Experiments in Spoken Document Retrieval at CMU

M. A. Siegler, M. J. Witbrock, S. T. Slattery*
K. Seymore, R. E. Jones, and A. G. Hauptmann
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890

*Justsystem Pittsburgh Research Center
4616 Henry St.
Pittsburgh, PA 15213

Abstract

We describe our submission to the TREC-6 Spoken Document Retrieval (SDR) track and the speech recognition and the information retrieval engines. We present SDR evaluation results and a brief analysis. A few developments and experiments are also described in detail including:

- Vocabulary size experiments, which assess the effect of words missing from the speech recognition vocabulary. For our 51,000-word vocabulary the effect was minimal.
- Speech recognition using a stemmed language model, where the model statistics of words containing the same root are combined. Stemmed language models did not improve speech recognition or information retrieval.
- Merging the IBM and CMU speech recognition data. Combining the results of two independent recognition systems slightly boosted information retrieval results.
- Confidence annotations that estimate of the correctness of each recognized word. Confidence annotations did not appear to improve retrieval.
- N-best lists where the top recognizer hypotheses are used for information retrieval. Using the top 50 hypotheses dramatically improved performance in the test set.
- Effects of corpus size on the SDR task. As more documents are added to the task, the gap between perfect retrieval and retrieving spoken documents gets larger. This makes it clear that the size of the current TREC SDR track corpus is too small for obtaining meaningful results.

While we have done preliminary experiments with these approaches, most of them were not part of our submission, since their impact on the IR performance on the actual TREC SDR training corpus was too marginal for reliable experiments.

1. The SDR Data and Task

The speech data for the 1997 TREC spoken document retrieval track consisted of about 70 hours of broadcast news mostly from CNN and NPR shows. The data had been segmented into stories and manually transcribed. There were three "versions" of the data available: A manually generated transcript (which also contained some errors), a speech recognition transcript provided by IBM, and the raw audio data, to be transcribed by our own recognizer. About 35 hours of this corpus was classified as training data, which we used to train the Sphinx-III speech recognition system. The remainder was held out as unseen test data. There were about 1200 stories in the training data set and 1451 in the test set. To develop and debug the system, there were 5 training queries available and the test data consisted of 49 queries.

Scoring Metrics

The test queries were designed to simulate a *known item retrieval* task. For each query, there was only one document considered relevant for the purposes of this evaluation. While other documents may have some relevance to the query, only the document it was designed to retrieve was scored as a correct retrieval. To reflect the nature of this task, we used the following metric:

Inverse Average Inverse Rank (IAIR)

$$\text{IAIR} \equiv \frac{1}{\sum_i \left(rank_i^{-1} \right)}$$

Where $rank_i$ is the rank of document i

One characteristic of the IAIR is that it rewards correct documents near the top more than documents in the middle or towards the end of the rankings. In our opinion this is a reflection of desired behavior in an IR system, and we used the metric exclusively in our analysis.

Idiosyncrasies of Known Item Retrieval

One of the idiosyncrasies of the known item retrieval paradigm is that only one document is defined to be relevant to the query. Therefore, it is in the interest of the IR system to maximize the score for this document, rather than maximize the overall number of relevant documents retrieved. As a consequence, we found that query expansion did not produce a better IAIR score. In addition, the IR system performed better when as many of the query terms as possible appeared in the correct document, despite the presence of erroneously recognized query terms in the incorrect documents. Generally, known item retrieval seems to favor the detection of correctly identified query terms over the rejection of falsely identified query terms and this is demonstrated in our experiments below.

2. System Overview

In this section we give a system description of the actual CMU TREC-6 SDR submission. The speech recognition system is outlined as well as a fully automatic information retrieval weighting scheme suitable for retrieving documents transcribed (with errors) by automatic speech recognition.

The Speech Recognition Component

The Sphinx-III speech recognition system was used for the CMU TREC SDR evaluation, and it was configured similar to the 1996 DARPA CSR evaluation [10], although several changes have been made since then. Sphinx-III is a large vocabulary, speaker independent, fully continuous hidden Markov model speech recognizer with separately trained acoustic, language and lexical models.

