Understanding Machine Performance in the Context of Human Performance for Multi-document Summarization

Judith D. Schlesinger[*] Mary Ellen Okurowski[†] John M. Conroy[‡]
Dianne P. O'Leary[§] Anthony Taylor[¶] Jean Hobbs[||] Harold T. Wilson[**]

Abstract

We present a discussion of our summarization algorithm—both single and multi-document—along with a discussion of the evaluation experiments we have undertaken, what we have learned from them, and how we intend to utilize this information.

1 Introduction

For DUC '01, we submitted both single- and mult-document summaries. Our single document summaries were generated by two different algorithms, a logistic regression model (LRM) and a hidden Markov model (HMM). Document sets were randomly chosen from each algorithm's output to create a single submission in order to receive feedback on the two models. Our results were reasonable, somewhere in the middle of all the submissions, but there was certainly much room for improvement.

Our multi-document summaries were poor. We discovered a coding error that caused us to choose *low* scoring sentences rather than high. Thus, we were eager for DUC '02 to assess the quality of multi-document summaries our system was actually capable of producing.

For DUC '02, we have fixed all known bugs, have made some modifications to the HMM, have merged the LRM with the HMM, and have added some heuristics to aid continuity. Additionally, we have performed extensive evaluations of the human generated summaries, especially the multi-document summaries, to identify ways to improve our system.

2 The Algorithms

2.1 Single Document Summaries

The two methods that we submitted last year ([CSO01]), the HMM and the LRM, were merged for DUC 02. We accomplished this by including all the features of the LRM ([SBD01]) in the HMM.

An HMM has fewer assumptions of independence than a naive Bayesian approach ([KPC95], [AOGL97]). Specifically, we expect the probability that the next sentence is included in the summary will differ, depending on whether the current sentence is a summary sentence or not, i.e., an HMM permits this dependent probability with marginal additional cost over a simple Bayesian classifier. Additionally, the HMM uses a joint distribution for the features set, unlike the assumption of independent features used by naive Bayesian methods.

Our Hidden Markov model for text summarization uses three features:

- position of the sentence in the document—built into the state-structure of the HMM.
- number of tokens (non-stop words) in the sentence—$o_1(i) = \log(number\_of\_tokens + 1)$.
- number of "pseudo-query" terms (defined below) in a sentence—$o_2(i) = \log(Pr(\log(number\_of\_pseudo - query\_terms + 1)))$.

The first feature was included explicitly in the LRM; it is implicit in the HMM as part of the state-space. Furthermore, the state space of the HMM incorporates the conditional probability that a sentence is a summary sentence given that the previous

[*] IDA/Ctr. for Computing Sciences, judith@super.org
[†] Department of Defense, meokuro@nsa.gov
[‡] IDA/Ctr. for Computing Sciences, conroy@super.org
[§] University of Maryland, oleary@cs.umd.edu
[¶] SRA, ataylor@sra.com
[||] Kathpal Technologies, Inc.
[**] Department of Defense, htwilso@nsa.gov

sentence was a summary sentence. The second feature was the same in both models.

The third feature requires identifying terms which are more likely to occur in the document (or document set) than in the corpus at large. We call such terms "pseudo-query" terms since they replace the role of the query terms that would be present in a query-based summary model. We identify these terms using the log-likelihood statistic suggested by [Dun93]. This statistic is equivalent to a mutual information statistic and is more robust than the Z-score we used in DUC 01. The statistic is based on a 2 by 2 contingency table of counts for each term. See [Dun93] or [LH 00] for details.

A fourth feature, the distance to the nearest query term, was used by the LRM presented at DUC 01. We found this feature to be redundant *for the DUC data* since almost all sentences had at least one psuedo-query term or were adjacent to a sentence which contained one. Consequently, including this feature does not improve the resulting summaries.

