Identification of Important Text in Full Text Articles Using Summarization

Improving MTI Full Text Indexing

1.0 Introduction

Other research has shown that although the abstract is more information dense, the full text of a scientific article in the biomedical domain has much greater information content.[1] We know from observing indexers and studying their indexing process that some of the assigned MeSH concepts do not appear in the abstract. The indexing manual also dictates that the abstract should not be used during indexing. Thus for accurate subject analysis the full text is important. However, the greater content in the full text makes more difficult the task of deciding which of the concepts identified by the MTI indexing methods are the most important.

We propose to address this problem by using summarization techniques to identify the important text and then submit that text to MTI for indexing. Automatic text summarization is the distillation of the most important information from a source to produce an abridged version for a particular user and task. The indexing task is the capturing of the essential intent of the author in a manner to support accurate retrieval. The distillation task could be viewed as a component of the more complex indexing task. The first phase of summarization is the analysis of the source and selection of a few salient features. This process maps closely to" the subject analysis of the source," the language used in the indexing manual when referring to the indexing process. Many summarization approaches simplify the transformation and synthesis phases by ranking all the sentences and then using the top *N* sentences as the summary. The number of sentences, *N* is either some fixed value or some percentage of the document length. For MTI's purposes we will tune *N* to maximize MTI performance.

Following the example of Yeh, Ke, Yang, and Meng[2] we will use Latent Semantic Analysis (LSA) and a Text Relationship Map (TRM) to create a ranked list of the sentences in the article. They use such a ranking to prepare an indicative, extract-based summary. They mention that "the most important thing is the selection of salient terms, and to take more semantics, such as named entities and noun phrases into consideration." I propose that we use MetaMap mappings to the ULMS Metathesaurus as the terms in the sentence vectors. This includes semantics by raising the analysis from term based analysis to a concept based one. Alternatively, we could keep the heads of the noun phrases that are not mapped by MetaMap and the verbs from the sentences in the sentence vectors. (This approach would complicate the implementation and increase the size of the vectors.)

Latent semantic analysis is a mathematical technique for extracting and inferring relations of expected contextual usage of words in passages of discourse. Yeh *et al* use it to convert

the keyword based representation of the sentences of a document into a semantic sentence representation. The text relationship map shows intra-document links between nodes whether they are paragraphs or sentences. They use the semantic sentence representation as the nodes of the TRM. The weights on the links are based on the similarity of the nodes. We will apply the TRM to summarization by ranking the sentences by the sum of the weights of the links at the node for that sentence. The weights for the links in the TRM are based on the semantic sentence representation of the document. The summary was then generated by taking the required number of sentences from those with the most significant links.

1.1 Related Work

Using summarization techniques to improve text categorization has also been used by Ko, Park, and Seo[3]. They recognized the limitations of classification based on the conventional representation of the document as a vector of features using term frequency. The term frequency does not take into account the term location. They point out that "each sentence in a document has different importance for identifying the content of the document. They achieved better results by assigning weights according to the importance of sentence. They determine the importance of each sentence (1) by the similarity of the sentence to the title, and (2) by the importance of terms in the sentence. The latter was measured by the normalized product of the term frequency, inverse document frequency and χ^2 statistic which reflects the correlation of the term to the classification category. They only achieved a .005 to .01 improvement in the F1 measure using a SVM classifier.

Although this work shows some utility for the use of summarization techniques its particular techniques are not applicable in the MTI environment. MTI in its post processing already gives special emphasis to terms identified from both the title and the abstract. Since our classification is run in a binary fashion, there is no specific category context in which to compute a χ^2 statistic. Their document collections were two Newsgroups which are presumably much smaller than the full text biomedical articles. The larger documents in the full text collection should show more benefit from this approach.

Yeh, Ke, Yang, & Meng point out the similarity of information retrieval and text summarization and the extensive use of the former for the latter in the 1990s. Their criticism of most of those techniques is that those retrieval techniques focus on symbolic-level analysis and do not take into account semantics. The use of latent semantic analysis to cluster terms into semantic groups, that I would call concepts, feeds into the text relationship map raising the summarization from keyword-level analysis to semantic-level analysis.

Gong and Lui[4] proposed two methods: one used relevance measure to rank sentence relevance, and the other used latent semantic analysis to identify semantically important sentences.

Yeng *et al* got better performance from a modified corpus-based approach that employed a genetic algorithm to optimize the combination of features. However, they point out advan-

tages that are relevant to our situation and convinced us to use their latent semantic analysis and a text relationship map.

We avoid some of the problems they identified with the LSA + T.R.M. approach. We will have little polysemy when we map to UMLS concepts instead of MeSH. Also, we will not face the presence of proper names in different contexts since the use of medicines, proteins, and genes may be more consistent within a single article.