For the current evaluation a gender-independent HMM with 6000 senonically-tied states [5] and 16 diagonal-covariance Gaussian mixtures was trained on a union of the CSR Wall Street Journal corpus and the 1996 TREC-6 training set.

The decoder used a Katz-smoothed trigram language model trained on the 1992-1996 Broadcast News Language Modeling (BN LM) corpus. This is a fairly standard language model, much like those that have been used in the DARPA speech recognition community for the past several years. As a space optimization singleton trigrams and bigrams were excluded. As a new feature, this language model incorporated cross-sentence-boundary trigrams to better model utterances containing more than one sentence.

The lexicon was chosen from the most common words in this corpus, and to be a size that balances the trade-off between leaving words out-of-vocabulary and introducing acoustically confusable words [9]. For this evaluation, the vocabulary was comprised of the most frequent 51,000 words in the BN LM corpus, supplemented by some 200 multi-word phrases and some 150 acronyms. The vocabulary size was initially based on our experience with broadcast news, and a subsequent careful analysis of the trade-offs showed that our choice was a very good one. More details of the trade-off involved in vocabulary selection are provided below.

In contrast to the earlier Sphinx-II speech recognition system, Sphinx-III boasts a higher accuracy but at significant cost. To achieve a lower word error rate of 27.4% versus 45.9% for Sphinx-II on a subset of the training data, the original Sphinx-III system processing time increased to 120 times real time on a 266 MHz DEC Alpha compared with only 1.4 times real time for Sphinx-II. By reducing the beam width of the search and optimizing the space required, we reduced the Sphinx-III processing time to about 30 times real time, with only a slight loss in word transcription accuracy. Decoding the audio files in the test data thus required about 1000 hours of CPU time.

The Information Retrieval Component

Both documents and queries were processed using the same conditioning tools, namely noise filtering, stopword removal, and term stemming:

- Noise Filtering: The goal of noise filtering was simply to remove non-alphabet ASCII characters, punctuation, and other junk considered irrelevant to IR. All punctuation was removed except for spelled-letter words, e.g. "C. M. U," and the use of the apostrophe for contractions, e.g. "CAN'T." Any changes in case were removed.
- Stopword removal: A set of 811 stopwords was compiled from a combination of the SMART IR engine and several selected by hand based on document frequency. These words were removed entirely.
- Term mappings: A set of 4578 mappings was used to map words with irregular word endings that were not properly covered by an implementation of the Porter [7] algorithm. An on-line Houghton-Mifflin dictionary was used for this lookup of irregular words and their roots.
  An example of this mapping is `APPENDICES→APPENDIX`
- Term stemming: An implementation of the Porter algorithm was applied to map words to their common root.

A heavily stripped down core of the CMU Informedia SEIDX engine was used to compare queries with documents. A relevance score was created for each pair according to the following equation:

$$\text{Relevance Score} \equiv \frac{\sum_i (qtf_i * dtf_i * \log(idf_i))}{\sqrt{\sum_i dtf_i^2}} * \left[1 + \left(\sum_i \text{sign}(qtf_i * dtf_i)\right)^2\right]$$

qtf_i	Query term frequency for vocabulary word *i*
dtf_i	Document term frequency for vocabulary word *i*
idf_i	Inverse document frequency for vocabulary word *i*
sign	Sign of value function (0 if 0, 1 if positive)

3. Official TREC-6 SDR Results

Table 1 shows the official CMU TREC SDR results. Since the transcriptions were subject to filtering as discussed above, the word error rates are reported for both the unfiltered and filtered references and hypotheses. An analysis of the results showed several preprocessing errors and confirmed an insight into the relationship between word error rate and information retrieval.

