AQUAINT system given the differences in problem specification.

In the following sections, we first provide a system description, focusing on the changes that we made this year. These included changes in routing of documents to summarization strategy, the addition of revision rules to rewrite references to people, a new version of the MULTIGEN component that is used to generate, rather than extract, sentences for the summary, and the module for focus-based summarization which we based on our AQUAINT project. We then turn to a discussion of the evaluation results.

2 System Description

2.1 Routing

The MULTIGEN subsystem of Columbia's summarizer is targeted towards input sets that focus on a single event and contain multiple descriptions of that event from different sources; MULTIGEN's algorithms are largely based on repetition of nearly the same information across multiple sources. DEMS is targeted towards input sets that contain loosely related articles. Thus, every input to the summarizer is first examined by a fully automatic routing system that decides which of the two summarizers should produce the summary. The decision is based on the overall similarity of the documents in the cluster, as well as on considerations of the time span covered by the articles. During DUC 2002 the time constraints on single events were specified as "one event within a seven day time span". In addition, training data was available for fine-tuning our system. For DUC 2003, though, there were no specific constraints provided for the test clusters and there was no training data that could help us adjust the similarity threshold for routing the input. Our approach is based on an iterative approach to adjust the constraints for routing an article set to MULTIGEN. Initially, we decided on a strategy in which MULTIGEN gets all document sets that have at least three separate articles all falling within a period of seven days. If no set within the entire test set would be assigned to MULTIGEN, the timespan is iteratively relaxed to eight, nine, and finally ten days. Unfortunately, none of the DUC 2003 clusters had more than two sources and since we wanted to test both components of the summarizer, we relaxed the source requirement to at least two sources. This resulted in six articles (all written within seven days) being routed to MULTIGEN.

2.2 Summary Rewriting

We developed a summary rewriting module (Nenkova and McKeown, 2003) that normalizes all references to people in the summary. This module uses the IBM NOMINATOR named entity recognition system (Wacholder et al., 1997) to find references to people in both the summary and the entire input set. Identified references to people in the summary are rewritten, so that the full name of the person and any descriptive modifiers (e.g., *French President Jacques Chirac*), if available, replace the first mention of that person in the summary, and any subsequent mentions only use the last name (*Chirac*). Modifying reference realizations in this manner is intuitively appealing since it reduces unnecessary repetition of description modifiers and avoids the problem of an underspecified reference which occurs when the first mention of a person in the summary is not the first mention of that name in the original text. Reference rewriting was based on a corpus study of the syntactic realizations of references to people in terms of name form and type and number of pre- and post-modifiers in human written texts. The transitions between realizations were modeled with a Markov chain and the currently implemented rewriting rules correspond to the highest probability path in the chain.

2.3 Generation in MULTIGEN

This year, we changed the fusion component of MULTIGEN, keeping the rest of the system intact. As in the previous version of MULTIGEN, the sentence fusion algorithm operates over *themes*, clusters of related sentences computed by the analysis part of MULTIGEN. Given a theme, the problem is to generate a concise and fluent fusion of information in this theme, reflecting facts common to all sentences. The fusion component uses input sentences for content selection (to select the phrases conveying common information) as well as for surface realization (to guide the combination process of the selected phrases). The result is a generated, rather than extracted, sentence.

During the content selection stage, our algorithm performs local alignment of dependency trees to identify repeated information across pairs of sentences. Our alignment of dependency trees is driven by two sources of information: a measure of similarity between two given words, and the similarity between the structure of the dependency trees. More specifically, the lexical similarity measure takes into account more than word identity: it also identifies similar words which appear as synonyms in WordNet or paraphrases according to the automatically constructed dictionary derived from large comparable corpora (Barzilay, 2003). In determining the structural similarity between two trees, we take into account the types of edges (which indicate the relationship between nodes); for example, it is unlikely that an edge connecting a subject and verb in one sentence corresponds to an edge connecting an adjective and noun in another sentence. We use dynamic programming to find the optimal local alignment of two trees. The high similarity regions of aligned trees, which we call intersection subtrees, are selected to be included in the fusion sentence.