An adaptation to our feature set, that we made this year, is to include a heuristic to recognize boilerplate sentences. The above feature set is good at extracting key sentences in the document. However, it often chooses headlines and, on occasion, some long by-lines that appear in the DUC data. We have developed a number of heuristics to recognize such sentences[1]. We want to make sure such sentences are not chosen by the HMM. We do this by modifying the features of boilerplate sentences to have features that correspond to a sentence that is a non-summary sentence. Modification of the features, in lieu of just omitting a sentence, allows the HMM to exploit the position of the boilerplate sentence to its fullest.

An HMM handles the positional dependence, dependence of features, and Markovity. (For more details about HMMs the reader should see [BPS70] and [Rab89].) The model we propose has $2s+1$ states, with s summary states and $s+1$ non-summary states. A picture of the Markov chain is given in Figure 2.1. Note that we allow hesitation only in non-summary states and skipping of states only from summary states. This chain is designed to model the extraction of up to $s-1$ lead summary sentences and an arbitrary number of supporting sentences. Using training data, we obtain a maximum-likelihood estimate for each transition probability and this forms an estimate M for the transition matrix for our Markov chain, where

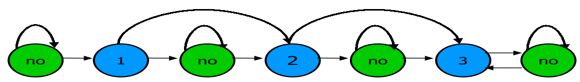

Figure 1: Markov Model to Extract 2 Lead Sentences and Supporting Sentences

element (i, j) of M is the estimated probability of transitioning from state i to state j.

Associated with each state i is an output function, $b_i(O) = Pr(O|state\ i)$, where O is an observed vector of features. We make the simplifying assumption that the features are multivariate normal. The output function for each state can be estimated by using the training data to compute the maximum-likelihood estimate of its mean and covariance matrix. We estimate $2s+1$ means, but assume that all of the output functions share a common covariance matrix.

With this model we compute $\gamma_t(i)$, the probability that sentence t corresponds to state i. We compute the probability that a sentence is a summary sentence by summing $\gamma_t(i)$ over all even values of i, values corresponding to summary states. This posterior probability, which we define as g_t, is used to select the most likely summary sentences. We refer the reader to [CO 01] for details.

2.2 Multiple Document Summaries

Two methods for multi-document summarization were investigated. Both methods use the HMM described earlier (Section 2.1) to score each sentence in the document *set* by posterior probability. We then take the top-scoring sentences as candidates for the multi-document summary. We select enough sentences to generate an extract of twice the maximum size requested, which for DUC would be 800 words.

The candidate sentences are then used to form a token-sentence matrix, which we call A. The columns of this matrix are normalized so that their 2-norm is equal to the posterior probability given by the HMM. We wish to choose columns from A which give good coverage of the tokens. We considered two approaches to solving this problem: pivoted QR factorization and

[1] While the performance of these heuristics is quite good, we may occasionally misidentify a sentence, both false positive and false negative.

a SVD based method due to [GKS76]. The latter method is known to be more robust for ill-conditioned matrices. An ill-conditioned matrix might arise in multi-document summarization if several sentences which collectively have a lot of overlap were selected for consideration by the HMM.

Pivoted QR factorization attempts to select columns of A in the order of their importance in spanning the subspace spanned by all of the columns. The standard implementation of the pivoted QR decomposition is a "Gram-Schmidt" process. See [CO 01] for details. The first r sentences (columns) selected by the pivoted QR are used to form the summary. The number r is chosen so that the summary length is close to the target length.

The second method is similar to the QR method and uses the same normalized matrix A. The SVD of A ($A = U\Sigma V'$) is then computed to determine a rank k approximation. The rank is chosen so that the difference in Frobenius norm between the original matrix and the rank k matrix is within 1% of the original norm. Finally a pivoted QR decomposition on the transpose of the first k columns of V is computed. The first r sentences (columns) selected by the pivoted QR are used to form the summary. The number r is chosen so that the summary length is close to the target length.