Transcription Source	WER		IAIR
	Unfiltered	Filtered	
Reference	0	0	1.35
CMU-SR	35.5	26.4	1.44
IBM-SR	45.6	47.4	1.64

Table 1: Performance of the CMU TREC-6 SDR Evaluation System

Vocabulary Coverage

The words that were in the queries but were missing from the speech recognizer's 51,000 word vocabulary were "CIA", "TORCHED?", "SMOKING?", "WELL_KNOWN", and "GOLDFINGER". These problems are primarily due to inconsistencies in the preprocessing phases. While "C.I.A." was in the vocabulary, "CIA" was not, resulting in a completely missed word during information retrieval. Similarly, an oversight in the preprocessing phase allowed the question mark to become part of the word in "torched?" and "smoking?". For "well-known", each of the component words "well" and "known" were in the vocabulary, but the compound "well-known" was not there as a single token, and thus was treated as an irretrievable word. The only true missing word in our 51,000-word vocabulary was "Goldfinger". Thus the 51,000-word vocabulary selection provided excellent coverage for this test evaluation.

Recognition Accuracy versus Information Retrieval Quality

The official results confirm that vastly reduced word errors rate translates into slight improvements in information retrieval. Comparing the performance on the IBM speech recognition data to the CMU speech recognition, on the filtered texts, we find that nearly **doubling** the word error rate led to only a 14% decrease in information retrieval quality.

4. Experiments

In order to create meaningful experiments with the TREC-6 training data, 1167 documents were selected from the set and headlines were generated for 374 of them by hand. In addition, a much smaller test set composed of 103 broadcast news stories from a privately collected corpus was acquired to investigate ideas involving the speech recognition configuration. We shall refer to this latter test set as the "small test set."

4.1. Vocabulary Size Experiments

Prior to the evaluation we attempted to find a good vocabulary size that was optimized for both speech recognition and information retrieval. We chose three different vocabulary sizes, 40,000, 51,000 and 64,000 words, constructed a language model for each one, and then performed speech recognition. Table 2 shows that as the vocabulary got larger, the rate of out-of-vocabulary words decreased, but beyond 51,000 words speech recognition accuracy did not improve. Additional vocabulary coverage was thus obtained at the cost of adding many acoustically confusable words, and information retrieval effectiveness decreased slightly. We chose to use the 51,000-word vocabulary for our official submission, resulting in only one query word in the final 49 test to be missing.

Vocabulary Size	Out Of Vocabulary Rate	Word Error Rate	IAIR
40k Words	1.13 %	26.4 %	1.24
51k Words	0.83 %	26.8 %	1.21
64k Words	0.75 %	26.8 %	1.22

Table 2: Effect of Vocabulary Size on System Performance.

4.2. Stemmed Language Models

Using a small test set described above and the 51,000-word vocabulary, we also investigated the concept of language modeling tailored specifically to information retrieval. Since the words in the recognition output are filtered, a language model was built from a stemmed version of the LM training data. Each root word in the language model had multiple pronunciations to reflect the original words before filtering. Others have used this technique to improve language modeling when the vocabulary is open-ended or indeterminate [3].

For example, suppose the root forms of the words "`recognize`", "`recognized`", and "`recognition`" all map into the common root "`recogni`"+suffix, where suffix in this case is either "`ze`", "`zed`", or "`tion`". The stemmed language model would provide only one transition from the root "`recogni`" into words that can follow, in effect collapsing multiple paths between individual words into one path between root words. The lexicon would reflect the alternate original words as alternate pronunciation of the root word, i.e.

```
Recogni          R EH K AX G N AY Z
Recogni(2)       R EH K AX G N AY Z DD
Recogni(3)       R EH K AX G N IH SH AX N
```

The premise was that this stemmed language model would avoid much of the confusion due to acoustic variations in suffixes of words, but would aid in the correct recognition of the important roots of the words. Table 3 shows the results of these experiments. The word error rate of the stemmed language model was higher than for the baseline language model. The WER increased both if only stemmed words were counted, as well as when all original words were compared. Furthermore the information retrieval effectiveness (as measured by the inverse average inverse rank metric) also showed a decrease.