Now, we need to put together intersection subtrees. We can not explore every possible combination, since the lack of semantic information in the trees prohibits us from assessing the quality of the resulting sentences. Instead, we select a combination already present in the input sentences as a basis, and transform it into a fused sentence by removing extraneous information and augmenting the fused sentence with information from other sentences. The selection of the basis tree is guided by the number of intersection subtrees it includes. Using the similarity function described above, we identify a centroid by computing for each sentence the average similarity score between the sentence and the rest of the input sentences, and then selecting a sentence with a maximal score.

Next, we augment the basis tree with information present in the other input sentences and delete extraneous subtrees. First, we add alternative verbalizations for the nodes in the basis tree and the intersection subtrees which are not part of the basis tree. For each node of the basis tree we record all verbalizations from the nodes of the other input trees aligned to a given node. Then, we prune off subtrees of the basis tree which are not part of the intersection. However, removing all such subtrees trees may result in an ungrammatical or semantically flawed sentence; for example, we might create a sentence without a subject. Therefore, we perform more conservative pruning, deleting self-contained components which can be removed without leaving non-grammatical sentences. As previously observed in the literature (Jing and McKeown, 2000), such components include a clause in the clause conjunction, relative clauses, and some elements within a clause (such as adverbs and propositions). Once these subtrees are removed, the fusion tree construction is completed.

Finally, the fusion tree is linearized into a sentence; this requires selecting the best phrasing as well as determining optimal ordering. Since we do not have sufficient semantic information to perform such selection, our algorithm is driven by corpus-derived knowledge. We generate all possible sentences[1] from the valid traverses of the fusion tree, and score their likelihood according to statistics derived from a corpus. This approach, originally proposed by (Knight and Hatzivassiloglou, 1995; Langklide and Knight, 1998), is a standard method in statistical generation. We trained a trigram model over 60 MB of news articles using the CMU-Cambridge Statistical Language Modeling toolkit (second version). The sentence with the lowest entropy is selected as the verbalization of the fusion tree.

2.4 Focus-Based Summaries

Columbia University's AQUAINT (Advanced Question Answering for Intelligence) project system was also leveraged against DUC Task 4, "Short summaries in response to a question." The application of our AQUAINT system to this task seemed a natural one, but the problem domain of the AQUAINT project is more general than that defined by Task 4, and consequently many of our system components were not of use here.

[1] In practice, we sample only $n = 20$ paths for efficiency reasons.

Specifically, the problem domain of Task 4 is greatly focused by its input specification: We are given, a priori, a set of sentences deemed pertinent to the target summary by a panel of human judges. One of AQUAINT's main points of research is the automatic identification of source material pertinent to a given question and, consequently, the portion of our system that focuses on information selection was not needed.

We were able to adapt our clustering algorithm which operates on a set of sentences which are assumed to be topically related and produces ranked clusters of sentences from the original set having similar content. A cosine distance calculation employing word-stem IDF features lies at the heart of this clustering technique. Stem IDF values initially calculated from a large independent corpus are augmented using an IDF that is local to the input set of sentences. This technique was employed based on the observation that often times, a given set of topically related input sentences would contain terms specific to their common topic or domain having significant IDF values. Without compensating for these significant term IDF values common to an input sentence set, resulting sentence clusters lose accuracy as outlier sentences conglomerate around such terms.

Once the clustering algorithm had produced clusters of similar sentences within the original input sentence set, a fitness function was used to choose sentences from these clusters to include in the final answer result. Iterating over the sentence clusters, this function chose one sentence from each cluster containing a small number of pronouns and fulfilling a word length requirement. These sentences, extracted from the cluster set in order of most to least central were concatenated to produce the answer result. The summary length restriction was used as the stopping criterion of the sentence extraction loop.

2.5 Problems in Columbia's submission

There were two major problems that arose. The first was already discussed in the router section—we were aiming at a fair distribution of the testing sets between our two summarizations components, MULTIGEN and DEMS. We had to change slightly the initially selected constraints that MULTIGEN receives input sets containing articles from at least three sources.