Goldstein, et al.[6] evaluated sentence ranking metrics and combined statistical and linguistic features in the context of text document summarization. The linguistic features are selected by empirical studies of existing summaries for a particular corpus with the goal of finding those that distinguish sentences in the summaries from the others in an article. For example, they found that the indefinite article “A” started summary sentences 62% more often than it did in non-summary sentences. Also that auxiliary verbs, such as “was” or “could”, appeared more often in summary sentences. This suggests an alternative to our current selection approach based on a more conventional bag of words approach. For queries used in their approach, we could use the title as our query, since they found adding the title to the user query a sometimes effective query expansion technique. To create training summaries since we do not have human prepared summaries, we can take the Medline indexing and determine which sentences of the article contain MTI identified terms matching those terms. (It would be informative to also know which Medline terms are not found by MTI.)

2.0 Methods

Our overall approach will be to represent the document as a set of vectors, one for each sentence. We will use ULMS Metathesaurus concepts as the features in those sentence vectors. We will use the MMI output to identify those concepts in each sentence. Applying the LSA + TRM approach we will generate a ranked list of the sentences in the document. We will then experiment with processing of increasing amounts of text by MTI to determine the optimal body of text for indexing.

To summarize the LSA + TRM approach: We will build a concept by sentence matrix. This matrix is then transformed by singular value decomposition and dimension reduction. Each column of the resulting matrix becomes the semantic representation of a sentence. These semantic sentence vectors are used to compute the similarity between each pair of sentences. Those values become the link weights on the text relationship map. The bushiness metric for each sentence (node) is the sum of the weights of the outgoing links. The k globally bushiest sentences from the map become the target for the MTI indexing.

2.1 Feature Identification

Our first step is to build a concept by sentence matrix. This requires that we first identify the sentences and the concepts for the whole document. For our purposes the document is either the full text article or a subset of its contents such as our current section-based model.

The text will be re-extracted from the XML files dropping the references, authors, and formatting information. Sentence boundaries will be determined. The sections, and paragraphs will be preserved in a numbering scheme for the sentences. This will facilitate possible later use of subsets of the full text.

For identifying UMLS concepts the previously computed MMI output will be used and the concepts identified will be associated with the sentence from which they came. The set of identified concepts become *W,* the vocabulary for the document. Let $|W| = m$. If *S* is the set of sentences in document *D*, then $|S| = n$. We also compute the frequency of W_i in S_j and call it w_{ij}. These values become the building blocks for the concept by sentence representation of the document.

2.2 Semantic Model Analysis

Latent semantic analysis takes a term by sentence matrix representation of a document and applies singular value decomposition (SVD), a technique from linear algebra. The term by sentence matrix, *A*, is an $m \times n$ where *m* is the number of terms and *n* is the number of sentences. The SVD factors matrix *A* into the product of three matrices the outer two of which have columns of left and right singular vectors respectively, see.equation 1.

$$.A = U\Sigma V^T \quad \text{(EQ 1)}$$

The center matrix is a diagonal matrix(Σ) whose diagonal elements are non-negative singular values sorted in descending order. For some rank, *r*, the rest of the diagonal elements are zero. From a transformation point of view the SVD derives a mapping between the m-dimensional space spanned by the weighted term-frequency vectors and the r-dimensional singular vector space with all of its axes linearly-independent.

Next, dimension reduction is applied to diagonal matrix based on a parameter discussed below, the dimension reduction ratio. Since up to the rank of *r* the diagonal entries are zero, there is no loss of information in Σ. This mapping projects the term-frequency vectors of a sentence (column of A) to the a column vector of V^T with *r* rows. Thus we can also remove the final rows up to *r.* Similarly, a row vector of *A*, showing the frequency of some term in all the sentences, is mapped to a row vector in *U* that has only *r* columns. Thus we get a semantic model of the document when we generate matrix A' by multiplying the three reduced dimension components:

(EQ 2)

$$A' = U'\Sigma'V'^T$$

From a semantic point of view, the SVD derives the latent semantic structure from the document represented by the $term \times sentence$ matrix. This operation reflects a breakdown of the original document into r linearly-independent base vectors or concepts. A unique SVD feature is that it is capable of capturing and modeling interrelationships among terms so that it can semantically cluster terms and sentences. Furthermore, if a word combination pattern is salient and recurring in a document this pattern will be captured and represented by one of the singular vectors.[4]

To analyze the effects of expected contextual usage of words in different levels, Yeh *et al.* constructed two types of semantic matrices, one for single-document level, and the other for corpus level. Their experiments got better performances (~.10, 37%) at the single-document level. They discussed the reasons for the weaker performance at the corpus level as though they expected the corpus-level to do better because of the properties discussed above.

For our text collection, there is so much diversity and the documents are numerous and long, that an investigation of the whole corpus is inappropriate and would be computationally difficult. Performing latent semantic analysis at the journal issue level may be more appropriate since common patterns of concept occurrence would reflect common concept or point of view. (Even some journals such as the *Proceedings of the National Academy of Science*, may be too diverse for expansion beyond the document level.) Thus, we will construct two types of semantic matrices. The first will be document level and the second will be at the journal level.