Language Model	Word Error Rate		IAIR
	Unfiltered	Filtered	
Baseline	26.8 %	22.6 %	1.17
Stemmed	35.1 %	23.8 %	1.25

Table 3: Using a language model built from stemmed LM training texts.

4.3. Merging Multiple Sources of Speech Recognition Data

Since the IBM speech recognition system was developed independently of the CMU system, and it used different training data, vocabulary, and language models, it occurred to us that a combination of the two speech recognition transcripts might allow some randomly distributed errors to be recovered. Instead of mixing the recognition outputs, we formed a weighted relevance score in the following way:

$$Score_{MIX} \equiv Score_{CMU} * \lambda + Score_{IBM} * (1 - \lambda)$$

$Score_{CMU}$	The relevance score using the CMU recognition output
$Score_{IBM}$	The relevance score using the IBM recognition output
λ	An interpolation weight

Results on the TREC-6 testing set are shown in Table 4, showing a slight reduction of retrieval error when the CMU weight is 0.8 and the IBM weight is 0.2. Thus multiple recognizers, even with widely varying word error rates, can be combined to improve information retrieval performance.

CMU Weight	IBM Weight	IAIR
1.0	0.0	1.382
0.9	0.1	1.379
0.8	**0.2**	**1.375**
0.7	0.3	1.395
0.6	0.4	1.394
0.5	0.5	1.421
0.4	0.6	1.467
0.3	0.7	1.462
0.2	0.8	1.467
0.1	0.9	1.548
0.0	1.0	1.581

Table 4: Results of merging relevance from separate recognition systems.

4.4. Confidence Annotation

Since state-of-the-art speech recognition software does not produce a perfect transcript of what was said, we would like to obtain any extra information we can about the likelihood of correctness of particular words. This is akin to a human annotator guessing a mumbled word and indicating a possible transcription error.

An ideal automatic confidence annotator would label each word produced by the speech recognizer with a label *correct* to indicate that this is in fact the word that was spoken, and *incorrect* to indicate that this word was not spoken. We will compare the results of our annotation to this ideal, which we call Perfect Annotation.

Features for Confidence Annotation

The confidence annotation we performed is based on work by Lin Chase [1], though annotation has been explored by many others including [2][3][4]. Typically confidence annotation is performed by taking information available about individual occurrences of words in the hypothesized text, from information produced within the speech recognizer, or outside the recognizer. These features are then automatically examined to find indicators of likely correctness and incorrectness.

The candidate features we considered were:

- *Acoustic Score*. This is the score the speech recognizer assigns the word based the probability that the acoustics observed were generated by the hypothesis.

- *Language Model Score*. This is a score assigned by the speech recognizer, based the probability that the word is to occur given the previous two words.

- *Duration*. This is the duration of the word, and helps offset the duration dependence of the acoustic score.

- *N-best Homogeneity*. The N-best list is the list of the best n guesses at the words spoken in the document, sorted according to a weighted combination of acoustic and language model scores. A word appearing in our hypothesis may appear in many or few of the competing hypotheses. N-best list homogeneity is the proportion of hypotheses that the word appears in. We set n to 200 for the confidence annotation experiments.

Experimental Description - Confidence Annotation

For each set of features, the experiment proceeds as follows:

- Label all words in training set as *correct* or *incorrect*[1] by comparing them to the words in the words in the reference transcript

- Build a decision tree that finds sets of features that perform well in distinguishing between *correct* and *incorrect* words in speech recognition hypotheses.

- Use decision tree to test features of words in test set. Once a word has been sorted into a leaf node, the proportion of correct and incorrect words from the training set with these features is used to calculate an approximate probability of correctness

- Perform information retrieval by weighting each word according to the probability that it is correct (the *confidence*).

We conducted experiments by splitting the training data into two sections, training our decision tree on one half, testing on the other half, then reversing the roles.

