Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition

John Conroy
Center for Computing Sciences
Institute for Defense Analyses
17100 Science Drive
Bowie, MD 20715
conroy@super.org

Dianne P. O'Leary
Computer Science Dept. and
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742 USA
oleary@cs.umd.edu

February 22, 2001

The work of the second author was supported by NSF Grant CCR 97-32022.

Abstract

A sentence extract summary of a document is a subset of the document's sentences that contains the main ideas in the document. We present two approaches to generating such summaries. The first uses a pivoted QR decomposition of the term-sentence matrix in order to identify sentences that have ideas that are distinct from those in other sentences. The second is based on a hidden Markov model that judges the likelihood that each sentence should be contained in the summary. We compare the results of these methods with summaries generated by humans, showing that we obtain higher agreement than do earlier methods.

CR category: H.3.3, G.1.3
Keywords: text summarization, extract summaries, hidden Markov models, automatic summarization, document summarization.

1 Introduction

In this paper we present two new methods for automatic text summarization. Automatic summarization has been studied for over 40 years [Luh58], but with pervasive use of information retrieval systems in the last 6 years this area has been given wider attention [HM00]. For example, Microsoft Word has a built-in tool to summarize documents created by it. Such a summary is an example of a *generic summary*, i.e., one that attempts to capture the essential points of a document. In addition, summarization methods are regularly used by web search engines to give a brief synopsis of the documents retrieved by a user query. Such *query-based summaries* can be more focused, since the user's query terms are known to the retrieval system and can be used to target the summaries to the query.

Summaries may also be categorized *indicative* or *informative*. An indicative summary is a summary that simply gives the main focus of a document and would be used to determine if the rest of the document should be read. Such a summary would be fairly short, perhaps as little as two sentences. In contrast, an informative summary is a summary that can be read in place of the document and its length is not restricted [HM00].

Here, we focus on generic summaries, although most of the ideas presented can be adapted to generate query-based summaries of text. We generate *sentence extract summaries*; i.e., the summary consists of a subset of the document's sentences. We will present results for both indicative and informative summaries.

The first method we present summarizes a document by seeking the main *ideas*. We follow [AOGL97] in the use of Natural Language Processing techniques to "go beyond the words" and instead focus on *terms*. Co-location is used to disambiguate the meaning of words that rely on context. We use the SRA's NameTagTM [Kru95] to recognize named entities and WordNet [MBF+90] to associate synonyms. Once the terms are defined a term-sentence matrix is formed. The sentences are viewed now as vectors analogous to Salton's vector space model for information retrieval [BYRN99]. The job of the automatic summarization system is to chose a small subset of these vectors to cover the main ideas (terms) in the document. The method we propose is taken from numerical linear algebra: the *QR decomposition with partial pivoting* [GL96, Sec 5.5.6]. Broadly speaking this approach iteratively chooses the vector (sentence) with the largest weight. Then the weights of the remaining vectors (sentences) are updated to emphasize ideas not contained in the chosen sentence. This selection criterion is related to the Maximum Marginal Residule method [CG98], but the latter method only considers pair-wise overlap of sentences, while the QR method considers overlap with the entire set of sentences chosen thus far. This method and its application will be described in Section 2.

The second method we propose for text summarization is a *Hidden Markov Model (HMM)*. Jing and McKeown [JK99] previously proposed a HMM for decomposing a human summary, i.e., mapping the component parts of a summary generated by a human back into the document. Here, we present the first HMM for use in summarizing text. Our HMM has two kinds of states, corresponding to summary sentences and non-summary sentences. The beginning of the document is handled by special states so that early document structure can be easily captured. This method is described in Section 3.

Both the HMM and QR decomposition produce generic summaries that achieve F_1 scores higher than scores previously reported, and we present sample results in Section 5. When combined these approaches capture the salient information in a document while eliminating redundancy.

2 QR Method of Text Summarization

Given a document, we want to choose a subset of sentences in that document in order to form a summary of it. A good summary contains a small number of sentences but captures most of the main ideas of the document.

The basic idea behind our first algorithm is simple. Given the sentences in the document and a measure of the importance of each, we choose the most important sentence to add to our summary. Once this is done, the relative importance of the remaining sentences changes, because some of them are now redundant. We repeat this process until we have captured enough of the important ideas.

Our task, therefore, is to define "ideas" and then to develop a measure for their importance, a means of updating these measures, and a criterion for determining when to stop. The tool that we use is an algorithm from computational linear algebra known as the *pivoted QR decomposition of a matrix.*

2.1 Ideas and Importance Measures

Exactly what is an "idea"? In our work we take a simplistic definition: idea = term. Parsing the document for terms is easily done, in our case, using the Brill tagger [Bri93]. A more sophisticated definition of "idea" would clearly lead to better results, but our algorithm is applicable regardless of how "idea" is defined.

