Retrieval of Spoken Documents: First Experiences

Fabio Crestani* and **Mark Sanderson**
Department of Computing Science
University of Glasgow
Glasgow G12 8QQ, Scotland

9th October 1997

Abstract

We report on our first experiences in dealing with the retrieval of spoken documents. While lacking the tools and the know-how for performing speech recognition on the spoken documents, we tried to use in the best possible way our knowledge of probabilistic indexing and retrieval of textual documents. The techniques we used and the results we obtained are encouraging, motivating our future involvement in other further experimentation in this new area of research.

*Supported by a "Marie Curie" Research Fellowship from the European Community.

Contents

1 Introduction **3**

2 Probabilistic Information Retrieval **3**

2.1 The binary independence retrieval model 4

2.2 Term weighting schemas 9

3 The SIRE Information Retrieval system **11**

4 The Abbot speech recognition system **12**

5 Spoken document retrieval at TREC-6 **14**

6 The SDR TREC-6 data set **15**

7 Experimenting probabilistic retrieval of spoken documents **18**

7.1 The PFT weighting schema 18

7.2 Generating a weighting schema by merging different transcriptions 21

8 Results and discussion **22**

8.1 Results from the training data set 23

8.2 Results from the test data set 30

8.3 Official SDR results for TREC-6 32

9 Related work **37**

10 Conclusions and future works **37**

1 Introduction

Retrieval of spoken documents is a fast emerging area of research in Multimedia Information Retrieval. It involves the effective combination of the most advanced techniques used in speech recognition and Information Retrieval (IR). The increasing interest in this area of research is confirmed by the inclusion, for the first time, of a retrieval of spoken documents track in the TREC initiative [7].

Despite our lack of know-how and tools for speech recognition, we decided to participate in the retrieval of spoken documents track of TREC-6, confident that our knowledge of natural language processing and probabilistic information retrieval would compensate. Because of the way the retrieval of spoken documents track of TREC-6 is set up, groups with no speech recognition tools can participate all the same. However, thanks to our research contacts, we were able to team up with the Speech and Hearing Research Group of the Department of Computing Science of the University of Sheffield. They provided us with the transcripts of their speech recognition system, Abbot, that we used for most of our experiments.

This paper reports our first experiences in retrieval of spoken documents. The results we obtained are encouraging towards continuing working in this interesting area of research.

The paper is structured as follows. In Section 2 we describe in detail the necessary background on probabilistic IR so that some of the indexing choices taken in the rest of the paper will be better understood. Sections 3 and 4 describe the tools used for the experiments reported in this paper. Section 5 describes the experimental framework in which our investigation took place: the data sets used, the evaluation procedure, etc. Section 7 describes the two indexing schema we experimented with. The results of a set of experiments with these indexing schemas are reported and discussed in Section 8.

2 Probabilistic Information Retrieval

In IR, probabilistic modelling concerns the use of a model that ranks documents in decreasing order of their evaluated probability of relevance to a user's information need (also known as Retrieval Status Value). Past and present research has made much use of formal theories of probability and of statistics in order to evaluate, or at least estimate, these probabilities of relevance. These attempts are to be distinguished from looser ones like, for example, the "vector space model" [14] in which documents are ranked according to a measure of similarity with the query. A measure of similarity cannot be directly interpretable as a probability. In addition, similarity based models generally lack the theoretical soundness of

probabilistic models.

The first attempts to develop a probabilistic theory of retrieval were made over thirty years ago [9]. Since then, there has been a steady development of the approach, so that there are already several operational IR systems based upon probabilistic or semi-probabilistic models.

One major obstacle with probabilistic or semi-probabilistic IR models is that of finding methods for estimating the probabilities used to evaluate the probability of relevance that are both theoretically sound and computationally efficient. The problem of estimating these probabilities is difficult to tackle unless some simplifying assumptions are made. In the early stages of the study of probabilistic modelling in IR, assumptions related to event independence were employed in order to facilitate the computations. One of the first models to be based upon such assumptions was the *binary independence retrieval model*.

2.1 The binary independence retrieval model

As in most IR models, it is assumed that queries and documents are described by sets of index terms. Let $T = \{t_1, \ldots, t_n\}$ denote the set of terms used in the collection of documents. We represent the query q_k with terms belonging to T. Similarly, we represent a document d_j as the set of terms occurring in it. If we use a binary representation then d_j is represented as the binary vector $\vec{x} = (x_1, \ldots, x_n)$ with $x_i = 1$ if $t_i \in d_j$ and $x_i = 0$ otherwise. The query q_k is represented in the same manner.

The basic assumption, common to many other models, is that the distribution of terms within the document collection provides information concerning the relevance of a document to a given query. This is because it is assumed that terms are distributed differently in relevant and non-relevant documents. This is known as the *cluster hypothesis* (see [18] pp. 45-47). If the term distribution was the same within the sets of relevant and non-relevant documents then it would not be possible to devise a discrimination criterion between them. In which case, a different representation of the document information content would be necessary.

The term distribution provides information about the *probability of relevance* of a document to a query. If we assume binary relevance judgements, then the term distribution provides information about $P(R \mid q_k, d_j)$.