Decision Tree Building

The decision tree building algorithm we use is C4.5 [8]. It functions by taking all training data, and attempting to find rules based on features which distinguish between classes. Each item of training data is a word along with its associated features (described above), and its class of *correct* or *incorrect*. Taking each feature does this in turn, asking a question about that feature, and using the answer to partition the data. A feature is chosen if it has high information gain, i.e. if the resulting two groups of data contain less of a mix of *correct* and *incorrect*. The ideal split would create classes that contain exclusively *correct* or exclusively *incorrect* examples.

Since such ideal splits are rare, the decision tree building halts when no more information gain (reduction in entropy) can be achieved. At this point, each leaf of the tree contains examples which have all the same features for questions asked at each partition, and which are mostly of one class. The proportion of *correct* examples at this node is the probability of correctness that will be assigned to any word with the same features.

When using the decision tree to classify a new word, we check each of its features to find which leaf-node of the decision tree to classify it into. At that point, it is classified as having the probability of correctness corresponding to this leaf node.

Evaluating Confidence Annotation: Cross-Entropy Reduction

The most common method of evaluating word confidence annotation is cross-entropy reduction. Cross-entropy is a measure of how well our model of the probability of word correctness corresponds to Perfect Annotation (as defined above). If our model annotates perfectly, its cross-entropy is 0. The worse the annotation performs the higher the cross-entropy.

---

[1] incorrect words are all insertions and substitutions in the hypothesis

The most naive form of confidence annotation we can perform is to tag each word with a probability of correctness equal to the overall word-accuracy. Thus if we know that our recognizer generally gets 80% of words correct, the baseline confidence annotator assigns each word an 80% probability of correctness. We then measure the quality of our annotation by measuring how much better it performs than this baseline.

$$CrossEntropy \equiv \frac{1}{n}\sum_{i}^{n} P(w_i) * \log_2 \frac{1}{Q(w_i)}$$

$P(w_i)$ The actual probability that word *i* is incorrect

$Q(w_i)$ The probability that word *i* is incorrect as predicted by the annotation

Thus we attain a figure for cross-entropy for the default model of classifying each word as correct with probability equal to the word-accuracy, and score our improvements in modeling the probability of correctness by how much they reduce cross-entropy as a percentage of this baseline.

Information Retrieval Using Word Confidence Weights

First we describe two orthogonal ways of using word confidence weights in the relevance scheme described above:

- **Expected Term Frequency** (ETF): The ETF is an estimate of how many times the term actually occurred given the number of observations. Assuming independent observations, this is a sum of the probability of a word being correct over each instance.

- **Expected Inverse Document Frequency** (EIDF): To calculate EIDF, we first calculate the probability that this word occurs somewhere in the document, for each document:

$$\begin{aligned} P(w \in d) &= 1 - P(w \notin d) \\ &= 1 - \prod_{i}^{n} P(w \neq w_i) \\ &= 1 - \prod_{i}^{n} [1 - P(w = w_i)] \end{aligned}$$

Since typically, $P(w = w_i)$ is very small when $w \neq w_i$, we only take the product over terms for which the recognized word was *w*. Summing this value over all documents and dividing by the total number of documents gives us an approximate value of the expected document frequency for this word

Oracle Experiments

Since the interaction between confidence annotation and information retrieval may be complex, we also conducted an experiment to see how we could make use of confidence scores in the idealized case in which we know exactly which words are correct, and which are incorrect. We removed words in two different ways:

- Pre-filter: Before the hypothesis is filtered, all the words that are not found in the reference are removed.
- Post-filter: After the hypothesis is filtered, all the words that are not found in a filtered version of the reference are removed

Table 5 shows that for both training and testing sets, the Post-Filter Oracle annotation was able to significantly reduce the IR error of the decoded transcripts. This indicates that a more realistic experiment might be able to do this as well.

We performed an analysis of some of the differences between documents in the stemmed oracle experiment, and reference information retrieval experiments. We should expect the number of query words in the correct document to decrease, since oracle confidence annotation cannot *correct* for substitutions and deletions, but will drop all incorrectly substituted and inserted words. A cursory glance at documents and queries revealed that some documents contain **more** query words as speech hypotheses then the corresponding reference transcription. Our intuition here is that speech recognition can occasionally correct for spelling errors in the references, and so words that are incorrect with respect to the reference transcription may be correct for the purposes of information retrieval.