3 Training

Our work was set up to compare system generated summaries to human tagged extracts which we derived by having analysts map the abstracts for each of 148 documents (half of the training data) to the information source sentence(s) in the document. For a human, this tended to be a straightforward task, in contrast to our attempt at automatically generating the mappings. The analysts were able to easily handle abstracts containing a synthesis of information, which were especially difficult for the automatic process. These created extracts are the same as those we used in DUC 01 and they were shared with the other participants for DUC 02.

The analysts' extracts ensured that we now had the tagged sentences we needed for training and evaluation purposes. The analysts chose sentences to match, as closely as possible, the informative nature of the abstracts. This enabled us to switch between informative and indicative summaries, based on the training data used.

The HMM was trained and tested using 119 of the tagged documents. The remaining 29 documents were discarded during training due to problems with sentence boundaries as uncovered by Hans van Halteren, a fellow DUC participant. These extracts were generally longer than the required 100 word count, since the original abstracts often drew information from multiple sentences. The precision, as measured by number of sentences in the extract that agreed with the human extract, was used as a simple score. A ten-fold cross validation was done using the 119 extracts. The precision for the 100-word single document extracts was 0.55, which was an improvement over our precision of 0.52 for DUC 01. This gain was due largely to the use of the cleaner training data.

A further improvement was achieved by the boilerplate sentence recognition heuristics which increased the precision to 0.57 for 100-word single document abstracts.

We tested the multi-document summary methods by comparing the generated summaries directly with the human abstracts. The metric used was the cosine score. Both the pivoted QR and the SVD subset algorithm had comparable 0.53 and 0.52 cosine score for 400 word summaries. This is an improvement over our corrected QR based multi-document algorithm of last year which had a cosine score of 0.47.

4 Generating the Final Summaries

The HMM was used to generate single document extract summaries. Sentences were chosen by score, with the highest scoring sentences included until the 100-word length was met or exceeded by some constrained amount. The selected sentences were then reordered in their original document order to create the final summary.

The multi-document summaries were generated by the pivoted QR method described in the previous section. The sentences are output in their original document order and the document order is lexicographical, which has the side effect of a temporal order within a group of documents from the same source (e.g. Wall Street Journal) due to the naming convention.

After the sentences were selected for either a single or multiple document summary , we ran another set of heuristics which removed sentence starting discourse markers (And, Yet, But, etc.) as well as known boilerplate (bracketed words, very short phrases ending in ";", etc.). This greatly improved the cohesiveness of the generated summaries.

5 Results

Evaluating the results to determine our performance was, as usual, difficult. For the 200- and 400-word extracts, we used the f-scores calculated over the entire collection (TMEAN) as calculated by Hans van Halteren and shown in Table 1. Details on his calculations are available in his paper (see [vH 02]).

	200 words		400 words	
System	word	sentence	word	sentence
sys21	0.211	0.188	0.290	0.258
sys19	0.199	0.183	0.240	0.223
sys24	0.193	0.172	0.249	0.222
sys28	0.167	0.136	0.241	0.197
sys20	0.144	0.126	0.191	0.172
sys29	0.102	0.089	0.179	0.156
sys31	0.094	0.082	0.172	0.153
sys25	0.092	0.080	0.165	0.148
sys16	0.077	0.063	0.156	0.128
sys22	0.042	0.038	0.097	0.084
pbase	0.215	0.191	0.294	0.265

Table 1: Word- & Sentence-Based F-scores for 200- and 400-Word Multi-Document Extracts

For the single document, 50-, 100-, and 200-word multi-document summaries, we performed our own calculations: precision = # of marked peer units/# of peer units; recall = # of marked peer units/# of model units; and f-score = (2 * precision * recall)/(precision + recall). Tables 2 and 3 show the relative rankings of all the systems per our calculations. For multi-document abstracts, rankings were determined by summing the ranking of each system in each of the 50-, 100-, and 200-word summaries (10-word summaries were omitted since not all systems participated).