An idea is important in a sentence if it appears in that sentence; otherwise its importance is zero. In our computations, nonzeros in A were taken to be ones, but other schemes are possible [BYRN99].

We collect the importance measures in a term-sentence matrix A, where each column corresponds to a different sentence. The entry in row i and column j is nonzero if the ith term appears in the jth sentence, and is equal to zero otherwise.

2.2 Applying the QR with Partial Pivoting to Summarization

To choose sentences, we will measure the "importance" of a sentence by the norm (Euclidean length) of its column. (Thus, sentences that have a large number of terms are initially rated as very important.) At each stage of the algorithm, the next column (i.e., sentence) that we include is the one with the largest norm. This choice is called *pivoting.*

Once a sentence is added to the summary, we want to update the importance measure for each remaining sentence: ideas that it shares with the added sentence are no longer important to capture, so the norm is reduced proportionally, by subtracting off the component of that column that lies in the direction of the column that we just added. This process of making the remaining matrix orthogonal to the previously chosen columns forms the *QR decomposition*.

In other words, any standard implementation of the pivoted QR decomposition gives us an ordering of columns that defines their priority for inclusion in the summary. The only remaining issues are how to define the nonzero entries in the matrix and how to determine the length of the summary.

2.3 Determining the Weighting Factors and Summary Lengths

Given the term-sentence matrix A, we want to weight sentences by position in the document: the nonzeros in column j of matrix A_w are $(g * exp(-8 * j/n) + t)$ times those in A, where n is the number of sentences in the document and g and t are parameters to be determined.

Using a training (development) set of documents, the parameter t is determined so that the function $(g * exp(-8 * j/n) + t)$ has a tail approximately the same height as the histogram (with 20 bins) of the distribution of summary sentences.

Given t, we determine g and the percent of the document to capture by maximizing F_1, the standard measure of a good summary compared to a human generated summary, over the training documents. F_1 is defined by

$$F_1 = \frac{2r}{k_h + k_m} \tag{1}$$

where k_h is the length of the human summary, k_m is the length of the machine generated summary, and r the number of sentences they share in common.

This can be done by using standard software for maximizing a function of a single variable. For any given value of g, we determine summary lengths by including enough sentences to reduce the norm of the remaining matrix to less than d% of the norm of the original, and choosing d so that the mean length of our summaries matches the mean length of the given summaries as closely as possible. (We obtained somewhat better results by doing this measurement on A rather than A_w.)

2.4 Implementation issues

The standard implementation of the pivoted QR decomposition is a "Gram-Schmidt" process implemented as follows.

Algorithm 2.1 (Pivoted QR Decomposition) *Suppose A_w has T columns and m rows: i.e., the document has T sentences and m terms. The following iteration constructs a matrix Q with columns q_i, a matrix R with nonzero elements r_{ji}, and an ordering for the columns in an array Index.*

For $i = 1, 2, \ldots, \min(m, T)$,

> *Choose column ℓ of A_w to be the column of maximum norm among all columns not jet chosen. Denote this column by a_ℓ.*
>
> *Set $Index_i = \ell$.*
>
> *Set $q_i = a_\ell / ||a_\ell||$.*
>
> *Update the other columns of A_w to make them orthogonal to the chosen column: for each unchosen column a_j, set $r_{ji} = a_j^T q_i$ and set $a_j = a_j - r_{ji} q_i$.*

The summary of length k contains sentences $Index_1, \ldots, Index_k$.

The work for this algorithm is proportional to $mT \min(m, T)$.

There are several ways to reduce the work.

First, we do not need the complete QR decomposition; we only need to determine the first k columns, where k is the length of our summary. This makes the work proportional to mTk.

Second, we do not need to update all of the columns at every step; we only need to update the column chosen at this step and norms of the other columns, so that we know which column to choose next. The change in norm for an unchosen column at step i is

$$||a_j||^2 = ||a_j||^2 - r_{ij}^2 \, ,$$

and by not updating a_j we avoid introducing new nonzero elements and thus keep the matrix sparse and cheap to store.

Other low-storage variants of the pivoted QR algorithm have been proposed by Stewart [Ste99].

3 Hidden Markov Models

In this section we describe an approach that given a set of features computes an a-posterior probability that each sentence is a summary sentence. In contrast to a naive Bayesian approach [KPC95] [AOGL97], the Hidden Markov model has fewer assumptions of independence. In particular, the HMM 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.