	Baseline Performance		Oracle Annotation	
	Reference Transcripts	Speech Transcripts	Pre-Filter	Post-Filter
Training Set	1.233	1.283	1.285	1.269
Testing Set	1.332	1.382	1.374	1.338

Table 5: Baseline and Oracle Annotation on TREC-6 Training and Testing Sets. Values are IAIR

Information Retrieval Experiments for Confidence Annotations

In order to see how well cross-entropy reduction translates into gains in information retrieval accuracy, we conducted a series of experiments. Since we also hoped to find the best way of incorporating weights into information retrieval we performed the following information retrieval experiments:

- **ETF**: for this experiment, we used ETF, and regular IDF.
- **EIDF**: for this experiment, we used EIDF, and regular TF.
- **ETFIDF**: we use both ETF and EIDF

	Pre-Filter			Post-Filter		
	ETF	EIDF	ETFIDF	ETF	EIDF	ETFIDF
Training set	1.276	1.283	1.277	1.273	1.281	1.274
testing set	1.378	1.383	1.399	1.381	1.382	1.382

Table 6: Confidence Annotation Performance on TREC-6 Training and Testing Sets. Values are IAIR.

The results of these experiments are found in Table 6. Although the IAIR was reduced in most cases, the upper bound found in the Oracle Annotation was not attained.

4.5. Using N-best Lists for Information Retrieval

Typically, speech recognition systems produce a transcription of each spoken utterance in much the same way that a human transcriber might. However, the transcription used is only the most probable decoding of the acoustic signal, out of a large number of hypotheses that are considered during the recognition process. It is a relatively simple matter to obtain a list of these different hypotheses, ranked in order of decreasing likelihood.

Using these additional hypotheses seems promising for information retrieval, since it offers the hope of including terms that would otherwise be missed by the speech recognizer in documents, allowing them to match with query terms and increase document recall. On the other hand, words incorrectly identified in lower ranked recognition hypotheses may cause spurious matches with query terms, decreasing retrieval precision.

Experiments Using N-Best Lists

In the context of the TREC-6 SDR task, an initial attempt was made to evaluate retrieval effectiveness using N-best hypotheses lists generated from the speech recognition decoder lattice. N-Best hypotheses were generated for the 1451 stories in the TREC-6 SDR test data. Of these, decoding failed completely in four cases, resulting in empty transcriptions. For the remaining 1447 stories, lists of the two hundred most likely hypotheses were generated for each utterance. Table 7 shows an example of N-best hypotheses.

Ideally, one would use hypothesis probabilities generated during decoding to weight the terms during retrieval, but for this preliminary experiment, the N hypotheses for each utterance were simply concatenated together into one larger document. No discounting of weights for less probable hypotheses was done.

N	Nth most likely decoder hypothesis
1	HATE FAIR ADEQ EDUC CHILD WITHSTAND CALM
2	HATE FAIR ADEQ EDUC CHILD WITHSTAND COMMON
3	HATE FAIR ADEQ EDUC CHILD WITHSTAND INTERCOM
4	HATE FAIR ADEQ EDUC CHILD WITHSTAND CALM

Table 7: The top four hypotheses for utterance three of story j960531d.7, after stop word removal and stemming. Note that the fourth hypothesis is identical to the first, and differed only in inflected forms.

The effect on retrieval effectiveness of using the documents generated from the N-best lists in the TREC-6 test set is illustrated in Table 8. Note that for N set to 50, the performance on the hypothesized transcripts is actually slightly lower than performance on the reference transcripts (1.332) This may be again due to effects of misspellings in the reference transcripts.

Number of Hypotheses (n)	1	2	5	10	20	50	100	200
IAIR	1.368	1.353	1.366	1.365	1.367	1.317	1.320	1.325

Table 8: IR Performance of N-Best hypotheses on the TREC-6 test set.