Overall, system performance on either extract or abstract summarization was low. Only two systems (sys27 and sys19) out-scored the single document summary baseline (last line, labeled "base1") score; *no* system out-scored the *extract* multi-document baseline (last line, labeled "pbase") score; and only a single system (sys19) consistently beat the *abstract* multi-document baseline (last line, labeled "base3"), with the second best system (sys26) beating it for two out of three cases.

[2] base1 was created by taking the first 100 white-space delimited, non-tag tokens in the document.

[3] base3 created by taking the first sentence in each consecutive document in the set until the summary length is reached. This was the higher scoring of the two available baselines. There was no baseline for 10-word summaries.

System	f-score
sys27	0.475
sys19	0.469
sys28	0.441
sys15	0.435
sys31	0.433
sys29	0.432
sys21	0.430
sys23	0.408
sys18	0.368
sys25	0.368
sys16	0.363
sys31	0.148
sys17	0.138
base1[2]	0.466

Table 2: F-scores for Single Document Abstracts

System	10 words	50 words	100 words	200 words
sys19	0.827	0.489	0.475	0.451
sys26	0.664	0.548	0.418	0.423
sys28	—	0.439	0.423	0.384
sys24	—	0.318	0.387	0.393
sys20	0.489	0.314	0.369	0.388
sys29	0.376	0.380	0.320	0.322
sys25	0.445	0.325	0.315	0.326
sys16	0.454	0.278	0.297	0.265
base3[3]	—	0.421	0.434	0.413

Table 3: F-scores for All Multi-Document Abstracts

Our own system (sys28) is fourth in the overall extract rankings and third for both the single- and multi-document overall abstract rankings. The following section (Section 6) discusses our investigation of system performance within the context of human performance and attempts to identify how we can improve our system performance.

6 Evaluation

Prior to entering DUC, our evaluation methodology was centered only on users of *indicative* single document summaries. We understand this task and, therefore, our system performs well. However, we don't have a good understanding of human performance in multi-document summarization nor of our system performance in the context of the DUC human model. We devised our own internal evaluation using the DUC 01 training and testing data. Our goal was to discover from the human generated abstracts, a replicable procedure for identifying the most informative concepts across a set of documents for the purpose of improving our system performance.

Our first investigation attempted to locate the source/s of the information in a human abstract from the multi-document collection. Using the SRA TagTool, an experienced analyst mapped the individual EDUs of a 100 word multi-document human abstract to the relevant document sentences in the multi-document collection for 23 different DUC training sets. Each EDU in an abstract is numbered, enabling us to correlate individual EDUs to the corresponding relevant document sentence/s throughout the collection. For each collection, we created a reference table that identifies the source/s of abstract information by gauging:

- Popularity within a document–the frequency with which an abstract EDU occurred in any given document
- Breadth across the collection—the number of documents within the collection in which the EDU source occurred

Table 4 shows EDU distribution for data set d25. We see there that abstract writers heavily utilized some documents and gave others little attention. Also, EDU sources can occur throughout the document body. Some EDUs are source-rich (EDU-10 in the Table) and can be traced to multiple documents. Others are source-poor (EDUs-1, -2, and -8), and arise from a single reference sentence within the collection. Some EDUs (although none in this example) have no traceable text source within the document collection.

There is no predictable abstract template or structure in evidence across the document collections. Informative abstracts are generated by synthesizing information from anywhere in the documents, often using background knowledge or inferences. The writers appear to adopt a focus that is often distinct from the focus of the document authors, and then create the abstract with that focus. Because they are generated, ideas can be streamlined.

As mentioned earlier, our system is effective in generating indicative single document summaries. However, our system is less effective in generating informative summaries, especially multi-document summaries, even after training on informative summaries. This is because informative summaries require a new feature set and we have not yet identified that set.

