# Using HMM and Logistic Regression to Generate Extract Summaries for DUC

John M. Conroy, Judith D. Schlesinger
Center for Computing Sciences
Institute for Defense Analyses
Bowie, MD 20715
{conroy, judith}@super.org

Dianne P. O'Leary*
Computer Science Dept. and UMIACS
University of Maryland
College Park, MD 20742
oleary@cs.umd.edu

Mary Ellen Okurowski
Department of Defense
Fort George G. Meade, MD 20755-6142 meokuro@afterlife.ncsc.mil

November 6, 2001

**Abstract**

*We present a discussion of our extract-based summarization algorithms and how we used them to produce summaries for the DUC test data. For single document summaries, we used two different algorithms, one utilizing a logistic regression model to choose the "best" sentences of the document for the extract and the second based on a Hidden Markov model that judges the likelihood that each sentence should be contained in the summary. For multi-document summaries, we combined two different approaches with the HMM. We compare the results of these methods with summaries generated on both the DUC training and testing data sets.*

Keywords: text summarization, extract summaries, hidden Markov models, logistic regression, automatic summarization, document summarization.

## 1 Introduction

Efforts to improve both generic and query-based extract-based summaries produced by an operational system resulted in two new algorithms for summarization. The first uses logistic regression (LRM); the second a Hidden Markov model (HMM). Both algorithms generate summaries that are at least 50% better (using f-score verification) then those produced by the original system, which was based upon [1].

The DUC evaluation presented an excellent opportunity for comparing the summaries our algorithms generate with those produced by summarization systems other than our own. There were, however, various issues that we needed to address. First, the DUC guidelines specified that the summaries were to be informative. We developed our algorithms to produce indicative summaries, per our users' preference. The informative abstracts provided with the training data had to be used to train and evaluate both algorithms in order to produce summaries that were appropriate for DUC submission (see Section 3 for details).

Second, the training data consisted of the document sets along with *abstracts* for the data. Our training and evaluation both relied on having one or more *extracts* for the training data. Section 3 also discusses how we handled this.

Third, the DUC data did not contain query terms or anything that could be used as query terms. We needed to supply query terms for both the LRM and the HMM without violating the "no manual modification or augmentation" DUC

*The work of DPO was supported by NSF Grant CCR 97-32022.

guideline. The LRM was developed assuming the existence of query terms and is not effective without them. While the HMM can be used without query terms, we discovered that summaries are significantly improved when query terms are used. Section 4 describes how we accomplished this.

Fourth, prior to receiving the training data, we had only given some initial thought to how to do multi-document summarization algorithms. The original summarization system did not do multi-document summarization and, while this was a planned extension, only general algorithm ideas had been proposed and no code had yet been developed. Section 2.3 describes the algorithms we developed to generate multi-document summaries.

Last, since we had two algorithms, we had to decide which to use for the actual DUC submission. See Section 5 for this discussion.

This paper presents a brief account of our algorithms along with a more detailed account of how we prepared and produced the submitted summaries.

## 2 The Algorithms

### 2.1 Single Document—Hidden Markov Model

In contrast to a naive Bayesian approach ([4], [1]), an HMM has fewer assumptions of independence. In particular, it does not assume that the probability that sentence $i$ is in the summary is independent of whether sentence $i-1$ is in the summary. Furthermore, we use a joint distribution for the features set, unlike the independence-of-features assumption used by naive Bayesian methods.

We consider three features in the development of a Hidden Markov model for text summarization.

- position of the sentence in the document—built into the state-structure of the HMM.
- number of tokens (non-stop words) in the sentence—value is $o_1(i) = \log(number\_of\_tokens + 1)$.
- number of "pseudo-query" terms (defined in Section 4) in a sentence: $o_2(i) = \log(Pr(\log(number\_of\_psuedo - query\_terms + 1)))$.

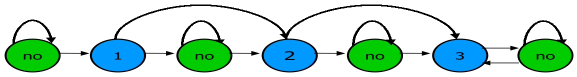


Figure 1: Summary Extraction Markov Model to Extract 2 Lead Sentences and Additional Supporting Sentences

We expect that 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. A first order Markov model allows such differences with marginal additional cost over a simple Bayesian classifier.

An HMM handles the positional dependence, dependence of features, and Markovity. (For more details about HMMs the reader should see [2] and [5].) 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 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 [3] for details.