The quantity $P(R \mid q_k, \vec{x})$, with $\vec{x}$ as a binary document representation, cannot be estimated directly. Instead, Bayes' theorem is applied [10]:

$$P(R \mid q_k, \vec{x}) = \frac{P(R \mid q_k) \cdot P(\vec{x} \mid R, q_k)}{P(\vec{x} \mid q_k)}$$

$C_j(R, dec)$	retrieved	not retrieved
relevant document	0	λ_1
non relevant document	λ_2	0

Table 1: The cost of retrieving and not retrieving a relevant and non relevant document

To simplify notation, we omit the q_k on the understanding that evaluations are with respect to a given query q_k. The previous relation becomes:

$$P(R \mid \vec{x}) = \frac{P(R) \cdot P(\vec{x} \mid R)}{P(\vec{x})}$$

where $P(R)$ is the prior probability of relevance, $P(\vec{x} \mid R)$ is the probability of observing the description $\vec{x}$ conditioned upon relevance having been observed, and $P(\vec{x})$ is the probability that $\vec{x}$ is observed. The latter is determined as the joint probability distribution of the n terms within the collection. The above formula evaluates the "posterior" probability of relevance conditioned upon the information provided in the vector $\vec{x}$.

The provision of a ranking of documents by the Probability Ranking Principle [12] can be extended to provide an "optimal threshold" value. This can be used to set a cut-off point in the ranking to distinguish between those documents that are worth retrieving and those that are not. This threshold is determined by means of a *decision strategy*, whose associated *cost function* $C_j(R, dec)$ for each document d_j is described in Table 1.

The decision strategy can be described simply as one that minimises the average cost resulting from any decision. This strategy is equivalent to minimising the following *risk function*:

$$\mathcal{R}(R, dec) = \sum_{d_j \in D} C_j(R, dec) \cdot P(d_j \mid R)$$

It can be shown (see [18], pp. 115-117) that the minimisation of that function brings about an optimal partitioning of the document collection. This is achieved by retrieving only those documents for which the following relation holds:

$$\frac{P(d_j \mid R)}{P(d_j \mid \overline{R})} > \lambda$$

where

$$\lambda = \frac{\lambda_2 \cdot P(\overline{R})}{\lambda_1 \cdot P(R)}$$

It remains now necessary to estimate the joint probabilities $P(d_j \mid R)$ and $P(d_j \mid \overline{R})$, that is $P(\vec{x} \mid R)$ and $P(\vec{x} \mid \overline{R})$ if we consider the binary vector document representation $\vec{x}$.

In order to simplify the estimation process, the components of the vector $\vec{x}$ are assumed to be stochastically independent when conditionally dependent upon R or $\overline{R}$. That is, the joint probability distribution of the terms in the document d_j is given by the following product of marginal probability distributions:

$$P(d_j \mid R) = P(\vec{x} \mid R) = \prod_{i=1}^{n} P(x_i \mid R)$$

and

$$P(d_j \mid \overline{R}) = P(\vec{x} \mid \overline{R}) = \prod_{i=1}^{n} P(x_i \mid \overline{R})$$

This *binary independence assumption*, is the basis of a model first proposed by Robertson and Spark Jones in 1976 [13]: the *Binary Independence Retrieval model* (BIR). The assumption has always been recognised as unrealistic.

Nevertheless, as pointed out by Cooper [1], the assumption that actually underpins the BIR model is not that of binary independence, but that of the weaker assumption of *linked dependence*:

$$\frac{P(\vec{x} \mid R)}{P(\vec{x} \mid \overline{R})} = \prod_{i=1}^{n} \frac{P(x_i \mid R)}{P(x_i \mid \overline{R})}$$

This states that the ratio between the probabilities of $\vec{x}$ occurring in relevant and non relevant documents is equal to the product of the corresponding ratios of the single terms.

Considering the decision strategy of the previous section, it is now possible to devise a decision strategy by using a logarithmic transformation to obtain a linear decision function:

$$g(d_j) = \log \frac{P(d_j \mid R)}{P(d_j \mid \overline{R})} > \log \lambda$$

To simplify notation, we define the following quantities: $p_j = P(x_j = 1 \mid R)$, and $q_j = P(x_j = 1 \mid \overline{R})$ which represent the probability of the jth term appearing in a relevant, and in a non relevant document, respectively. Clearly: $1 - p_j = P(x_j = 0 \mid R)$, and $1 - q_j = P(x_j = 0 \mid \overline{R})$. This gives:

$$P(\vec{x} \mid R) = \prod_{j=1}^{n} p_j^{x_j} \cdot (1 - p_j)^{1-x_j}$$

and

$$P(\vec{x} \mid \overline{R}) = \prod_{j=1}^{n} q_j^{x_j} \cdot (1 - q_j)^{1-x_j}$$

Substituting the above, gives:

$$\begin{aligned} g(d_i) &= \textstyle\sum_{j=1}^{n} (x_j \cdot \log \frac{p_j}{q_j} + (1 - x_j) \cdot \log \frac{1-p_j}{1-q_j}) \\ &= \textstyle\sum_{j=1}^{n} c_j \; x_j + C \end{aligned}$$

where:

$$c_j = \log \frac{p_j \cdot (1 - q_j)}{q_j \cdot (1 - p_j)}$$

and

$$C = \sum_{j=1}^{n} \log \frac{1 - p_j}{1 - q_j}$$

This formula gives the Retrieval Status Value (RSV) of document d_j for the query under consideration. Documents are ranked according to their RSV and presented to the user. The cut-off value λ can be used to determine the point at which the displaying of the retrieved documents to the user should be stopped, although, the RSV is generally used to rank the entire collection of documents.