For a 100-word multi-document summary, our system tends to focus on a small number of the documents in the collection, routinely extracting 2–3 initial document sentences from only 2–3 of the documents. Rather than adopt a focus, we inherit multiple summary focuses that characterize the leads of the individual documents. Thus, our abstract EDUs tend to be only source-rich for one or two collection documents. For example, looking at the EDUs for the d25 multi-document machine-generated summary, only two of the documents contain multiple references to these summary EDUs and only two other documents reference even one of the twelve EDUs. The remaining documents in the set are not represented at all in the summary EDUs. Our approach currently restricts us to whole sentence extraction and thus includes a lot of information baggage, averaging 2–3 sentences per generated abstract. We have no chance of extracting source-poor information because their document information sources do not tend to occur in the initial sentences of the text and, perhaps more importantly, there are no observable text clues as to why one fact rather than any other was selected by the abstract writer.

Our second investigation attempted to understand what role, if any, single document summaries play when a human creates a multi-document summary. Here, an analyst mapped EDUs in the human-generated 100-word summary to EDUs in the single document summaries for five DUC 01 training sets—d02, d16, d17, d25, and d35. In most cases, the multi-document abstract could have been generated from the single document summaries and the multi-document EDUs are often source rich with references to more than one document. However, there are cases where the multi-document EDUs are not overtly referenced in the single document summaries at all. In these, the abstract writers rely on world knowledge, use of information in a document that is not in the single document abstract, or inferencing.

Utilizing the single document summaries may be a replicable strategy employed by the abstract writers. Our system uses this approach. To examine its efficacy, an analyst mapped from EDUs in the 100-word multi-document human summary to EDUs in the machine generated single document summaries for the same five DUC 01 training sets as above. More than half of the human abstract content was missing from the machine single document summaries and thus the quality of these single document summaries negatively affects coverage for the system multi-document summaries.

Our third investigation was to identify the impact of increased length on abstract content. We again analyzed the same five DUC 01 training sets and

Document	EDU-1	EDU-2	EDU-3	EDU-4	EDU-5	EDU-6	EDU-7	EDU-8	EDU-9	EDU-10	EDU-11	EDU-12
LA053089-0081	0	0	0	0	0	1	2	2	6	5	0	0
LA04i290-0125	0	0	0	0	0	0	0	0	0	0	0	0
LA092290-0175	0	0	0	0	0	0	0	0	0	0	1	0
LA111989-0125	1	1	2	2	3	0	0	0	2	2	1	5
LA112389-0104	0	0	0	0	1	0	0	0	2	2	0	0
LA113089-0118	0	0	0	0	2	0	0	0	0	0	0	0
LA121590-0056	0	0	0	0	0	1	1	0	3	2	0	0
SJMN91-06340029	0	0	0	0	0	0	0	0	0	0	1	1
WSJ900420-0022	0	0	0	0	0	0	0	0	0	0	0	2
WSJ911213-0029	0	0	0	0	0	0	0	0	0	0	2	0

Table 4: EDU Distribution for the d25 Data Set

compared coverage for three pairings of abstracts—50-word compared to 100-word, 100-word compared to 200-word, and 200-word compared to 400-word—for the 5 sets. Specifically, an experienced analyst judged whether the EDUs in the shorter abstract were contained in the longer abstract, for each pairing. We discovered that for some collections, even the abstract writers were "losing" data that we assumed should be subsumed into the longer abstract. Table 5 shows that for data set d16, 5 of 6 EDUs in the 50-word abstract are not subsumed in the 100-word abstract.

Document Collection	50-word to 100-word	100-word to 200-word	200-word to 400-word
d02	1/4	1/8	4/22
d16	5/6	5/11	2/25
d17	0/0	0/0	2/27
d25	1/5	3/12	2/24
d35	0/0	0/0	0/0

Table 5: EDUs Not Subsumed/Number of EDUs in the Smaller Abstract

Our system consistently subsumes information as the abstract length increases. High scoring sentences are retained as the summaries expand, though sentence order may change. The weights generated by our scoring technique impose a uniformity that does not necessarily characterize the human summaries.