The second problem was that we neglected to include in the test run our preprocessing script that removes datelines and was successfully used in previous DUCs. The presence of datelines reduced the quality of the summary as evaluators considered initial words such as "LONDON" capitalization errors and unconnected sentence fragments. It also probably had a negative effect on our coverage results.

3 Evaluation

3.1 Content Evaluation

Columbia participated in two of the four tasks for the Document Understanding Conference 2003. In contrast to the previous DUC evaluations, there was only one length for each task (100 words for the tasks we participated in).

Peer summaries (created by systems, baselines, or by humans) were evaluated by human assessors against *model* summaries created by humans. Summaries were broked down into units (sentences for peers, approximately clauses for models), and each peer unit was rated according to how well it covered the content of one or more model units. The three content measures used in the evaluation are *precision* (the percentage of peer units matching at least one model unit), *coverage* (the average percentage of models matched by at least one peer unit, weighted according to how completely the content of the model unit was represented in the peer unit(s)), and *related but unmarked* (the fraction of unmatched peer units that were related to the subject of the model summary). A length-adjusted version of the coverage score interpolates two-thirds of the coverage defined above and one-third of a brevity bonus defined as

$$\frac{\text{target length} - \text{actual length}}{\text{target length}}$$

but only if the original coverage is non-zero. This adjustment is carried out separately for each model unit.

We computed average scores for all systems, baselines, and humans using both micro- and macro-averaging. Micro-averaging is the normal average across all model or peer units, while macro-averaging assigns the same weight to each document set and summary (i.e., is the average across document sets of the average of the corresponding metric within each document set).

3.1.1 Task 2

Thirty document sets of 10 documents each were drawn from the Topic Detection and Tracking corpus, each covering a particular event over a time span of a few months or less. The summaries are to be "general summaries, not focused in any particular way other than by the selection of documents and the topic." This instruction to the NIST summarizers is a potential difficulty for the Columbia system, since neither of the two summarization strategies has any capability to take a query or topic into account when building the summaries. Both the DEMS summarizer and MULTIGEN are intended to take a cluster of related documents as input and to return a summary that reflects the important content in those documents. However, it is not possible at this time to determine to what extent the NIST summarizers molded their

System code	Macroaveraged		Microaveraged		Topic-related unmarked units
	Coverage	Precision	Coverage	Precision	
B2	9.06%	52.50%	9.40%	52.03%	62.00%
B3	11.68%	76.67%	12.10%	72.63%	41.33%
06	18.24% (2)	80.67% (7)	18.57% (2)	79.65% (4)	37.33% (7)
10	14.52% (10)	81.67% (5)	14.85% (11)	84.42% (2)	26.67% (13)
11	14.90% (9)	82.28% (4)	15.09% (9)	79.45% (7)	30.00% (12)
12	14.36% (11)	72.44% (10)	14.54% (12)	71.77% (10)	58.67% (2)
13	18.90% (1)	81.11% (6)	19.44% (1)	79.59% (5)	35.33% (11)
14	**17.47% (6)**	**76.72% (9)**	**18.03% (5)**	**76.04% (9)**	**40.67% (6)**
15	5.52% (16)	26.98% (16)	5.91% (16)	27.35% (16)	42.67% (5)
16	17.92% (3)	80.50% (8)	18.16% (4)	79.05% (8)	36.67% (8)
17	9.84% (15)	68.06% (14)	10.18% (14)	67.01% (14)	45.33% (4)
18	15.19% (8)	69.11% (11)	15.48% (8)	68.10% (12)	51.33% (3)
19	9.96% (14)	68.50% (12)	10.05% (15)	67.37% (13)	36.67% (9)
20	16.55% (7)	86.94% (1)	17.04% (7)	86.67% (1)	22.00% (16)
21	12.58% (13)	68.50% (13)	12.70% (13)	68.69% (11)	36.00% (10)
22	17.79% (4)	86.67% (2)	17.82% (6)	83.61% (3)	26.67% (13)
23	17.56% (5)	84.89% (3)	18.41% (3)	79.55% (6)	26.67% (13)
26	14.29% (12)	66.70% (15)	15.03% (10)	66.01% (15)	68.67% (1)

Table 1: Summary of content measures for Task 2. The Summarizers starting with *B* are the baselines, those designated by letters are people. The numbers in parentheses next to the machine summarizers show the rankings.The coverage figures are length-adjusted. Columbia's results are in bold face.

summaries to the topics nor to see which systems were able to take the topic descriptions into account.