2.2.1 Matrix construction

We will build a concept by sentence matrix, A, populating it with the entries, a_{ij}, that are the product of G_i, the global weight of W_i in D (based on normalized entropy of the concept) and L_{ij}, the local weight of W_i in S_j (based the local frequency). These two factors are defined in equation 3, 4, and equation 5. w_{ij} is the frequency of W_i occurring in S_j; c_j is the number of concepts identified in S_j. Although the simple frequency can serve as the local weight we will follow Bellegarda *et al*[7] and use the log to dampen the effects of large differences in counts and normalize for sentence length:

$$L_{ij} = log_2\left(1 + \frac{w_{ij}}{c_j}\right) \quad \text{(EQ 3)}$$

The relative frequency of W_i in S_j is obtained as:

$$f_{ij} = \frac{w_{ij}}{d_i} \quad \text{(EQ 4)}$$

d_i is the frequency of W_i in D. The normalized entropy of W_i, E_i, has a value close to 1 when the term is distributed across many sentences throughout the article. Conversely, a value of E_i near zero means it occurs in just a few sentences and is thus more valuable for

indexing a sentence. So the global weight is 1-E_i. If n is the number of sentences in D, then the global weight for sentence i is specified by equation 5.

$$G_i = 1 - E_i \qquad E_i = \frac{-1}{\log(n)} \sum_{j=1}^{n} f_{ij} \log(f_{ij}) \qquad \text{(EQ 5)}$$

Gong and Liu found binary term weighting to be most effective in their application of LSA to summarization. This local weighting scheme assigns 1 to terms appearing in the sentence and 0 to those that do not. They used no global weighting or normalization of terms weights in the term-frequency vectors. The Yeh *et al* weighting scheme detailed above uses formulas that differ from those tested by Gong and Liu. The rationale above from Bellegarda seems sufficient to justified the more complex weighting, though testing the binary option could be considered.

2.2.2 Singular Value Decomposition

This matrix is then transformed by singular value decomposition and dimension reduction. The columns of the resulting matrix become the semantic representation of the sentences.

This step requires linear algebra libraries to handle the implementation of the algorithms.A variety of platforms are available:

- Could include sub-matrix SVD (see Li, Lu, & Shi)[5]
- Fortran routines available: http://www.caam.rice.edu/software/ARPACK/
- Mathmatica 5.1 has support.
- There are also free Java libraries that support SVD: mathlab at SourceForge
- JAMA is a complete Java implementation, but there were know problems with the SVD implementation.

Our final selection was a Java toolkit on top of a C based implementation of BLAS and LAPACK. Matrix Toolkits for Java (MTJ) was created by Bjørn-Ove Heimsund at a Norwegian university. Here is the link to its web page: http://rs.cipr.uib.no/mtj/.

Installing was complicated, because there were several underlying packages that needed to be compile on the local host. (They were built for solaris 9 (SunOS 5.9) and are on nls11 at /usr/local/lib: libclapack.so 5.8M and libblas.so 4.4M.

2.2.3 Dimension Reduction Ratios

The selection of factors remaining in A' is determined by the dimension reduction ratio. The initial reduction is applied to the singular value matrix Σ. If the rank of Σ is n and the dimension reduction ratio is .7, then the rank of Σ', r', is $0.7 \cdot n$. If r is the rank at which the diagonal values of Σ become zero, then there is no distortion in the mapping from A to A'. Bellegarda, *et al.*[7] assert that reasonable values for r' are 100 to 200. (Other LSA researchers speak of 100-300 as parsimonious.) This is in the context of a vocabulary with the order of ten thousand terms. So their reduction ratio is approximately 0.02, but their use of latent semantic analysis is for word clustering. Yeh, et al. use different dimension reduction ratios for each corpus and compression rate. (The compression rate is the ratio of the number of sentences selected for the summary compared to the total number of sentences in the document.) The smaller summaries did better with higher reduction ratios. At the same compression rate, the optimal dimension reduction ratio varied from 0.6 to 0.8. (Note, that their documents had 25-30 sentences on average. The articles in our document collection are probably 5-10 times that size.)

In our context we do not have an a priori compression ratio or target number of sentences. Since our ultimate target is 25 MeSH terms and we usually have no trouble finding those in text the size of an abstract, we probably will be interested in a small number of the most important sentences. This argues for a low compression rate and following their experience a lower dimension reduction ratio, i.e. fewer meta-concepts. We will start at 0.5 for our dimension reduction ratio. After we have found a optimal number of sentences for our condensed document we will return and tune the dimension reduction ratio experimentally. Another check on the starting value will be to look at the values of *r*, the rank of the last of the non-zero singular values in Σ. We want our r' to always be less than *r*.

2.2.4 Dimension Reduction

So having determined our dimension reduction ratio, the value for r' becomes $0.5 \times n$. The matrix Σ' is trimmed to $r' \times r'$. Thus the matrix U' becomes $m \times r'$, so we must remove some high index columns. Similarly, for V' that becomes $r' \times n$.