### 2.2 Single Document—Logistic Regression Model

We first modeled our training data using a simple linear regression to identify predictive and non-predictive features. Our initial list of 14 features (including some, such as discourse markers and collocation information that were eliminated due to their complexity) was narrowed to the four predictive features we included in the model:

- (V1) the number of unique query terms in a sentence—the greater the better.
- (V2) the number of tokens (non-stop words) in a sentence—the greater the better.
- (V3) the distance of a sentence from one with a query term—a distance of 0, i.e., the sentence contains 1 or more query terms, is best.
- (V4) the position of a sentence in the document—there is a bias to "early" sentences[1]

When sentences that the model scored above 1 were labeled "extracted", we found that roughly 50% of the sentences were misclassified, which was not acceptable. To achieve better results, we simplified the problem and switched to a logistic regression.

A logistic regression model is designed for a binary response variable. Unlike ordinary linear regression, logistic regression constrains the fitted values to lie between 0 and 1. We transformed our features V1, V2, and V4 by dividing by the maximum discovered on a per-document basis. V3 was transformed using $\log(1+V3)$. The logarithm captures the intuitive feeling that a sentence that is just one or two away from one containing a query term is likely to be more "important" than one that is 5 or especially 10 away.

The logistic regression formula is as follows:

$$score = f(\alpha+\beta_1(V1)+\beta_2(V2)+\beta_3(V3)+\beta_4(V4))$$

where $f(x) = \frac{e^x}{1+e^x}$, the V$i$s are the transformed values as described above, and $\alpha$ and $\beta_i$ are coefficients defined from training. See [6] for a more detailed explanation of this algorithm.

### 2.3 Multi-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 by choosing all sentences with scores exceeding a fixed threshold value $\tau$ (set empirically to 0.65). This selects far more sentences than are included in single document summaries.

The candidate sentences are then used to form a token-sentence matrix, which we call $A$. We wish to choose columns from $A$ which give good coverage of the tokens. We considered two approaches to solving this problem: the pivoted QR factorization and a greedy selection algorithm.

[1] Note that this bias, while sometimes valid, is often a result of a natural human inclination to stop the activity once a reasonable set of summary sentences has been tagged even if the "best" sentence(s) have not yet been found. Evaluation methods that go solely by sentence number (which we are using) miss "good" selected sentences, that have almost identical meaning as some earlier tagged sentences or are different but equally as good choices.

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 [3] for details.

The second method we used was a variation on a greedy method to produce a covering of the tokens covered by $A$. The first column selected is the same chosen by the pivoted QR algorithm. Once a sentence is selected, all of its tokens are considered covered. The next sentence selected is the one with the largest ratio of uncovered tokens to tokens within that sentence.

## 3 Training

Our work was set up to compare generated summaries to human tagged extracts. Since the training data only had abstracts, not extracts, we had a problem. Our first attempt to remedy this was to use the cosine score to map the abstract into sentences of the documents and then treat those sentences as the extract. While this gave us a metric with which to compare our summaries, human evaluation of the contrived extracts determined that the sentences chosen were simultaneously over-generated and under-generated. Simply put, we had too much of the same information and too much missing information.

As an alternative, we had analysts map the extracts 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 the attempt at automatically generating the mappings. The analysts were able to easily handle extracts containing a synthesis of information, which were especially difficult for the automatic process.

The analysts' extracts solved two problems for us. First, we now had the tagged sentences we needed for training and evaluation purposes. Second, the analysts chose sentences to match, as closely as possible, the informative nature of the abstracts. This demonstrated that we can easily switch between informative and indicative summaries, based on the training data used.

The HMM was trained and tested using the 148 documents which were tagged by the analysts. These extracts were generally longer than the required 100 word count, since the original abstracts often drew information from multiple sentences. As such, two measurements were made: the *precision* of a 100-word computer-generated extract relative to the longer length human generated extract and the *precision* of a computer-generated extract with the same number of sentences as the human extract relative to the human extract. A ten-fold cross validation was done using the data. The precision for the 100 word extracts was 0.52 and the precision for the longer extracts was 0.44.

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 greedy covering algorithm submitted to DUC scored a 0.35 cosine score for 400 word summaries; however, our submission to DUC contained a couple of severe errors. With the errors corrected, the average cosine scores for the training data were 0.47 and 0.43 for the pivoted QR and greedy covering methods, respectively. In the results section, we illustrate how the corrected methods would have performed.