3.1 Features

We consider five features in the development of a Hidden Markov model for text summarization. Four features have been discussed previously in the literature, and the last is apparently new.

- position of the sentence in the document. This feature is built into the state-structure of the HMM and is discussed in the next section.

Feature Name	Label	Value
Paragraph Position	o_1	1,2,3
Number of Terms	o_2	$\log(w_i + 1)$
Baseline Term Probability	o_3	$\log(Pr(\text{terms in } i \vert \text{baseline}))$
Document Term Probability	o_4	$\log(Pr(\text{terms in } i \vert \text{document}))$

Table 1: Features used in the Markov model

- position of sentence within its paragraph. We assign each sentence a value $o_1(i)$ designating it as the first in a paragraph (value 1), the last in the paragraph (value 3), or an intermediate sentence (value 2) The sentence in a one-long paragraph is assigned the value 1, and sentences in a two-long paragraph are assigned values of 1 and 3.
- number of terms in the sentence. The value of this feature is

$$o_2(i) = \log(\text{number of terms} + 1).$$

- how likely the terms are, given a baseline of terms. Given the frequencies b_j equal to the number of times term j occurred in a collection $\mathcal{B}$ of "baseline" documents, we compute for each sentence i in a document D

$$o_3(i) = \log(Pr(\text{terms in sentence } i|\mathcal{B})) = \sum_{j \in i} \log(\frac{b_j}{\sum_{k \in D} b_k}). \tag{2}$$

  Our baseline document set was the same used by [AOGL97] and consisted of one million news articles.

- how likely sentence terms are, given the document terms.

$$o_4(i) = \log(Pr(\text{terms in sentence} i|D)) = \sum_{j \in i} \log(\frac{d_j}{\sum_{k \in D} d_k}). \tag{3}$$

  where d_j is the number of times term j occurs in document D.

The feature o_3 is a variant of the commonly used tf/idf [BYRN99]; we prefer this variant over others because of its natural probabilistic interpretation. In particular this feature gives the probability that the terms would occur in a "baseline" document. The features are summarized in Table 1.

3.2 The Markov Model

Given the set of features described in the previous section, one approach would be to use a naive Bayesian classifier [Kupiec] [AOGL97]. A limitation of such an approach is the assumption of independence, violated in our application, since several features may depend upon the sentence's position in the document. Clearly this is the case for paragraph position, but a bit more subtle is dependence of length of sentence and position. A second type of dependence is that among the features components o_k, for k=1,2,3, and 4.

A third dependence we wish to exploit is Markovity. 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.

A Hidden Markov Model can handle the positional dependence, dependence of features, and Markovity. We now present our HMM for text summarization. (For more details about HMMs the reader should see [BPSW70] [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 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. To see how it extracts the lead sentences note that every path through the chain visits each of first $s-1$ summary states. Note that the last two states in the chain allow for an arbitrary number of summary and non-summary sentences. This Markov chain has a total of $2s$ free parameters defining the probability of the various transitions between pairs of states. These parameters are estimated based on training data: for example, the probability of transition between summary state $2j$ and summary state $2j + 2$ is the number of times summary sentence $j + 1$ directly followed summary sentence j in the training documents, divided by the number of documents; and the probability of transition between summary state $2j$ and non-summary state $2j + 1$ is defined to be one minus this probability. Through these calculations, 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.

In a similar way, we compute p, the maximum likelihood estimate of the initial distribution for the chain, with

$$p(i) = \Pr(\text{The first sentence corresponds to state } i) \tag{4}$$

Note that $p(i) = 0$ for $i > 2$, since the first sentence is the first summary sentence (state 2) or a sentence that precedes the first summary sentence (state 1).

A slight modification of the chain allows for extraction of exactly s summary sentences. This chain given in Figure 2 differs from the chain of Figure 1 in eliminating the cycle between the last summary and non-summary states. This chain is most appropriate for generating a fixed length indicative summary. It has $2s$ free parameters to be estimated from training data.