We next turned to comparing the human generated abstracts to one another. For each of 15 DUC test sets, EDUs in each abstract of the set were compared to the other two abstracts in the set and given a score of the number of EDUs in the abstract that were unique, i.e., not contained in either of the other two abstracts in the set, divided by the total number of EDUs in the abstract. This gave us 45 scores. The median value of this score was 0.60—for a typical abstract, 60% of the EDUs are not be found in either of the other two abstracts for the same document set. I.e., assuming the EDUs are indicative of content, the majority of content in one abstract is distinct from the content in the other two abstracts. This suggests that the content is highly influenced by the abstract writer, and that one multi-document abstract may not provide a representative description of the original document set. This quantitative data underscores our contention that the abstract writers were adopting a focus that influenced the selection of content.

We assumed that this focus might also be evident in the discourse structure of the abstracts. One of the original annotators of the RST-Corpus ([CMO02]) applied the RST framework and created discourse trees for both 50-word and 100-word abstracts for 5 DUC test sets (30 total). Analysis of the set of discourse trees for the different abstract writers for the five DUC 01 test sets highlighted the significant differences in content coverage. Further details of this analysis will be presented in a future paper.

Our analysis of EDU distribution, use of single document summaries, abstract lengthening, and the rhetorical structure of abstracts have revealed a number of ways to enhance our system performance. Our team plans to

- enhance the informativeness of single document summaries by identifying and extracting new features
- increase collection coverage by applying individual sentence pruning techniques, along the lines of our existing heuristic, to remove discourse markers
- identify technically feasible solutions (e.g., creation of multi-document headlines) to simulate discourse structure

These enhancements also need to be coupled with the development and implementation of techniques for the abstract writers that reduce the production of idiosyncratic abstracts.

References

[AOGL97] Aone, C., M.E. Okurowski, J. Gorlinsky, and B. Larsen. "A Scalable Summarization System Using Robust NLP". *Proceeding of the*

ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization, pages 66–73, 1997.

[BPS70] Baum, L.E., T. Petrie, G. Soules, and N. Weiss. "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains". *Ann. Math. Stat.*, 41:164–171, 1970.

[CMO02] Carlson, L., D. Marcu, and M.E. Okurowski. "Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory". Forthcoming chapter in *Discourse and Dialogues*, Kluwer Academic Press, 2002.

[CO 01] Conroy, J.M. and D.P. O'Leary. "Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition. Technical Report, University of Maryland, College Park, Maryland, March, 2001.

[CSO01] Conroy, J.M., J.D. Schlesinger, D.P. O'Leary, and M.E. Okurowski. "Using HMM and Logistic Regression to Generate Extract Summaries for DUC". *DUC 01 Conference Proceedings*, 2001.

[Dun93] Dunning, T. "Accurate Methods for Statistics of Surprise and Coincidence", *Computational Linguistics*, 19:61-74, 1993.

[GKS76] Golub, G.H., V. Klema, and G.W. Stewart. "Rank Degeneracy and Least Squares Problems". Technical Report No. TR-456, Department of Computer Science, University of Maryland, 1976.

[KPC95] J. Kupiec, J. Pedersen, and F. Chen. "A Trainable Document Summarizer". *Proceedings of the 18th Annual International SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73, 1995.

[LH 00] Lin, C.-Y. and E. Hovy. "The Automatic Acquisition of Topic Signatures for Text Summarization". *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, 2000.

[Rab89] L.R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *Proceedings of the IEEE*, 77:257–285, 1989.

[SBD01] Schlesinger, J.D., D.J. Baker, and R.L. Donaway. "Using Document Features and Statistical Modeling to Improve Query-Based Summarization". *DUC 01 Conference Proceedings*, 2001.

[vH 02] van Halteren, H., "Writing Style Recognition and Sentence Extraction", *DUC '02 Conference Proceedings*, 2002.