For a variety of reasons, we never trained the LRM. Instead, we ran the training data using coefficients that we had calculated from other training data. These coefficients came from two sources. The first was training data from a proprietary data base which, while consisting of a variety of news items, is substantially different from the DUC data. The second was training data from the TREC data base[2].

For reasons not yet understood, the TREC-trained model did not do as well as the proprietary data-trained model, with precisions of 0.36 and 0.44, respectively, for summaries the same length as the human-generated extracts. For 100-word extracts, the precisions are 0.42 and 0.5, respectively.

[2]TREC data sets 110, 127, and 132 to be exact.

We had our analysts evaluate the summaries generated by both algorithms and they found the two to be different, yet equally as good (and bad), which supported the similar precision scores.

## 4 Identifying Query Terms

The last issue we need to discuss is that of query terms. As mentioned earlier, it was essential for the LRM to have these terms and very useful for the HMM. Generally, our users know what they are looking for and supply the query terms for their summaries. Having humans look at the data sets and determine a set of query terms violated the spirit of the competition. We needed an automatic way to generate "pseudo-query" terms.

To do this, we created a token list, with token usage counts, for both the individual document sets as well as the entire document collection. The individual document set token counts were then normalized using the following z-score:

$$\sigma_k(i) = \frac{s_k(i) - S_k p(i)}{\sqrt{S_k p(i)(1 - p(i))}}$$

where $p(i) = \frac{c(i)}{\sum_j c(j)}$, $c(i)$ = the number of times the token occurs in the collection, $sum_j c(j)$ = the number of tokens in the collection, $s_k(i)$ = the number of times the token $i$ occurs in document set $k$, $S_k = \sum_j s_k(j)$ is the number of tokens in document set $k$, $k$ ranges over the number of document sets, and both $j$ and $i$ range over the tokens. Note that $p(i)$ represents the probability that token $i$ occurs at-large. If we assume tokens are binomially distributed, then $\sigma_k(i)$ is the number of standard deviations (sigmage) above or below the expected frequency based on the at-large distribution of tokens.

For each document set, we then chose all the tokens which had a normalized count that was greater than or equal to 10 (which was an empirically set threshold). These chosen tokens became the pseudo-query terms for each document set and were used by both algorithms.

## 5 Generating the Final Summaries

The HMM and LRM were 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. Selected sentences were then reordered in their original document order to create the final summary.

We planned to submit a full set of single document summaries from each algorithm. However, with the change in conference rules mandating a single submission per site, we decided to evaluate our output to select which summary set to use. The results of this evaluation showed that the summaries generated by the two algorithms were different but neither could be declared superior to the other.

Given that both algorithms had strengths and weaknesses, we decided to create a single submission by including summaries generated by both algorithms in order to garner the feedback on quality that the DUC multiple review strategy would provide. Our analysts classified the thirty document sets based on the DUC descriptions. For each type of document (single main event, biographical, etc.), we pseudo-randomly assigned document sets, as evenly as possible, to each algorithm. The final document set split is:

- HMM: d06, d08, d12, d13, d14, d19, d24, d32, d34, d37, d39, d44, d50, d53, d59

- LRM: d04, d05, d11, d15, d22, d27, d28, d30, d31, d41, d43, d45, d54, d56, d57

Similarly, after human review of the generated multi-document summaries, we used the greedy covering method for the 50, 100, and 200 word summaries and the QR with partial pivoting method for the 400 word summaries.

## 6 Results

The DUC evaluation gives a wealth of information. To understand the effectiveness of our summaries, we focus on six metrics provided by DUC. A score from 0 to 4 was given for each metric, where all = 4, most = 3, some = 2, hardly any = 1, none = 0. Our six metrics are:

- # overall peer grammaticality
- # overall peer cohesion
- # overall peer organization
- # unmarked PUs needed in model
- # unmarked PUs only related
- # unmarked PUs unrelated

where high scores are good for the first 5 while a low score is good for the sixth.

We compared the single document results with those of the other participants. To do this, we computed an average of each of the six metrics for all other participants and selected the individual high and low score for each metric. These crude "yardsticks" were then compared with our submissions. Since both the LRM and HMM were submitted for the single document evaluation, we further subdivided the metrics by LRM and HMM. Table 1 shows the metrics for the single document summaries.

Examples of the best-scoring and worst-scoring summaries generated by both the LRM and the HMM are shown in Figures 2, 3, 4, 5. Both the LRM and the HMM had scores of 4.0, 4.0, 4.0, 0.0, 4.0, 0.0 for their best summaries, all perfect since the number of unmarked PUs unrelated has a perfect score of 0.0.