In both coverage (5th under micro-averaging and 6th under macro-averaging) and the proportion of related but unmarked summary units (6th), the Columbia summarizer was in the bottom of the top third of the 16 systems participating in this task (see Table 1). Our ranking was invariant under the length-based adjustment process. In precision, we fared somewhat worse (9th under both micro- and macro-averaging), placing in the middle of all participating systems. However, it is clear that most of the automatic systems are bunched in a rather narrow range of scores. We testedw whether the differences in coverage between the 16 automatic systems were statistically significant using a paired-sample T-test. Among the top ten systems (including Columbia's), none is significantly better than the other even at significance level 0.05. Of the remaining six systems, three systems were significantly worse than most of the top ten systems.

All but one system outscored the first baseline, B2 in Table 1, which took the first 100 words in the last article, assuming the articles are ordered chronologically. Thirteen of the systems scored better than the second baseline, B3 in Table 1, which takes the first sentences of the first n documents until the summary has 100 words.

Human summaries obtain approximately twice the coverage scores of automatic ones. In precision, the automatic summaries rivaled the human summaries, and many of the systems scored slightly higher than many of the human summaries. Since multiple marked peer units are often related to only one model unit, it is unclear how much of this success is due to simple repetition of points that are worded differently in the input documents.

3.1.2 Task 4

In this task, summarizers were given 30 questions and lists of sentences that had been chosen by humans as relevant to each question. The summarizers were to generate a summary that answers the question. The inputs were taken from the Text Retrieval Conference (TREC) Novelty Track, where TREC topics were used as the question or query.

The Columbia system fared relatively well against the other participants (See Table 2), ranking second or third in coverage, and fourth or fifth in precision, depending on micro- or macro-averaging and the use of the length-adjustment. Our system ranked third in related but unmarked units. In addition, for Task 4, NIST asked two assessors to rate each summary between 0 and 4 on how well it responded to the question. We averaged the scores of the two human judges, and the Columbia system was in a three-way tie for first place, three-tenths of a point ahead of baseline *B4* (see Table 3). The machine systems were bunched closely together with the exception of the weakest system. Nine of the ten human summaries scored

System code	Macroaveraged		Microaveraged		Topic-related unmarked units
	Coverage	Precision	Coverage	Precision	
B4	11.32%	68.61%	11.66%	62.50%	44.00%
B5	12.59%	65.56%	13.33%	63.83%	52.00%
10	10.38% (6)	70.83% (2)	10.81% (5)	69.57% (2)	36.67% (8)
13	10.73% (5)	53.61% (7)	10.53% (6)	54.84% (6)	58.67% (1)
14	**13.42% (2)**	**64.94% (5)**	**13.95% (2)**	**63.25% (4)**	**55.33% (3)**
16	12.14% (4)	68.21% (3)	12.15% (4)	67.24% (3)	56.55% (2)
17	8.44% (8)	55.28% (6)	8.96% (7)	54.26% (7)	51.33% (4)
19	4.79% (9)	32.22% (9)	4.60% (9)	29.03% (9)	48.00% (6)
20	8.50% (7)	48.28% (8)	8.95% (8)	48.54% (8)	48.67% (5)
22	12.79% (3)	65.94% (4)	13.07% (3)	61.32% (5)	44.00% (7)
23	13.67% (1)	82.22% (1)	13.97% (1)	83.33% (1)	20.00% (9)

Table 2: Summary of content measures for Task 4. The Summarizers starting with *B* are the baselines, those designated by letters are people. The numbers in parentheses next to the machine summarizers show the rankings.The coverage figures are length-adjusted. Columbia's results are in bold face.

between 3.4 and 3.8, while the tenth was only slightly ahead of the automatic summarizers.

As in Task 2, Tables 2 and 3 show that the machine systems are closely bunched in a narrow range. We ran the same pairwise T-tests with a 5% confidence level, and obtained similar results: There were no significant differences among the top six of the nine systems, and only one system lagged far behind the others. In addition, there were no significant difference between the top six systems and the two baselines.