While it is encouraging that an improvement in retrieval can be obtained at all by this method, it is clear that further work will be required if the promise of this idea is to be realized. In particular, the increasingly harmful effect of adding large numbers of less probable hypotheses to the documents suggests that discounting each hypothesized word by its recognition score may improve performance even more.

4.6. Scaling The Collection Size

Many of our experiments, including some of the ones reported here, seem to suffer from two problems. The effect size of our experimental variables seems to be fairly small, and the difference between the reference text retrieval and the speech recognition transcript retrieval is only a few percent of the inverse average inverse rank. If this relationship holds even as we scale to larger, more realistic, and more useful collections, then we can consider the problem of spoken document retrieval practically solved to within a few percent of perfect text retrieval effectiveness.

To test this hypothesis using the TREC-6 training set, we increased the number of text documents in the corpus up to 14,000 and measured the inverse average inverse rank for the same retrieval queries. However, instead of actually performing speech recognition on the added documents, artificially degraded texts were used. In this case, the degradation method attempted to only model word errors through deletion of query words. Although a primitive model of speech recognition errors this may represent an upper performance bound.

Figure 1 shows the relationship between the inverse average inverse rank information retrieval performance and the size of the document collection. As more documents are added to the collection, the gap between the reference (perfect text) retrieval and the speech recognition based retrieval grows. At collections larger than 10,000 documents the gap starts to widen significantly. We can expect to experience larger discrepancies between speech transcribed and perfectly transcribed documents, which may make spoken document recognition unusable for collections numbering in the 100,000 or larger.

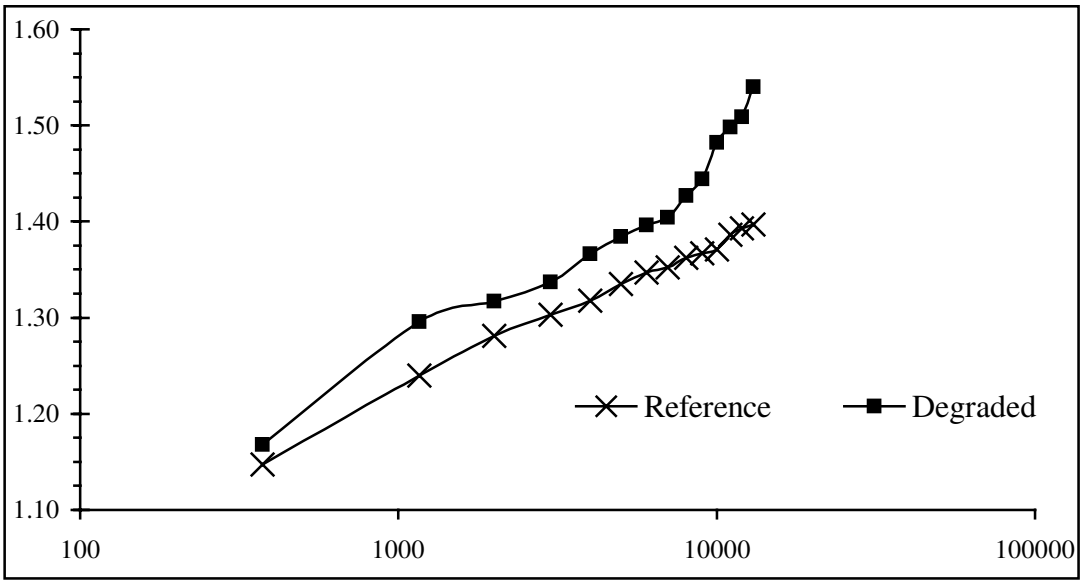

Figure 1: Effect of collection size on IR performance of the TREC-6 training set with reference and artificially degraded documents. The X Axis is the number of documents used in the analysis, and the Y Axis is the IAIR.