The scores for the worst summaries differed considerably: 3.0, 0.0, 0.0, 1.0, 1.0, 2.0 for the LRM and 1.0, 2.0, 2.0, 0.0, 0.0, 4.0 for the HMM and demonstrate that both approaches can miss the mark. Quality suffered, with non-relevant information included and relevant information omitted. At least for these examples, the HMM seems to be better at cohesion and organization while the LRM is better in terms of grammaticality and PU selection/omission. Note that in both of these summaries, the system includes copy service information.

GOVERNMENT veterinary and health experts were yesterday putting out reassuring messages about bovine spngiform encephalopathy (BSE), or 'mad cow' disease, in the face of growing public anxiety. Dr Kenneth Calman, the government's chief medical officer, yesterday repeated the official advice that beef can be eaten safely: 'There is no scientific evidence of a causal link between BSE in cattle and CJD in humans.' One cause of concern is that the number of cases is continuing to rise, in spite of forecasts from the Ministry of Agriculture that the incidence of cases would peak last year and then decline rapidly.

Figure 2: Logistic Regression Best Summary—Document d05a/FT931-3883

Eds: SUBS 5th graf, 'Gov. Guy...' with 3 grafs to UPDATE with 25-mile path, tornado watches in Northeast. Rescuers crawled through collapsed homes and shops today looking for more victims of a tornado that carved a 3-mile stretch of destruction, killing 17 people, injuring 463 and leaving 1,000 homeless. In Huntsville, teams with cranes and floodlights searched for the injured or dead, hampered by wind-whipped rain and temperatures that plummeted overnight from 73 degrees into the 30s. In West Virginia, high winds believed to be tornadoes swept Jefferson County early today, overturning trailers, blowing roofs off homes and downing power lines, authorities said.

Figure 3: Logistic Regression Worst Summary—Document d11b/AP891116-0115

For the multi-document summaries, we again compared our entries in the same six categories against the average other DUC entry. The results are given in Table 2.

Examples of our best-scoring and worst-scoring multi-document summaries are shown in Figures 6 and 7, respectively. For the best summary, the scores are 4.0, 4.0, 4.0, 0.0, 4.0, 0.0, identical to the best by each of the single document algorithms. While our greedy algorithm chose a good sentence, we are still missing content. This indicates that an extract approach can capture information synthesized by a human in an abstract.

For the worst summary, the scores are 0.0, 0.0, 0.0, 0.0, 0.0, 4.0. We can't do any worse than that. If you look at the summary, it is clear that regardless of whether or not the one sentence chosen is appropriate, there is not nearly enough

Two days of racially charged hearings on police brutality and a report detailing widespread segregation in the nation's third-largest city show the new mayor must still heal some old wounds. Richard M. Daley was elected mayor April 4 amid fears by black activists that he would bring back the machine politics of his late father, Richard J. Daley, who was mayor for more than 20 years before his death in 1976. The younger Daley emphasized empowerment of minorities in his spring campaign, and after defeating black challengers in the primary and general election, he named minorities to 11 of his 21 Cabinet positions.

Figure 4: Hidden Markov Best Summary—Document d06a/AP890930-0100

Metro; Part B; Page 2; Column 1; Metro Desk METRO DIGEST / LOCAL NEWS IN BRIEF: ELIZABETH TAYLOR'S DOCTORS WILL NOT FACE CHARGES The Los Angeles County district attorney's office declined Friday to press charges against several physicians, ending its investigation into allegations that they overprescribed painkillers to actress Elizabeth Taylor. In a written report, the district attorney's office said the prescirbing practices "fell below the accepted standard of medical practice," but added that the doctors "were also attempting to deal with her addiction through alternative means of therapy and treatment, and . . . their conduct was devoid of criminal intent."

Figure 5: Hidden Markov Worst Summary—Document d24d/LA042190-0060

An amendment introduced by conservative lawmakers in the House and Senate, and tagged onto a HUD bill later signed by the President, states that no Community Planning and Development grants may be paid "to any municipality that fails to adopt and enforce a policy prohibiting the use of excessive force by law enforcement agencies within the jurisdiction of said municipality against any individuals engaged in nonviolent civil rights demonstrations."

Figure 6: The Best Summary—50-words for Document Set d06

content.