2.2.5 Semantic Matrix Reconstruction

The semantic matrix A' will be generated by the simple execution of equation *2*, multiplying the dimension reduced components of the singular value decomposition of *A*.

Gong & Liu rank the sentences by taking the ranking of the silent topic/concepts as reflected in the magnitude of the corresponding singular value and picking out the sentence that best represents each as indicated by its having the highest index value for that topic/concept. We will be using a text relationship map instead.

2.3 Text Relationship Map Construction

The column vectors from A' form the semantic sentence representation for our document. These vectors are used to compute the similarity between each pair of sentences. The k globally bushiest sentences become the target for the MTI indexing. The bushiness metric is the sum of weights of the outgoing links.

2.3.1 Generating graph

Our graph is complete in that all relationship of all pairs of sentences in S are considered. Thus our text relationship map (TRM) may be represent at a $n \times n$ matrix of the link weights. The bushiness of each node becomes the sum of the weights in its row.

2.3.2 Computing link weights

Since the nodes, i.e. sentences, are represented as vectors we compute their similarity using the inner product between the vectors of the corresponding sentences. The bushiness of any sentence with itself is 0.

$$sim(S_i, S_j) = \frac{\vec{S}_i \cdot \vec{S}_j}{|\vec{S}_i||\vec{S}_j|}$$

2.3.3 Text Relationship Map

Yeh *et al* applied a minimum weight threshold for inclusion in the map to the links. We will build a complete text relationship that includes all the similarity measures. Then we will use the approach of Kim, Kim, and Hwang[8] and compute the bushiness in the graph as the sum of the similarity measures on the links emanating from a given node. This aggregate similarity becomes our measure the importance of a sentence. Yeh et al followed Salton's approach that counts the number of links connecting a node to other nodes. Our approach gives us a weighted ranking of all the sentences in the article.

2.4 Sentence Selection

Yeh *et al* generated their summary by taking the required number of sentences from those with the largest number of links. We rank the sentences using the aggregate similarity.

Depending on our selection policy, we select some sentences from that ranking to form a new document, D', that will be submitted to MTI for processing. The title of D' will be the title from the original document. The fundamental selection policy is to take the *k*-first sentences from the ranked list to include in D'. The value of *k* may be tuned to find the value that yields the best MTI F_2-measure performance.

Goldstein, et al.[6] observed that the size of human created summaries are not proportional to the length of the article. Therefore, they discourage the use of a fixed compression ratio in the evaluation of summarizers. In the Medline context we know that the guidelines for the number of terms used in indexing is generally not related to the size of the articles. Those publication types that do not receive in-depth indexing are generally shorter articles, such as letters and editorials. However, the longest articles are often reviews and that publication type also gets a small set of indexing terms. Therefore, our target *k* will be a fixed number of sentences, and we will not seek to optimize a particular compression ratio but rather the summary length.

When experimenting on processing increasing amounts of text, we need to determine whether document length should be a factor in that setting that threshold. We will re-examine our initial decision at that time. For now if there are fewer sentences in a particular article than the target number (*k*), we will process the whole article.

2.5 MTI processing

The initial MMI processing was completed during feature identification. We will not be able to reuse that processing since the scores provided are for all the sentences for which a term was identified, not just the ones in the summary. So when the D' documents are created, they will be processed normally by MTI. Any future production use of this technique will require some modification of MMI to allow reuse of the MetaMap mappings.

2.6 Experiments

Once we have a list of ordered sentences we need to determine the size of the ideal or best performing summary. Our D' documents have k sentences, so optimizing k is the focus of our tuning trials. There are also other potential variables that need to be considered in our experimental design to test the use of summaries as a technique for enhancing MTI performance on full text articles.

One design decision for the selection policy involves the abstract. (1) It could be included in the analysis but always included in the selected text for processing regardless of its sentences ranking in the LSA + TRM. (2) Alternatively, it could be left out of the analysis and included by default. (3) A final option is to include abstract sentences in the processing and just select the top k sentences for MTI processing. While evaluating the use of summaries it seems to make sense to stick strictly to that approach and chose the third option for now.

Another policy consideration is whether to integrate this approach with the section based modeling that has been done so far. We could limit the document *D* that goes into the summarization process to the selection of sections in our best performing model. Initially we will treat this as a refinement of the section selection results and do the summarization on the current best performing model.

2.6.1 Test Collection

The full text collection consists of 496 articles with Medline indexing from 17 journals available online through PubMed Central. To provide statistically stronger results we will divide the test collection into four parts and use three parts for training and one for testing. We can do this in a rotating fashion to validate our results.