The baseline systems in this task were unusually strong, and only the top four machine systems in length-adjusted coverage, including Columbia's, outperformed the first baseline, which is *B4* in Table 2, and only the top three systems outperformed the second baseline, *B5* in Table 2. For *B4*, 100 words are drawn from the first n relevant sentences in the first document, and for *B5*, the summary is taken from the first relevant sentence in each of the first n documents.

As in Task 2, the human summaries scored roughly twice as high as the machine systems in coverage, and both the automated systems and humans were evenly matched in precision.

3.2 Quality Evaluation

The quality of summaries was measured according to 12 questions concerning grammatical and discourse features of the summary. The ranking given to a particular summary for a quality question was based on the number of mistakes of the kind described in the question. The answers were chosen in every case from the following set of four ordered categories: no mistakes, 1–5 mistakes, 6–10 mistakes, or more than 10 mistakes. These very broad ranges made it very difficult to distinguish between systems' ratings. This year, only 100 word summaries were produced and included three sentences per summary on average. Given this number, it is almost impossible to imagine a summary with more than 10 errors in capitalization or grammar and 5 seems to be a more reasonable upper bound. Thus, systems got the same penalty points regardless of whether they contained 1 or 5 errors. The distribution of penalties for the different systems indicates that a finer measurement of penalties, especially at the low-penalty end of the scale, would improve the ability to distinguish system performance on this task.

System code	Responsiveness
B4	2.42
B5	2.28
10	2.20 (3)
13	2.15 (4)
14	**2.45 (1)**
16	2.45 (1)
17	1.88 (6)
19	1.17 (7)
20	2.07 (5)
22	2.28 (2)
23	2.45 (1)

Table 3: Responsiveness scores for Task 4. Each score is the average of two assessors for each summary.Columbia's resulsts are in bold face.

At a first glance, the results for this question are surprising since many system often made errors. We looked at the actual summaries to see why capitalization is such a severe problem and it turned out that all the inclusions of datelines, such as "LONDON (AP)" were considered capitalization errors. This incurred a heavy penalty for

system code	Q1	Q2	Q3	Q4	Q5	Q6
B2	0.7000	0.0333	0.0333	0.1000	0.2000	0.0000
B3	1.0000	0.1000	0.0333	0.1333	0.2333	0.0000
06	0.0000 (1)	0.0000 (1)	0.0667 (11)	0.0667 (2)	0.1000 (7)	0.0000 (1)
10	0.2333 (8)	0.0333 (4)	0.0000 (1)	0.1333 (9)	0.0667 (5)	0.0000 (1)
11	0.0667 (5)	0.0333 (4)	0.0667 (11)	0.1000 (7)	0.0333 (3)	0.0000 (1)
12	0.1000 (7)	0.0667 (8)	0.0000 (1)	0.0667(2)	0.0333 (3)	0.0333 (11)
13	0.9333 (14)	0.1667 (15)	0.0667 (11)	0.2000 (15)	0.4000 (15)	0.0000 (1)
14	**0.8667 (13)**	**0.1333 (13)**	**0.0667 (11)**	**0.1667 (12)**	**0.3333 (13)**	**0.0667 (14)**
15	0.5667 (12)	1.3333 (16)	1.4667(16)	1.5667(16)	1.0000 (16)	0.4333 (16)
16	0.0333 (4)	0.0000 (1)	0.0000 (1)	0.0667 (2)	0.0667 (5)	0.0000 (1)
17	0.0000 (1)	0.0333 (4)	0.0333 (7)	0.1000 (7)	0.0000 (1)	0.0333 (11)
18	0.4333 (10)	0.1000 (11)	0.0667 (11)	0.1333 (9)	0.1000 (7)	0.0000 (1)
19	0.4667 (11)	0.0667 (8)	0.0333 (7)	0.1667 (12)	0.2667 (11)	0.1000 (15)
20	1.2000 (16)	0.0333 (4)	0.0333 (7)	0.0333 (1)	0.2667 (11)	0.0000 (1)
21	0.4000 (9)	0.1000 (11)	0.0000 (1)	0.1667 (12)	0.1667 (9)	0.0333 (11)
22	0.0667 (5)	0.0000 (1)	0.0000 (1)	0.0667 (2)	0.0000 (1)	0.0000 (1)
23	0.9333 (14)	0.0667 (8)	0.0333 (7)	0.0667 (2)	0.3333 (13)	0.0000 (1)
26	0.0000 (1)	0.1333 (13)	0.0000 (1)	0.1333 (9)	0.1667 (9)	0.0000 (1)