Overall, our peformance in multi-document summarization fell short, especially in comparison to our work in single document summarization. Our analysis of the multi-document output, including the disastrous example shown in Figure 7, led us to discover the two bugs alluded to in Section 3. In particular, an indexing error in the covering algorithm was to blame for a problem with summary lengths. This problem affected our entries for 50-, 100-, and 200-word summaries.

More importantly, the HMM scores used to select which sentences to choose for multi-document summaries were sorted in the *wrong order*, i.e., the worst sentences found by the HMM were passed to the covering and QR methods for sentence selection. This bug affected *all* of our multi-document entries, including the 400-word summaries generated with the QR algorithm.

The compound effect of these two errors is that the summaries generated using the covering algorithm were really nothing more than pseudorandom selections.

The corrected 200-word summaries for data set d30 are given for the covering and QR methods in Figures 8 and 9. Of these two improved summaries, the QR is superior, which is consistent with the higher cosine score on the training data as reported in Section 3. For contrast, the human-generated 200-word *abstract* used to score our submissions is shown in Figure 10.

**Resulting Common Equity/Asset ratio reflecting the additional prvision.

Figure 7: The Worst Summary—200-words for Document Set d30

He had decided he could not comply with requirements of a consulting job he had accepted with the Agency for International Development, and he was scrambling to come up with a suitable substitute. The bulk of African debt is owed to officil lenders under various aid agreements. The debts represent loans with a substantial grant element. The debts of African countries have often been cancelled or rescheduled, frequently several times for the same country. Mr. Lewis Preston, World Bank President, yesterday promised to strengthen the bank's efforts to reduce poverty in developing countries. 'We will look for specific increases in the share of lending going for these purposes.' Internationally, it appears that it will be even more difficult for economically troubled developing nations to attract new bank loans. But at the World Bank, Mr. Conable finds himself under fire. The bank has tried to help the countries by tiding them over with some new loans.

Figure 8: Correct Covering Method—200-words for Document Set d30

Foreing Minister Roberto de Abreau Sobre of Brazil told the opening session of the 42nd General Assembly that the Third Wolrd economic picture was dimming "due to the lack of progress in international economic relations." "It is ... sad to note that we, American, Asian, African brothers, still suffer from the same horrors and the same desolation which so badly affected our forebears," he said, adding, "hunger is endemically spreading throughout the continents." What he foresees is a body blow to world poverty through bootstrap economics. The link between loans and poverty alleviatin the bank's central mission in the 1990s. The shift in priorities is also reflected in a commitment to make comprehensive assessments of the nature and extent of poverty in the third world, allowing the bank to design more effective policies to fight poverty. The figure, contained in the bank's annual report and made public before its annual meeting here Sept. 23, was almost a third larger than in 1987, when the net pay-back totaled $38.3 billion. Domestically, the move reflects the competitive advantage that regional banks with large loan-loss reserves have over their big brothers in such money centers as New York, Chicago and San Francisco. World Bank President Barber Conable was so well regarded during his 20-year career as a Reppublican congressman from New York that some journalists nicknamed him "H.R." – for "highly respected."

Figure 9: Correct QR Method—200-words for Document Set d30

The programs of the World Band have yet to meet with success in reducing Third World debt. There are a number of ideas on how this should be accomplished. One is to invest in the private sector and bypass the governments of developing countries. Another is to lend new money only and not reduce any debt. Yet another is to write-off Third World debt as uncollectible. A persistant problem in the loaning of money by the World Bank to Third World governments is that they are often swamped by interest payments. Brazil, the largest debtor nation, blames the industriablized countries for its poverty problems. It is difficult to obtain a definitive statement from officials of the World Bank. Specific targets and goals toward the alleviation of poverty and the crushing debt are avoided. Some believe that the bank's policies are correct and that poverty in the Third World is caused by their own governments. Recent individual efforts to relieve Third World poverty and debt may be successful. "Village Banking" is a means of investment using loans to poor women at low interest rates to finance their private enterprise. Similar arrangements are being made by third world banks and development organizations.

Figure 10: 200-word Human-Generated Extract for Document Set d30

## 7 Conclusions

Our Hidden Markov and logistic regression algorithms produce single document extract summaries that are generally grammatical, cohesive, and organized. Machine performance does not, of course, exceed the quality of the human generated model but tends to capture related information and generally avoids extracting too much. We expect to improve the quality of these single document summaries as we continue to refine our algorithms.