2.6.2 Trial Design

Running an exhaustive set of trials to select the optimal summary length (k) is probably not practical. So we need a strategy for finding the optimal k without running documents of size 1 to k+3. Therefore, after the first partition of the test collection has been summarized we will have size statistics for all the documents. We will use the value of 20% of the average number of sentences as the starting k. Our trials can then be for smaller and larger values near that number going further in the direction that yields better results. The 20% value is chosen because that approximates the summary size when humans created selected sentence summaries.[6]

For the search to find the optimal value we will start by adding 2, then 4, 8, etc. When results fail to improve for 2 trials then we will try values in the gap between the best value and the next higher value tried. When searching a bounded interval, such as [0,j] we try the value closest to the midpoint. The next interval is one on the same side of the best values as the previous interval.When a maximum appears to have been found, a final trial is run in the interval between the two highest values (if possible).

Another computational short cut will be to apply the approach above on the first quarter partition of the collection. Then add the second partition and repeat the process this time starting at the value of k obtained so far. This is repeated again adding the third partition. The advantage of this approach is the fewer documents that must be processed for the initial more broadly searching trials. (Also this allows us to start trials, before summaries of all the articles in the collection are computed.)

Thus after the tuning on the first three partitions, the approach will be applied to the fourth partition, or test set, to determine if the results are durable.

2.6.3 Baseline

The fundamental question for these experiments will be how does the performance of MTI on the summary text compare to the performance of MTI on the text from the best performing model. Now to facilitate the trials selecting the optimal section model we used static Related Citations results from submitting the sections separately. If the text in the model were processed normally it would have been submitted all at once to Related Citations. Normal processing of preselected text by MMI is also slightly different due to the larger domain for the normalization of the term scores. Since with the summaries we planning to extract the text first and then process it, we will need to determine the performance of the model text with normal processing.

2.6.4 Experiments

We have developed a document model that selects a set of standard sections that best represent the article for indexing. For summarization we have chosen to include the concepts identified for each sentence as features while summarizing the articles. We look for the optimal summary size and optimal summarization processing to maximize MTI performance. Additional experiments were conducted to explore alternative processing schemes for full text articles.

EXPERIMENTAL VARIABLES

First we looked at summaries of fixed size. In addition we held the compression rate constant and let the size of the summaries vary while maintaining some minimum size for the summaries. Also we performed other tuning experiments by optimizing the values for the dimension reduction ratio.

MTI PROCESSING

There are some variations the MTI processing that may affect the results. If we submit the sections to MMI and Related Citations one at a time instead of submitting all of them together we get different results. If we allow MTI post processing to scan all of the full text instead of just the summary text, it will be more likely to find checktags. We tested these alternatives and used the better performing approach for all the subsequent experiments.

COLLECTION SUBSETS

Additional experiments looked at the performance of several subsets of the collection to see if the summarization of the model text is sufficiently different for some categories of articles. Since the modelling was shown to be more effective for some articles and not others, we looked at the results provided by summarizing the entire article for the subset for which the model summary was less successful.

SECTION MODEL EXTENSION

One weakness of the section model approach is that not all articles fit the frame or set of sections that dominate the collection. So handling like articles regardless of domain separately will probably allow more precise tuning of the model.

Scientific report. The dominant article structure or frame is the standard scientific report with introduction, materials and methods, results, and discussion sections occurs 264 times in the 500 articles. These does not include additional articles that may be included in this set when we allow the simple variants seen in the section classes to which these sections belong. (For example, a materials and methods section might have just “Methods” as a header.

Experiments with this subset of articles included:

- Determined if performance for this subset is better than for the collection as a whole.
- Looked at subsets determined by the size of the articles model.

Non Standard Article. Reviews, editorials, and letters are certainly standard articles but their structure is not that of a scientific report. The subset of the test collection is the set complement of the scientific reports. We considered the following questions:

- Is the performance for this subset below the baseline for the whole collection?
- How does the performance of full text compare to the collection and the model applied to this subset?
- Can we find a better classify these articles? Will short articles perform better with full text while long articles, such as Reviews, perform better with a model based on paragraph position?

Measure the efficacy of summarization on articles with non-standard structure to select the important text.

2.7 Software Implementation

The summarization program, Summarize, is implemented in Java 1.5 as a netBeans project in a package called svd. The key moveable parts are found in `/home/cliff/netbeans/svd/dist/` with svd.jar and the sources are in `/home/cliff/indexing/FullText/java/source/summary/svd/`.

3.0 Experimental Results

The details of the procedures followed in these experiments were dictated by the intermediate results as we progressed. So those details are included here. The variables discussed above were studied and the outcomes of the experiments are reported here.

3.1 The Baseline

As noted above the baseline sets the standard for appraisal of the experiments that follow. However, we need to determine whether the anticipated differences in the MTI processing for these experiments affects the results from the phase 2. So we will first try to duplicate those results with the new experimental setup.

The experimental aspect of this baseline is that it provides a comparison to the results from the initial 4-way partitioned results using the S2SH static data. Those results were based on the best performing model from phase 2 with all the figure and table data in separate sections.This processing also uses 15 citations recommended by Related Citations for each document.