Associated with each state i is an output function,

$$b_i(O) = Pr(O|\text{state } i),$$

where O is an observed vector of features (e.g., the 4 features of the previous section) belonging to a sentence. We make the simplifying assumption that

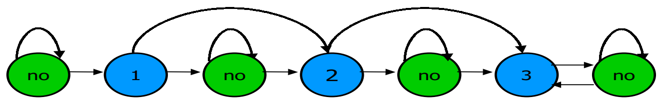

Figure 1: Summary Extraction Markov Model to Extract 2 Lead Sentences and Additional Supporting Sentences

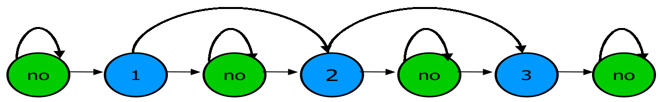

Figure 2: Summary Extraction Markov Model to Extract 3 Sentences

the features are multi-variant normal. This assumption keeps the number of parameters to be estimated at a minimum, while allowing for dependencies. If we use k features then there are $k + k(k+1)/2 = \frac{k^2}{2} + \frac{3k}{2}$ parameters for each multi-variant normal distribution and since we have one for each state the total number of parameters for the functions $b()$ is $2s(\frac{k^2}{2} + \frac{3k}{2})$ or $(2s+1)(\frac{k^2}{2} + \frac{3k}{2})$ depending on whether we use the first or second Markov chain.

For simplicity, in the remainder of the paper we will discuss only the first Markov chain.

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. Depending on the state, the mean vector predicts the expected value of each of the features when the sentence is a summary sentence, a non-summary before the first summary sentence, or a non-summary sentence occurring after some summary sentence. We estimate $2s + 1$ means, but assume that all of the output functions share a common covariance matrix.

Therefore, we estimate the mean for state $2j$ as the average of the output vectors for the jth summary sentence in each of the training documents. Similarly, we estimate the mean for state $2j+1$ as the average of the output vectors for all sentences between the jth and $(j+1)$st summary sentences in each of the training documents. The covariance estimate is the average of $(O-\mu)(O-\mu)^T$ over all sentences in the training set, where O is the 4×1 feature vector for the sentence and μ is the mean we computed for its corresponding state.

In summary, our model consists of three parts: p the initial state distribution, M the Markov transition matrix, and B the collection of multi-variant normal distributions associated with each state.

Let $\alpha_t(i)$ be the probability that we have observed the sequence $\{O_1, O_2, ..., O_t\}$ and are currently in state i $(1 \leq i \leq N)$ of our HMM. We can compute $\alpha_t(i)$ recursively as follows. Let $\alpha_1(i) = p(i)$ and compute

$$\alpha_t = D_{O_t} M^T \alpha_{t-1} \quad \text{for} \quad t = 2, \ldots, T,$$

where T is the number of sentences in the document and

$$D_{O_t} = I - \text{diag}\{b_1(o_1), b_2(o_2), ..., b_{2s+1}(o_{2s+1})\},$$

where I is the identity matrix, $b(.)$ is cumulative density function for the χ^2 distribution with the number of degrees of freedom equal to the number of components in O_i, and the argument $o_i = (O_t - \mu_i)^T \Sigma^{-1} (O_t - \mu_i)$, where μ_i is the mean for the ith state.

The probability of the entire observation sequence given the model is given by

$$\omega \equiv Pr(O) = \sum_{i=1}^{2s+1} \alpha_T(i). \tag{5}$$

We define $\beta_t(i)$ to be the probability that we will observe the sequence $\{O_{t+1}, O_{t+2}, \ldots, O_T\}$ given that we are at state i of our HMM. A backwards recursion lets us compute $\beta_t(i)$ by initializing β_T to all ones, and then computing

$$\beta_t = MD_{O_{t+1}}\beta_{t+1} \quad \text{for} \quad t = T-1, \ldots, 1.$$

The results of these two recursions are combined to form $\gamma_t(i)$, the probability of being in state i for sentence t given the sequence of observations $\{O_1, O_2, ..., O_T\}$ and the HMM. The formula is given by

$$\gamma_t(i) = \frac{\alpha_t(i)\ \beta_t(i)}{\omega}.$$

Thus, $\gamma_t(i)$ gives the probability that sentence t corresponds to state i. If i is even then this probability represents the probability that sentence t is the $i/2$-th summary sentence. If i is odd then it is the probability that it is a non-summary sentence. We compute the probability that a sentence is a summary sentence by summing $\gamma_t(i)$ over all even values of i. This posterior probability is used to select the most likely summary sentences. We denote this probability as

$$g_t = \sum_{i \text{ even}} \gamma_t(i). \tag{6}$$

3.3 Extracting the Best Sentences

For the HMM approach, we present two methods for extracting summaries with k sentences.

The first simply chooses those sentences with the maximum posterior probability of being a summary sentence. For a summary of length k, we choose the sentences with the k largest values of g_t.