Table 4: Quality evaluation for Task 2, Questions 1 to 6. Columbia is system 14, shown in bold

Columbia since we did not include our script for cleaning up datelines and thus every inclusion of a first sentence meant a penalty point for Q1. Columbia scored 13th in Task 2 and 3rd in Task 4, where there were fewer datelines in the input.

Questions 2 through 6 aim at measuring the grammaticality of the sentences in the summary. Again an inspection of the submitted summaries showed that even perfectly grammatical sentences extracted verbatim from the test set but including datelines were considered as a grammatical error; they were scored as missing component or unrelated fragment, for example.

Questions Q7 to Q9 aim at evaluating how well references to different entities are realized. Columbia came out 3rd, 6th and 4th for the respective questions in Task 2 and 6th, 5th and 7th in Task 4. For Task 2 we used our rewrite module to change references to people when appropriate. The effect of the module was not captured well by the evaluation for two reasons. First, few of the test sets were focused on people and thus there were not many mentions of people in most of the summaries. More importantly, some aspects of reference were actually recorded in Q12 that asks about repetitiveness in the summary. Comparisons between the human judgments and the actual summaries showed that often repetitive descriptions of people led the summary to be judged as "containing unnecessarily repeated information".

For Q10 Columbia came 1st (with four other systems) in Task 2 and 9th in Task 4. This is partly due to the fact that the rewrite module used for Task 2 also rewrites sentences that include initial discourse markers such as "and", "but" and "however" by removing them and adjusting the punctuation of the sentence.

On Q11, instances of unnecessarily repeated information, Columbia's system came out 2nd in Task 2 and 5th in Task 4. Both summarization strategies used for Task 2 have modules that ensure that sentences with duplicated information do not appear in the summary. Identifying such sentences in the first step in the summarization technique used in MULTIGEN and a special module for duplication checking is included in the final sentence selection in DEMS.

Q12 asks about how many sentences seem to be in the wrong place because they indicate a strange time sequence, suggest a wrong cause-effect relationship, or don't fit in topically with neighboring sentences. Columbia was 5th in Task 4 and 7th in Task 2, with one of the humans scoring on average worse than us. This question seems to capture the most significant problems with the summaries and all systems got their worst score on this question. In the Columbia Summarizer, we use two different sets of constraints for ordering in the different summarization strategies used for Task 2. In MULTIGEN, ordering is based on both coherence constraints and temporal constraints, while in DEMS ordering is based on temporal constraints alone. In the Task 4 summarizer, neither temporal nor coherence constraints were used, but may have been less essential because of the strong topical focus of input data. The initial goal of the question was to measure how well the sentences in the summary are ordered but we feel it ended up measuring the overall coherence of the summary. Sometimes it is difficult to see how summary sentences are related or how they fit together. Since incoherence seems to be problematic for all summarizers it might be well worth making finer distinctions in why sentences do not fit together – because they repeat the same information, because they indicate strange time sequence, because there is no apparent relationship between them etc.

3.3 Some issues with quality measurements

The silent assumption for the evaluation is that the summaries are plain text passages without any extra formatting or system specific annotation. Any deviation from this results in penalty in questions that were not meant to measure the "cleanness" of the text. Thus, a question about proper capitalization or grammar questions about missing parts or merging of unrelated material will receive a lower score even if everything except for the extraneous material (e.g., dateline or incremental results) is correct. Maybe adding an additional questions such as "Are there any system internal symbols or datelines present in the summary" could be used to distinguish these types of errors.