The differences in results shown below should be statistically significant. Confidence interval analysis was not performed on this data, but for similar collections they have been about +-.015->.019.

Table 1 shows the phase 2 results for the partitioned version of the PubMed Central test collection.

TABLE 1. Full Text Indexing by MTI (phase 2)

Collection	Num Articles	True Terms	Precision	Recall	IM Pre	IM Rec	F_2
average	123.50	3548.25	0.3050	0.6013	0.1187	0.8513	0.4894
training	123.50	3548.25	0.3025	0.6025	0.1200	0.8525	0.4893
test	123.50	3548.25	0.3075	0.6000	0.1175	0.8500	0.4894

The average is the average of all 8 partitions.(4 training, 4 test), The training values are the average of all the training partitions. The test value are the averages for just the 4 test partitions -- the official baseline.

Table 2 shows the actual F2 results for each of the document sets side by side.

TABLE 2. Individual Partition F_2 Measures - phase 2 v. production

	Training		Test	
Partition	phase 2	current	phase 2	current
#1	0.4857	0.4615	0.5003	0.4564
#2	0.4921	0.4635	0.4811	0.4505
#3	0.4880	0.4561	0.4933	0.4733
#4	0.4915	0.4599	0.4830	0.4612

Table 3 presents the results when the partitions are processed by MTI as planned for these experiments.

TABLE 3. Full Text Indexing by MTI (current processing)

Collection	Num Articles	True Terms	Precision	Recall	IM Pre	IM Rec	F_2
average	123.50	3280.25	0.3075	0.5512	0.1200	0.8150	0.4603
training	123.50	3280.25	0.3075	0.5500	0.1200	0.8150	0.4602
test	123.50	3280.25	0.3075	0.5525	0.1200	0.8150	0.4604

The phase 2 version of the baseline performance is better. The primary differences in these results are that the recall for the new baseline is about 5% worse, dropping the F2 measure from 0.46 to 0.49. Both sets of indexing were based on the article model from the phase 1 work and used the 15 related citations level established in phase 2. The difference in the processing was that during the previous run each section was submitted to MMI and RelCit as separate documents and those results merged during MTI post processing. For

the new run the sections were selected during extraction from the XML so that the text from the model sections was all sent to MMI and RelCit at once

It is likely that the more focused search by RelCit and the resulting larger set of terms had a positive effect on recall.

Since normal processing is found to degrade MTI performance significantly, then MTI behavior will have to be modified to allow it to recognize sections and process them independently. For these experiments we will submit the sections as separate documents to the indexing paths of MTI but collect all the results together before post-processing.

3.2 Summary Size

3.2.1 Initial Trials with One Partition

Single Document for each Article. The initial summarization trials did not have the modification just described. Rather the whole summary or model was processed as one document by MTI. Table 4 shows the MTI performance for the full model and several different fixed summaries. The summaries are based on the model sections not the complete full text. (Partition One articles were used. The actual files are text.PMC.SMS.1.<n>.out and for the baseline: text.PMC.MS.1.44.0.15.out)

TABLE 4. Performance of MTI with fixed size summaries.

Sentences	Cits	Rec	T	Precision	Recall	#T	IM P	IM R	F2
All	123	3239	955	0.30	0.55	7.76	0.12	0.80	0.4564
17	120	3120	925	0.30	0.55	7.71	0.13	0.82	0.4586
34	122	3198	949	0.30	0.55	7.78	0.12	0.81	0.4604
68	122	3233	960	0.30	0.56	7.87	0.12	0.81	0.4655
136	120	3205	954	0.30	0.56	7.95	0.12	0.82	0.4662

From the partition of 126 articles, Table 5 shows the number of articles affected by summarization for each size target. For partition one, the average size of the model was 87 sentences with the first quartile at 43 and the fourth quartile starting at 125. The plan was to start at 20% of the average article model size so we started with 17.

TABLE 5. Affect of summarization:

Size	Affected Articles
17	107
34	100
68	72
136	25

A study of how individual articles were affected by the condensation to 34 sentences showed the following:

- 21% of the articles got more correct terms and 27% got fewer.
- For those changed the average increase was 1.8; the average decrease 1.4.
- 33% or the articles got higher F_2 measure scores, and 30% got lower.
- For those whose F_2 measure changed, the average increase was 0.072; the average decrease was 0.065.

Multiple Documents for each Articles. We also ran trials with the separate sections presented to MTI as distinct documents, and doing the post processing on the complete document. In the summarization case we generated separate section documents for the selected sentences appearing in each section.