The second approach we propose is to use the QR decomposition to remove any redundant sentence that might be included by the HMM maximum posterior probability method. We use a variant of the Gram-Schmidt process to extract k "non-redundant" summary sentences. We first choose the sentence with maximum posterior probability, as in the previous method. Within the matrix A, we subtract from each remaining sentence the component in the direction of this chosen sentence, and then among the columns that have 2-norm greater than or equal to one, we choose the one with maximum posterior probability. This process is iterated, updating the matrix $k-1$ times until k sentences are chosen to be in the summary.

To perform this computation, we make a minor change to Algorithm 2.1; the choice of column is changed to

> Choose column ℓ of A to be the column of maximum posterior probability norm among all columns with remaining norm greater than one.

We also used the variant that updated norms but not all of the columns.

4 Evaluating Summaries

Evaluating the goodness of a summary is not a well-understood process. Here are some of the measures used, and their limitations.

- Sometimes we have summaries that have been created by human readers. Counting the number of sentences that are common to our summary and their summary (or summaries) and performing a precision/recall measurement gives an easily computed measure. The quality of the measure is limited, however. The coherence and readability of the summary is not considered. Sometimes two sets of sentences can be disjoint but contain the same information, so a good summary might be scored too low.
- The summaries can be evaluated by a human reader. This is somewhat subjective and too time consuming for large trials.
- We could see if the summary shares some useful statistic with the original document. For instance, we might want the distribution of terms to be similar in both. Unfortunately, such statistics can often be nearly approximated by random choice of a subset of sentences.

Despite its limitations, we will make use of the first measure in our experiments.

5 Experimental Results

The documents we used in our test were taken from the TREC data set [TRE97]. This included articles from the Associated Press, *Financial Times*, *Los Angeles Times*, *Washington Post*, *Wall Street Journal*, *Philadelphia Inquirer*, *Federal Registry*, and *Congressional Record*. A single person (M) generated summaries for all of the documents. We also had summaries generated by three other people (E, B, and T) for about 40 documents, and we present results on agreement with these summaries as well.

Each of these data sets was divided into two pieces, one used to train the parameters of the model and one used for evaluation of the methods.

We compared our algorithms' summaries with the human summaries, computing the following scores. For each document we let k_h be the length of the human summary, k_m the length of the machine generated summary, and r the number of sentences they share in common. Then we define, precision (P), recall (R), and F_1 as metrics to compare the two summaries by:

$$P = 100\frac{r}{k_m}, \tag{7}$$

$$R = 100\frac{r}{k_h}, \tag{8}$$

	# files	percntl	expht	tail	Length avdif	stdif
ap-dev	176	45	6.42	3.03	2.15	2.17
ap-test	176	45	6.42	3.03	1.73	2.19
cr-dev	73	50	5.40	3.82	-0.90	8.42
cr-test	37	50	5.40	3.82	-4.14	22.50
fr-dev	97	50	7.83	1.66	1.51	6.30
fr-test	99	50	7.83	1.66	0.52	7.48
ft-dev	112	45	7.01	3.07	0.83	2.32
ft-test	112	45	7.01	3.07	0.93	2.16
latwp-dev	70	15	6.38	1.37	-0.94	1.67
latwp-test	76	15	6.38	1.37	-0.47	1.78
pi-dev	42	20	3.55	1.20	0.10	2.29
pi-test	34	20	3.55	1.20	-0.15	1.96
wsj-dev	100	45	5.10	2.46	0.27	2.25
wsj-test	100	45	5.10	2.46	0.04	2.45

Table 2: QR method with pivoting. Values of parameters and size of data sets.

Data	DimSum(4,γ)	QR	HMM(2,γ)	HMM(2,QR)	HMM(4,γ)
ap-test	52	46	56	56	55
cr-test	34	39	53	54	56
fr-test	33	43	46	49	47
ft-test	46	51	59	59	57
latwp-test	35	57	45	45	45
pi-test	46	55	48	48	48
wsj-test	51	62	72	72	72

Table 3: Precision of Various Algorithms

and

$$F_1 = 100\frac{2PR}{P+R} = 100\frac{2r}{k_h + k_m} \tag{9}$$

In the following tables we report average precision, average recall, and average F_1 for various data sets. These are simply defined as the mean value of the respective score over the document set.