5. Summary

There are several conclusions we can draw based on our experiments:

- First of all, we have found that even large reductions in speech recognition word error rate result only in small information retrieval improvements. On the converse side, the quality of information retrieval is a lot higher than the speech recognition word error rate figures would indicate. Despite fairly high word error rates, information retrieval performance was only slightly degraded for speech recognizer transcribed documents.

- Stemmed language modeling did not help speech recognition or information retrieval.

- A 51,000 vocabulary covered the range of words used in the queries quite well. Only one query word was truly outside of this vocabulary.

- We could expect better performance on the reference texts if better IR weighting schemes and pre-processing functions were used. These improvements would probably also result in small gains in the speech corpus, although we have done no studies.

- Confidence Measures provide no benefit. Even an oracle confidence measure, which can reliably single out the correctly recognized words and discard all the other words provides only a small increase in retrieval effectiveness (as measured in IAIR). This points to the conclusion that deleted (missing) words are most critical, while inserted words do not affect the retrieval in the same proportion.

- Since deleted (missing) words are critical to the retrieval effectiveness, one can try to reduce this by adding probable words from the speech recognizer hypothesis N-best list. Using the N-best list to augment the speech recognition output with likely words shows great promise. Our experiments indicate that this approach might drastically reduce the difference between perfect text transcripts and speech recognizer generated transcripts.

- Merging the results from multiple independent speech recognizers may also improve IR effectiveness.

In general, most of our findings are very preliminary. While we believe we may have uncovered trends, there is too little data for conclusive experiments. As a result, we did not conduct significance tests to measure the practical effects of the observed trends since the TREC-6 SDR track provided too little data for definitive experiments. Furthermore, the difference between the speech recognizer generated transcripts and the perfect text transcripts was too small in this corpus. However, the experiments we have done on increasing the scale of these document collections by orders of magnitude leave a worrisome fear that the initially promising results for SDR will not hold up in larger data sets.

We have viewed this participation in the TREC-6 SDR track as a learning experience, which will guide both our own research as well as the design of future SDR track evaluations.

References

[1] L. Chase, *PhD thesis*, Carnegie Mellon University Robotics Tech Report, 1997.

[2] S. Cox and R. Rose, "Confidence Measures for the Switchboard Database," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1996.

[3] P. Geutner, "Using Morphology Towards Better Large-Vocabulary Speech Recognition Systems," *IEEE International Conference on Acoustics, Speech and Signal Processing, 1995*.

[4] L. Gillick and Y. Ito, "Confidence Estimation and Evaluation," *LVCSR Hub-5 Workshop Presentation*, 1996.

[5] M-Y. Hwang, "Subphonetic Acoustic Modeling for Speaker-Independent Continuous Speech Recognition". PhD Thesis, CMU-CS-93-230, Carnegie Mellon University, 1993.

[6] P. Jeanrenaud, M. Siu, H. Gish, "Large Vocabulary Word Scoring as a Basis for Transcription Generation," *Proceedings of Eurospeech,* 1995.

[7] M. F. Porter, "An algorithm for suffix stripping," *Program*, 14(3):130-137, July 1980.

[8] J. R. Quinlan, *Programs for Machine Learning*, San Francisco, Calif.: Morgan Kaufmann, 1993.

[9] K. Seymore, S. Chen, M. Eskenazi, and R. Rosenfeld. "Language and Pronunciation Modeling in the CMU 1996 Hub 4 Evaluation," *Proc. Spoken Language Systems Technology Workshop*. Morgan Kaufmann Publishers, 1997.

[10] M. Siegler, U. Jain, B. Raj, and R. Stern. "Automatic Segmentation, Classification, and Clustering of Broadcast News Audio," *Proc. Spoken Language Systems Technology Workshop*. Morgan Kaufmann Publishers, 1997.

Acknowledgments

This research was supported in part by DARPA under research contract F33615-93-1-1330 and N00039-91-C-0158. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DARPA or the U. S. Government. We would like to thank Ravi Mosur, Eric Thayer, and Stan Chen for their invaluable contributions to this work.