Table 6 shows the difference for one summary size (34) the dramatic effects of multiple document processing and the use of the full text (not just the model sentences for looking up check tags and other post processing. (Actual files: text.PMC.MDM.1f.34.out text.PMC.MDM.1.34.out)

TABLE 6. MTI Processing Options

MTI Processing	F2
model sections only - single document	0.4604
model sections only - multiple documents	0.4601
model sections + full text - multiple documents	0.4886

Because this style of processing with MTI produces a much better performance it was used for the remainder of the experiments. The set of fixed sized summary trials with this processing is shown in Table 7 . The optimal summary size is 85 sentences which meant

TABLE 7. Model Sections as Multiple Documents and Full Text

Sentences	F2
17	0.4860
34	0.4886
68	0.4939
77	0.4956
85	0.4975
93	0.4969
102	0.4940
136	0.4915
ALL	0.4911

that 55 of the 123 articles were actually summarized.

For the different partitions there were differing optimal summary sizes, such as 93, or 102.

3.2.2 Full Collection Optimization

It was recommended by another researcher that we not use a four fold cross validation for tuning the size of the summaries, but tune directly on the whole collection since we had a gold standard for evaluating the indexing. Therefore we ran full collection trials for the three values found in the partition trials. The results for the summarization study are shown in Table 8 .

TABLE 8. Performance for Full Collection with Fixed Size Summaries

Sentences	Cits	Rec	T	Precision	Recall	#T	IM P	IM R	F2
85	494	14223	4350	0.31	0.6	8.81	0.11	0.84	0.4924
93	494	14212	4351	0.31	0.6	8.81	0.11	0.84	0.4925
102	494	14199	4342	0.31	0.6	8.79	0.11	0.84	0.4917

So the best fixed summary size for the full collection is 93 sentences where the F2 measure is 0.4925. Computing the 90% confidence interval (10,000 samples) we get a lower bound of 0.48221 and an upper bound of 0.50269. The experimental result is just 0.0032 over the baseline (0.4893) which is within the confidence interval.

These are all based on the submitting separate documents to MTI for each section and pulling the initial mappings from each indexing path together during post processing and using the full text for geo and checktag lookup. This summarization scheme does not produce significant improvement in MTI performance on these articles.

3.3 Compression Ratio

The experiments reported below are for summaries that have the same compression rate. The summaries studied before had a constant size as measured by the number of sentences. The trials were run only on partition 4 because it has the average article length closest to the whole collection. The baseline for these experiments is the best performing size for this partition which was at 102 sentences.

The trials are for different compression rates applied to this same partition. Table 9 shows the results when the size of the summary depends on the size of the article model. The first two rows show the comparible results for best static size and for the whole model (without any summarization). For the meaning of the column headers see the legend below. Again this different way of selecting sentences for the summary has no significant effect on the results.

Legend:

A number of recommendations

B number of matching terms

C number of matching IM terms

D precision

E recall

F average number correct

G F2 measure

TABLE 9. Compression Rate affect on MTI Performance with Summaries

Size	A	B	C	D	E	F	G
102 sentences	3648	1140	407	0.31	0.59	9.05	0.4876
all sentences	3642	1139	408	0.31	0.59	9.04	0.4871
50%	3638	1108	399	0.03	0.57	8.79	0.4739
60%	3635	1125	405	0.31	0.58	8.93	0.4817
70%	3635	1141	409	0.31	0.59	9.06	0.4880
80%	3633	1140	409	0.31	0.59	9.05	0.4876
90%	3638	1138	408	0.31	0.59	9.03	0.4868

3.4 Dimension Reduction Ratio

The compression rate and summary size have to do with how many sentences to include in the summary. The dimension reduction ratio affects the concentration of the semantic structure hopefully revealed by the SVD process. These factors are independent so we will use the same baselines for evaluating the Dimension Reduction Ratio (DRR) trials.

The baselines are the full model (all sentences) and the best performing summary length (102 sentences) with a DRR of .50. For these baselines and the trials with each section was submitted as a separate document to the indexing paths.

Because it has a mean article model size closest to the mean model size of the whole collection, partition 4 with 126 articles was used for these trials. The articles of partition 4 were summarized to a maximum of 102 sentences using the DRRs from 0.40 to 1.0. Table 10 shows these results along with the baselines. Note that the columns have the same meaning as in the previous section.

The best performing DRR of 1.0 produces only a .5% improvement over the all sentences

TABLE 10. Dimension Reduction Ratio affect MTI Performance on Summaries

DRR	A	B	C	D	E	F	G
102 sentences	3648	1140	407	0.31	0.59	9.05	0.4876
all sentences	3642	1139	408	0.31	0.59	9.04	0.4871
.40	3649	1141	407	0.31	0.59	9.06	0.4874
.50	3648	1140	407	0.31	0.59	9.05	0.4876 (baseline)
.60	3648	1140	407	0.31	0.59	9.05	0.4876
.70	3647	1142	407	0.31	0.59	9.06	0.4884
.80	3647	1144	407	0.31	0.59	9.08	0.4890
.90	3647	1144	408	0.31	0.59	9.08	0.4892
1.00	3647	1145	408	0.31	0.59	9.09	0.4894*

baseline.

The best performing compression rate was 70% with a minimum size of 50 sentences: 0.4890. So the 1.0 DRR gives the best performance of any of the summarization options tested.