5.1 Comparison Among Methods

First we compared the performance of several algorithms:

- The naive Bayesian approach of [AOGL97] using three features: position of the sentence in the document, paragraph position, and tf.idf.
- The pivoted QR method. We chose the parameters to maximize the F1 score over the set of training documents. Table 2 gives the values of the

Data	DimSum(4,γ)	QR	HMM(2,γ)	HMM(2,QR)	HMM(4,γ)
ap-test	52	60	58	58	58
cr-test	36	40	43	43	44
fr-test	49	47	51	54	51
ft-test	53	57	50	50	49
latwp-test	61	49	67	66	66
pi-test	44	53	66	66	65
wsj-test	56	62	61	61	62

Table 4: Recall of Various Algorithms

Data	DimSum(4,γ)	QR	HMM(2,γ)	HMM(2,QR)	HMM(4,γ)
ap-test	52	52	56	56	55
cr-test	35	39	47	47	48
fr-test	39	41	46	48	46
ft-test	46	53	53	53	51
latwp-test	45	51	53	52	53
pi-test	41	53	55	55	54
wsj-test	54	60	65	65	65

Table 5: F1 Scores of Various Algorithms

parameters. Results were insensitive to the choice of g and t; almost as good results were obtained from $t = 3$, $g = 10$.

- Variations of the HMM method. A 9-state HMM was build using the combined training data from six sets. We chose the length to be T^α where T is the number of sentences in the document and the optimal α, based on the training data, is approximately 0.5636. We choose the sentences in the summary either as those with largest γ scores, or by using the QR decomposition to remove redundant sentences, as discussed in Section 3.3. We also tested the use of 2 features, number of terms and document term probability, (o_2 and o_4) vs. the use of all four features from Table 1.

The resulting scores are given in the tables: Table 3 for precision, Table 4 for recall, and Table 5 for F1.

The results are a significant improvement over naive Bayesian results even for the non-news data sets; the HMM was 4-14 points better.

5.2 Short Summaries

We now look at the performance of the HMM for extracting short summaries. Tables 6 and 7 give results for extracting summaries of length either 2 or 4.

The QR method of sentence extraction improves the results here, whereas the results for longer summaries are comparable with those of simply using the maximum posterior probability method.

Data	Precision		Recall		F1	
	(4,γ)	(4,QR)	(4,γ)	(4,QR)	(4,γ)	(4,QR)
ap-test	92	92	37	37	50	50
cr-test	92	92	21	21	33	33
fr-test	62	69	20	22	29	32
ft-test	88	88	32	32	44	44
latwp-test	82	83	36	37	49	50
pi-test	85	85	41	41	52	52
wsj-test	94	95	44	44	55	55

Table 6: Scores of summaries of length 2 generated by HMM

Data	Precision		Recall		F1	
	(4,γ)	(4,QR)	(4,γ)	(4,QR)	(4,γ)	(4,QR)
ap-test	63	63	49	49	53	53
cr-test	72	72	33	33	42	42
fr-test	55	59	36	39	40	43
ft-test	63	64	43	44	48	48
latwp-test	57	56	49	49	52	51
pi-test	63	63	56	56	57	57
wsj-test	69	69	61	61	59	59

Table 7: Scores of summaries of length 4 generated by HMM

	HMM	QR	B	E	M	T
HMM	100					
QR	75	100				
B	58	59	100			
E	52	52	61	100		
M	54	53	48	49	100	
T	51	50	60	60	49	100

Table 8: Comparison of HMM and pivoted QR, by F_1 score, with various human-generated summaries. The QR training was performed with the M summaries for latwp-dev, omitting titles.

HMM	The probe started with the House Post Office but, now, two years after federal prosecutors began investigating Ways and Means Committee Chairman Dan Rostenkowski, D-Ill., the allegations of official misconduct have moved far beyond stamps. In fact, the initial allegations that Rostenkowski traded postage vouchers for cash at the House Post Office now rate as a comparatively weak portion of the government's case, so much so that his defense lawyers have plotted to use them to undermine the rest of the case, sources close to Rostenkowski say. Lawyers see the former House postmaster who told prosecutors Rostenkowski participated in the allegedly illegal exchanges as a less than ideal witness. Either choice would knock the powerful 66-year-old Chicagoan from his influential chairmanship and prominent role in shaping President Clinton's health care legislation and major trade, welfare and campaign finance bills. Such negotiations are considered normal and cannot be used against Rostenkowski should the case go to trial.
QR	The probe started with the House Post Office but, now, two years after federal prosecutors began investigating Ways and Means Committee Chairman Dan Rostenkowski, D-Ill., the allegations of official misconduct have moved far beyond stamps. In fact, the initial allegations that Rostenkowski traded postage vouchers for cash at the House Post Office now rate as a comparatively weak portion of the government's case, so much so that his defense lawyers have plotted to use them to undermine the rest of the case, sources close to Rostenkowski say. Lawyers see the former House postmaster who told prosecutors Rostenkowski participated in the allegedly illegal exchanges as a less than ideal witness. Either choice would knock the powerful 66-year-old Chicagoan from his influential chairmanship and prominent role in shaping President Clinton's health care legislation and major trade, welfare and campaign finance bills. U.S. Attorney Eric H. Holder Jr. has outlined for the Justice Department what has been described as a "kitchen sink" of alleged abuses of Rostenkowski's official accounts for postage, leased automobiles, office space, supplies and personnel. Such negotiations are considered normal and cannot be used against Rostenkowski should the case go to trial.