While we recognize that the presence of datelines reduces the quality of the summary, a penalization several times for the same error seems undesirable in general. In fact, a pairwise chi square test for statistical independence at the 0.05 significance level showed that questions Q2 to Q7 are not independent and that a penalty for one of them would most likely result in a penalty in any of the others. We also computed correlation coefficients between the different questions and they show that the grammar-related questions Q2 to Q4 are indeed strongly correlated with each other with correlation coefficients ranging form 0.73 to 0.68, and are less strongly correlated with Q5–Q8 and Q12 (coefficients between 0.48 to 0.34). In addition, Q5–Q8 are weekly correlated with each other (0.3 to 0.2). It is unfortunately not clear if this dependence is real or simply an artifact of the presence of datelines. It might be useful for future evaluations to adopt a more general question such as "Are there any major grammatical mistakes?" to measure this aspect of the summary quality.

4 Lessons Learned

The DUC evaluation has reached relative maturity with this third iteration. An increasing number of systems are participating and the different tasks introduce opportunities to measure the performance of state-of-the-art summarization techniques on different summarization applications and contexts.

Yet, one issue that has affected all DUC evaluations is the lack of certainty that the model summaries represent objective standards. People who are asked to summarize sets of multiple articles or even single articles tend to exhibit significant disagreement on what information they select, even when restricted to selecting only pieces of the original articles (Jing et al., 1998). This is demonstrated by the fact that applying the DUC metrics to judge a human-generated summary against another human-generated summary results in coverage scores only of about 35-50% and precision scores of 65-80%. Although systems obtain significantly less coverage compared to human summaries, the overall low scores for both human and system coverage indicate that the evaluation is not yet capturing the legitimate variability of summaries that does not affect its acceptability or likely usefulness of the summaries. This issue makes summarization evaluation a very hard problem in general. Significantly increasing the number of model summaries used and exploring ways to account for the interdependencies of information across model units in multiple models would partially address this problem.

In an effort to broaden the summarization task, NIST has experimented with a variety of datasets and conditions for summarization. Annotation is also expensive for this task, which has generally led to small-scale evaluations compared to efforts such as TREC. More importantly, it has also meant that almost no training data is available for system development and fine-tuning. This prevents a host of data-intensive, adaptive approaches from being used; we detailed in Section 2.1 how our system's adaptive routing component had to be based on a guess because we had no training data to estimate parameters. While training data does exist for cases when the evaluation exactly duplicates an evaluation of previous years, attempts to improve the evaluation by making changes (e.g., the change to TDT data this year) means that previous training data no longer applies. Providing modest training sets with associated model summaries would enable future DUCs to explore predictive models for adjusting the output to the expected standard. If such an undertaking is prohibitive, the provision of clear specifications for the data sets on broad parameters such as types of sources, number of closely related articles in each set, coherence of each set in topic (e.g., is the topic narrow such as "Canadian figure skaters receive gold medal" or broader such as "2002 Olympics"), and time span information could help systems incorporate and tune additional parameters that would guide the summary generation process, even in the absence of example articles.

Finally, an issue that surfaced near the time of this year's submission of results is the way the current formulas adjust coverage by giving a bonus to brief summaries. The originally published formula awarded a bonus of up to 33.3% for brief summaries. A strategy exploiting this formula would generate empty summaries—these would receive no base coverage but the maximum brevity bonus,