However, because these initial results are not significant, DRR trials for the full collection of 500 articles were not run.

3.5 Non-standard articles

Since the model develop in the orginal study of articles with full text seems to be biased towards articles with the section headers included in the model, we looked at summarization as perhaps a way to identify important text in articles that did not have the standard outline, i.e. non-standard articles.

When the document outlines considered standard are expanded to the set of headers in the section classes used during modeling, we end up with 309 articles. So section headers previously considered semantically equivalent to the literal standard headings were allowed when selecting standard articles. (So “Materials and Methods” was considered a standard header along with “Methods.”) This selection process left 185 articles with Medline indexing that we are calling "non-standard." This subset is 37% of the test collection.

3.5.1 Non-standard baseline

For our look at non-standard articles, we established some baselines.

The non-standard subset of articles with the optimal model for the whole collection has an F_2 of 0.4852. This is 0.0041 below the collection average.

By comparison the standard articles subset has an F_2 of 0.4918 that is slightly above the average (+0.0025). This means that there is a +0.0066 gap between the two subsets. This gap suggests that the non-standard subset might do better with different handling.

When the full text of the non-standard subset is processed the F_2 performance is 0.4386. Thus the current model provides a +0.0466 increase for this subset over the full text.

3.5.2 Non-standard Article Summaries

These trials to select an optimal summary for the non-standard articles looked at the same sort of variation in summaries used on the full collection.

TABLE 11. MTI performance on Summaries of Non-standard Articles

Processed Text	F_2 Measure
Compression Ratio	
70% min 50	0.4774
60% min 50	0.4773
60% min 10	0.4785
50% min 50	0.4816 *
40% min 50	0.4801
20% min 10	0.4767
10% min 10	0.4656
Fixed Size	
85	0.4761
102	0.4759
Dimension Reduction Ratio	
CR 50% min 50, DRR=1.0	0.4781
Baselines	
All sentences	0.4386
Section Model	0.4852

The range of compression rates explored yielded a maximum F_2 measure at 50% with a minimum summary size of 50. This was a still 0.0036 below the section model baseline, but shows an equivalent improvement over the unmodified full text. Attempts to tweak by reducing the minimum summary size did not suggests enough improvement to reach the baseline. Previously successful fixed sizes were tried, but were far short of the 50% compression rate score. The other results above all have DRR= 0.5. For whole collection, performance was better with DRR = 1.0 than for DRR = 0.5. This was not the case for this subset.

The final and suprising conclusion is that even for the non-standard articles the section model out performs the summary. This reconfirms the importance of the abstract and the figure and table captions which was usually the only sections the non-standard articles had to use with the section model.

4.0 References

1 [SCHUEMIE, WEEBER]

M. J. Schuemie *, M. Weeber, B. J. A. Schijvenaars, E. M. van Mulligen, C. C. van der Eijk, R. Jelier, B. Mons and J. A. Kors. Distribution of information in biomedical abstracts and full-text publications. *Bioinformatics* 2004 20(16):2597-2604; doi:10.1093/bioinformatics/bth291

2 [YEH, KE, YANG & MENG}

Yeh JY, Ke HR, Yang WP, Meng, IH. Text summarization using a trainable summarizer and latent semantic analysis. Information Processing and Management 41 (2005) 75-95.

3 [KO, PARK & SEO]

.Ko Y, Park J, Seo J. Improving text categorization using the importance of sentences. Information Processing and Management, 2004; 40: 65-79.

4 [GONG & LIU]

Gong Y, & Lui X. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (SIGIR'01). 19-25 New Orleans, LA, USA.

5 [LI, LU, SHI]

Li Z, Lu X, and Shi W. Process Variation Dimension Reduction Based on SVD. In *Proc. IEEE International Symposium on Circuits and Systems*, IEEE. (2003) URL: ece.tamu.edu/~wshi/pub/iscas03.pdf.

6[GOLDSTEIN, KANTROWITZ, MITTAL, & CARBONELL]

Goldstein J, Kantrowitz M, Mittal VO, Carbonell J. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In *Proceedings of SIGIR-99*, Berkeley, CA, August 1999. ACM 121-128.

7 [BELLEGARDA, BUTZBERGER, CHOW, COCARRO, & NAIK]

Bellegarda, JR, Butzberger JW, Chow YL, Coccaro NB, Naik D. A novel word clustering algorithm based on latent semantic analysis. In *Proceedings of the 1996 international conference on acoustics, speech, signal processing (ICASSP'96).* Atlanta, GA, USA.172-175.

Kim JH, Kim JH, Hwang D. Korean text summarization using an aggregative similarity. In *Proceedings of the 5th international workshop on information retrieval with Asian languages* (IRAL). 111-118. Hong Kong, China.

Wessa, P. (2005), Free Statistics Software, Office for Research Development and Education, version 1.1.17, URL http://www.wessa.net/