Table 9: Sample summaries generated by the various methods.

B	The probe started with the House Post Office but, now, two years after federal prosecutors began investigating Ways and Means Committee Chairman Dan Rostenkowski, D-Ill., the allegations of official misconduct have moved far beyond stamps. In fact, the initial allegations that Rostenkowski traded postage vouchers for cash at the House Post Office now rate as a comparatively weak portion of the government's case, so much so that his defense lawyers have plotted to use them to undermine the rest of the case, sources close to Rostenkowski say. Rostenkowski faces a Tuesday deadline for accepting a plea bargain and almost certain jail time or fighting to salvage what is left of his public reputation by challenging a litany of charges in court. U.S. Attorney Eric H. Holder Jr. has outlined for the Justice Department what has been described as a "kitchen sink" of alleged abuses of Rostenkowski's official accounts for postage, leased automobiles, office space, supplies and personnel. Rostenkowski, completing his 36th year in Congress, entered plea negotiations in an effort to reduce or eliminate any prison sentence while avoiding a lengthy legal battle and possibly retaining his Ways and Means chairmanship, sources familiar with the discussions said.
E	The probe started with the House Post Office but, now, two years after federal prosecutors began investigating Ways and Means Committee Chairman Dan Rostenkowski, D-Ill., the allegations of official misconduct have moved far beyond stamps. In fact, the initial allegations that Rostenkowski traded postage vouchers for cash at the House Post Office now rate as a comparatively weak portion of the government's case, so much so that his defense lawyers have plotted to use them to undermine the rest of the case, sources close to Rostenkowski say. Rostenkowski faces a Tuesday deadline for accepting a plea bargain and almost certain jail time or fighting to salvage what is left of his public reputation by challenging a litany of charges in court. U.S. Attorney Eric H. Holder Jr. has outlined for the Justice Department what has been described as a "kitchen sink" of alleged abuses of Rostenkowski's official accounts for postage, leased automobiles, office space, supplies and personnel. Rostenkowski, completing his 36th year in Congress, entered plea negotiations in an effort to reduce or eliminate any prison sentence while avoiding a lengthy legal battle and possibly retaining his Ways and Means chairmanship, sources familiar with the discussions said. Under normal procedures, Rostenkowski would have to relinquish his chairmanship if indicted on any felony punishable by at least two years in prison. If he pleads guilty and is given jail time, efforts to remove him from the chairmanship would likely come immediately. Faced with the ugly options, Rostenkowski is leaning toward fighting, knowing he will have to cast doubt on each of the allegations in the laundry list. According to sources knowledgeable about the case, the allegations of "ghost employees," unrelated to the House Post office, appear the most difficult to counter.

Table 10: Sample summaries generated by the various methods.

M	Rosty Weighs Options on Plea Offer (Washn) The probe started with the House Post Office but, now, two years after federal prosecutors began investigating Ways and Means Committee Chairman Dan Rostenkowski, D-Ill., the allegations of official misconduct have moved far beyond stamps. Rostenkowski, completing his 36th year in Congress, entered plea negotiations in an effort to reduce or eliminate any prison sentence while avoiding a lengthy legal battle and possibly retaining his Ways and Means chairmanship, sources familiar with the discussions said. According to sources knowledgeable about the case, the allegations of "ghost employees," unrelated to the House Post office, appear the most difficult to counter. The inquiry was also expanded to cover Rostenkowski's purchases of personal and gift items through his expense account at the House Stationery Store. His official leases of three automobiles from a Chicago-area dealership and subsequent acquisition of them as a private owner have also come under prosecutors' scrutiny.
T	The probe started with the House Post Office but, now, two years after federal prosecutors began investigating Ways and Means Committee Chairman Dan Rostenkowski, D-Ill., the allegations of official misconduct have moved far beyond stamps. In fact, the initial allegations that Rostenkowski traded postage vouchers for cash at the House Post Office now rate as a comparatively weak portion of the government's case, so much so that his defense lawyers have plotted to use them to undermine the rest of the case, sources close to Rostenkowski say. Rostenkowski faces a Tuesday deadline for accepting a plea bargain and almost certain jail time or fighting to salvage what is left of his public reputation by challenging a litany of charges in court. According to sources knowledgeable about the case, the allegations of "ghost employees," unrelated to the House Post office, appear the most difficult to counter. His official leases of three automobiles from a Chicago-area dealership and subsequent acquisition of them as a private owner have also come under prosecutors' scrutiny.