system code	Q7	Q8	Q9	Q10	Q11	Q12
B2	0.0333	0.1000	0.0000	0.0000	0.0667	0.0667
B3	0.0667	0.1667	0.1000	0.0000	0.6333	0.5000
06	0.0667 (8)	0.2000 (4)	0.0667 (12)	0.0000 (1)	0.3667 (14)	0.5000 (5)
10	0.1667 (12)	0.3333 (9)	0.0333 (4)	0.1000 (16)	0.3667 (14)	0.4000 (3)
11	0.0000 (1)	0.1667 (3)	0.0000 (1)	0.0667 (14)	0.1667 (4)	0.2667 (1)
12	0.0333 (3)	0.4000 (12)	0.0333 (4)	0.0333 (7)	0.0667 (1)	0.7333 (11)
13	0.1333 (10)	0.2333 (6)	0.0667 (12)	0.0000 (1)	0.5000 (16)	0.6000 (8)
14	**0.0333 (3)**	**0.2333 (6)**	**0.0333 (4)**	**0.0000 (1)**	**0.1333 (2)**	**0.5667 (7)**
15	0.8667 (16)	1.1000 (16)	0.1667 (16)	0.0333 (7)	0.3333 (12)	1.6667 (16)
16	0.0333 (3)	0.1000 (1)	0.0333 (4)	0.0000 (1)	0.3333 (12)	0.6000 (8)
17	0.3333 (14)	0.3667 (11)	0.0000 (1)	0.0333 (7)	0.1333 (2)	0.7667 (13)
18	0.3333 (14)	0.4333 (13)	0.0000 (1)	0.0000 (1)	0.2000 (6)	0.7667 (13)
19	0.0000 (1)	0.4667 (14)	0.0333 (4)	0.0333 (7)	0.2000 (6)	0.7333 (11)
20	0.0333 (3)	0.1333 (2)	0.0333 (4)	0.0000 (1)	0.3000 (10)	0.5000 (5)
21	0.2667 (13)	0.5000 (15)	0.0333 (4)	0.0333 (7)	0.3000 (10)	0.7000 (10)
22	0.1333 (10)	0.3000 (8)	0.0667 (12)	0.0333 (7)	0.1667 (4)	0.3667 (2)
23	0.0333 (3)	0.2000 (4)	0 .0333 (4)	0.0667 (14)	0.2333 (8)	0.4333 (4)
26	0.1000 (9)	0.3333 (9)	0.0667 (12)	0 .0333 (7)	0.2333 (8)	0.8333 (15)

Table 5: Quality evaluation for Task 2, Questions 7 to 12. Columbia is system 14, shown in bold.

for a total coverage score of 33.3%, higher than what any submitted system achieved! In recognition of this, NIST altered the rules before the submission of summaries, and specified that a coverage of 0 would not receive any bonus. However, this rule was implemented at the model unit (clause) level. As a result, summaries that concentrate their coverage on relatively few MUs will receive a significantly lower length-adjusted coverage than a same-length summary with a better spread of matches across MUs will, even if the second summary matches the MUs less well and has the same overall average coverage as the first one. In other words, the current length-adjustment formula adjusts not only for length, but also for distribution of coverage across MUs. Short summaries are then penalized twice: once because their starting base coverage will match fewer MUs since they will contain less text, and a second time because they will receive the length bonus only for a small proportion of MUs. Further, most summaries will miss most of the model units in any given model, as evidenced by the fact that median coverage across MUs is most often 0. As a result, the length-adjustment process *lowered* the score of all systems and humans by about 30% on average, even when they produced summaries shorter than the target length of 100 words. Clearly, further analysis and tweaking of the adjustment formula is needed so that it properly captures length and not content distribution effects.

References

R. Barzilay. 2003. *Information Fusion for Multi-Document Summarization: Paraphrasing and Generation*. PhD dissertation, Columbia University, April 2003.

H. Jing and K. McKeown. 2000. Cut and paste based text summarization. In *Proceedings of NAACL'00*.

H. Jing, R. Barzilay, K. McKeown, and M. Elhadad. 1998. Summarization evaluation methods: Experiments and analysis. In *AAAI Symposium on Intelligent Summarization*.

K. Knight and V. Hatzivassiloglou. 1995. Two-level, many-paths generation. In *Proceedings of ACL'95*.

I. Langklide and K. Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of COLING/ACL'98*.

K. McKeown, R. Barzilay, S. Blaire-Goldensohn, D. Evans, V. Hatzivassiloglou, J. Klavans, A. Nenkova, B. Schiffman, and S. Sigelman. 2002. Columbia's multi-document summarizer for document understanding conference. In *Proceedings of DUC'02*.

A. Nenkova and K. McKeown. 2003. References to named entities: a corpus study. In *Proceedings of NAACL-HLT'03*.

N. Wacholder, Y. Ravin, and M. Choi. 1997. Disambigaution of names in text. In *Proceedings of the Fifth Conference on Applied NLP*, pages 202–208.