Our multi-document summarization algorithms, in the early stages of development. After the DUC evaluation, we found two bugs in our implementation which caused our 400 word submissions to be based on the *worst* HMM scoring sentences, rather than on the best. The 50-, 100-, and 200-word summaries were additionally plagued with an indexing error which rendered them pseudo-random selections. Despite these major fiascoes, the multi-document results were better than average in the "PUs Needed" and "PUs Related" scores. Using our own evaluation of the debugged multi-document algorithms, we produced summaries with extract to abstract cosine similarity scores 34% higher than the ones we submitted to DUC.

We plan to retrain both methods with more effective training data in which the annotator more closely models how the abstracts were generated. We hope to see dramatic improvement from this retraining.

**Acknowledgements**

We want to thank our colleagues Steve Kratzer, Tony Taylor and Frank Krapcho. Steve contributed substantially to our algorithms for multi-document summarization. Tony did all the pre-processing necessary to transform the DUC raw data into the internal format we needed for processing and produced a script to generate the output in the DUC-format. Frank assisted in evaluating the quality of the generated summaries and classifying the document sets based on the DUC descriptions. Our task would have been far more difficult without their assistance.

## References

[1] C. Aone, M. 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.

[2] L. E. Baum, 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.

[3] J. M. Conroy 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.

[4] 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.

[5] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77:257–285, 1989.

[6] J. D. Schlesinger, D.J. Baker, and R.L. Donaway. Using Document Features and Statistical Modeling to Improve Query-Based Summarization. Proc. of the ACM SIGIR'01 Workshop on Text Summarization, 2001.

| Method | Grammar | Cohesion | Organization | PUs Needed | PUs Related | PUs Unrelated |
|---|---|---|---|---|---|---|
| HMM+LRM | 3.6 | 2.7 | 2.8 | 0.3 | 2.3 | 0.26 |
| HMM | 3.6 | 3.1 | 3.2 | 0.27 | 2.3 | 0.28 |
| LRM | 3.6 | 2.3 | 2.5 | 0.33 | 2.3 | 0.23 |
| Avg Other | 3.6 | 2.7 | 2.9 | 0.35 | 2.6 | 0.34 |
| High Other | 4.0 | 3.2 | 3.6 | 1.2 | 3.2 | 0.7 |
| Low Other | 3.2 | 2.5 | 2.6 | 0.2 | 1.7 | 0.0 |

Table 1: Metrics for Single Document Summaries

| Length | Grammar | Cohesion | Organization | PUs Needed | PUs Related | PUs Unrelated |
|---|---|---|---|---|---|---|
| Our 50 | 3.0 | 1.3 | 1.2 | 0.0 | 2.0 | 1.4 |
| Avg Other 50 | 3.4 | 2.2 | 2.6 | 0.2 | 2.8 | 0.4 |
| High Other 50 | 4.0 | 3.7 | 3.7 | 1.0 | 4.0 | 1.4 |
| Low Other 50 | 2.3 | 1.3 | 1.2 | 0.0 | 1.2 | 0.0 |
| Our 100 | 3.2 | 1.3 | 1.1 | 0.1 | 2.4 | 1.4 |
| Avg Other 100 | 3.5 | 2.0 | 2.3 | 0.3 | 3.1 | 0.4 |
| High Other 100 | 4.0 | 3.6 | 4.0 | 1.2 | 3.5 | 1.4 |
| Low Other 100 | 2.3 | 1.3 | 1.1 | 0.0 | 1.0 | 0.0 |
| Our 200 | 3.3 | 1.7 | 1.4 | 0.1 | 2.8 | 1.2 |
| Avg Other 200 | 3.5 | 2.1 | 2.1 | 0.3 | 3.2 | 0.4 |
| High Other 200 | 4.0 | 3.3 | 3.7 | 1.4 | 3.8 | 1.3 |
| Low Other 200 | 2.4 | 1.2 | 1.0 | 0.0 | 2.6 | 0.0 |
| Our 400 | 3.0 | 1.6 | 1.4 | 0.2 | 2.8 | 1.0 |
| Avg Other 400 | 3.4 | 2.0 | 2.1 | 0.4 | 3.3 | 0.4 |
| High Other 400 | 4.0 | 3.2 | 4.0 | 1.7 | 4.0 | 0.0 |
| Low Other 400 | 2.7 | 1.2 | 1.0 | 0.0 | 2.3 | 0.0 |

Table 2: Metrics for Multi-Document Summaries