Table 11: Sample summaries generated by the various methods.

5.3 Comparison to Other Human-Generated Summaries

To further test how well human summaries could be predicted, three additional people generated extract summaries for around 40 articles from the latwp-test data set. The QR method and the HMM with two features with the QR method of extraction of sentences, trained using the latwp-dev data and summaries generated by M, were compared against each of the four human summaries. In addition the four humans summaries were compared against each other. One caveat: the summaries generated by M included the title 98% of the time, while the other summarizers had been instructed to omit it, so we matched our assumption about the title to the data set with which we compared. Table 8 gives the F_1 scores that resulted from these comparisons. Comparing the HMM to the four human summarizers the F_1 scores range from 51 to 58; comparing the QR with the four human summaries gave scores between 50 and 59, while the F_1 scores between human summmarizers range from 48 to 61. Summarizer B had agreement at least 59 with summarizers E and T (but not with M) and also with the QR method. Summarizer B had the maximum agreement with HMM, and HMM and QR agreed at 75, significantly higher than agreements among the humans.

The actual summaries produced by the various methods and people for one of the articles in the latwp-test collection are given in Table 9, 10, and 11.

6 Conclusions and Future Work

We have presented two novel algorithms for generating sentence abstract summaries of documents. The algorithms are quite successful in generating summaries that agree well with human-generated summaries, despite using minimal natural language processing (NLP) information, just the extraction of terms.

Coupling these techniques with more sophisticated NLP techniques could enable us to generate summaries from phrases and clauses rather than complete sentences.

If the summary is query based, then we need to bias it toward information requested in the query. In the HMM, this can be implemented by adding a feature relating the query to each sentence. In QR it might be accomplished by term weighting.

We also plan to investigate multi-document summaries, in which a single set of sentences is needed to summarize a collection of documents.

7 Acknowledgments

The authors would like to acknowledge the substantial contributions of the people who generated summaries. Benay Dunn (B), Ed Beimel (E), and Toby Merriken(T) each summarized approximately 50 documents used here, and Mary Ellen Okurowski (M) summaried well over 1000. Without their efforts, evaluating our ideas would have been impossible.

We also benefited from helpful conversations with Oksana Lassowsky, David Harris, Scott Connelly, Anthony Taylor, and Mary Ellen Okurowski.

References

[AOGL97] C. Aone, 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.

[BPSW70] 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.

[Bri93] E. Brill. *A Corupus-based Approach to Language Learning.* Ph.d. thesis, University of Pennsylvania, 1993.

[BYRN99] R. Baeza-Yates and B. Ribeiero-Neto. *Modern Information Retrieval.* Addison Wesley, 1999.

[CG98] J. G. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing sumaries. *Proceedings of SIGIR-98 (Melbourne, Australia)*, pages 335–336, 1998.

[GL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, third edition, 1996.

[HM00] U. Hahn and Inderjeet Mani. The challenges of automatic summerization. *IEEE Computer*, 33(11):29–36, 2000.

[JK99] H. Jing and K. R. Mc Keown. The decomposition of human-written summary sentences. *Proceedings of SIGIR-99 (Melbourne, Australia)*, pages 129–136, 1999.

[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 Infomation Retrieval*, pages 68–73, 1995.

[Kru95] G. Krupka. Sra: Description of the sra system as used for muc-6. *Proceeding of Sizth Message Understanding Conference (MUC-6)*, 1995.

[Luh58] H. P. Luhn. The automatic creation of literture abstracts. *IBM Journal of Research Development*, 2:159–165, 1958.

[MBF+90] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five papers on wordnet. *Technical Report CSL REport 43, Conitive Science Laboratory, Princeton University*, 1990.

[Rab89] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77:257–285, 1989.

[Ste99] G. W. Stewart. Four algorithms for the efficient computation of truncated pivoted qr approximations to a sparse matrix. *Numerische Mathematik*, 83:313–323, 1999.

[TRE97] TREC Conference Series. Text REtrieval Conference (TREC) text research collection. Technical Report http://trec.nist.gov/, National Institute of Standards and Technology, Gaithersburg, Maryland, 1994, 1996, 1997.