In a real IR system, the presentation of documents ordered on their estimated probability of relevance to a query matters more than the actual value of those probabilities. Therefore, since the value of C is constant for a specific query, we need only consider the value of c_j. This value, or more often the value $\exp(c_j)$, is called the *term relevance weight* (TRW), and indicates the term's capability

to discriminate relevant from non relevant documents. Because of the binary independence assumption, in the BIR model term relevance weights contribute "independently" to the relevance of a document.

To apply the BIR model, it is necessary to estimate the parameters p_j and q_j for each term used in the query. This is performed in various ways, depending upon the amount of information available. The estimation can be retrospective or predictive. The first is used on test collections where the relevance assessments are known. The second is used with normal collection where parameters are estimated by means of relevance feedback from the user.

Relevance feedback is a technique that allows a user to interactively express his information requirement by adapting his original query formulation with further information [6]. This additional information is often provided by indicating some relevant documents among the documents retrieved by the system.

Let us assume that the IR system has already retrieved some documents for the query q_k. The user is asked to give relevance assessments for those documents, from which the parameters of the BIR are estimated. If we also assume to be working in the retrospective case, then we know the relevance value of all individual documents in the collection. Let a collection have N documents, R of which are relevant to the query. Let n_j denote the number of documents in which the term x_j appears, amongst which, only r_j are relevant to the query. The parameters p_j and q_j can then be estimated as follows:

$$\hat{p_j} = \frac{r_j}{R}$$

and

$$\hat{q_j} = \frac{n_j - r_j}{N - R}$$

These give:

$$TRW_j = \frac{\frac{r_j}{R-r_j}}{\frac{n_j-r_j}{N-n_j-R+r_j}}$$

This approach is possible only if we have relevance assessments for all documents in the collection, i.e. where we know R and r_j. According to Croft and Harper, given that the only information concerning the relevance of documents is that provided by a user through relevance feedback, predictive estimations should

be used. Let $\tilde{R}$ denote the number of documents judged relevant by the user. Further, let $\tilde{r_j}$ be the number of those documents in which the term x_j occurs. We can then combine this with the estimation technique of [2].

$$T\tilde{R}W_j = \frac{\frac{\tilde{r_j}+0.5}{\tilde{R}-\tilde{r_j}+0.5}}{\frac{n_j-\tilde{r_j}+0.5}{N-n_j-\tilde{R}+\tilde{r_j}+0.5}}$$

Usually, the relevance information given by a user is limited and is not sufficiently representative of the entire collection. Consequently, the resulting estimates tend to lack precision. As a partial solution, one generally simplifies by assuming p_j to be constant for all the terms in the indexing vocabulary. The value $p_j = 0.5$ is often used, which gives a TRW that can be evaluated easily:

$$T\tilde{\tilde{R}}W_j = \frac{N - n_j}{n_j}$$

Work up to this point requires the use of at least a few relevant documents, making this model more closely related to retrieval as an interactive precess between the user and the IR system. This is exploited by the relevance feedback technique. However, we need to be able to use this model even in the absence of relevance information from the user, as in the predictive case, when the user first submit his query. In the next section we will see how probabilistic retrieval work in this case.

2.2 Term weighting schemas

In the previous section we have described the probabilistic retrieval model known as Binary Independence Retrieval model. In that model there are two important quantities that need to be explicated in order to use it in the predictive case:

1. the component x_j of the document vector representation $\vec{x}_j$;
2. the term relevance weight (TRW).

In the previous section we have assumed a binary document vector representation, that is, a document is represented by a vector whose values are zeros and ones depending if the feature represented by the position of the element of the vector is present or absent from the document. If document features are terms (or keywords), this indexing schema is very simplistic, since it does not take into

consideration the fact that not all terms presented in the document have the same representational power. Some terms can be very important in characterising the document informative content, some other can be so useless that they can be completely discarded (these are called stopwords). One intuitive notion of the importance of a term for representing the document informative content is related to its frequency of occurrence in the document. The more frequently a term is present in the document the more likely it is that the document will be related in content to the topic represented by that term. Some terms, however, are simply highly frequent in documents only because they are use frequently in the language (eg. "the", "a", "of", etc.). Once, we get rid of highly frequent terms, a good weighting schema for measuring the importance of a term in the context of a document is the *term frequency*

$$TF_{ij} = freq_i(x_j)$$

where $freq_i(x_j)$ denotes the frequency of occurrence of term x_j in document $\vec{x}_i$. A more complex formulation of the TF weighting schema that has proved empirically more effective is the following:

$$TF_{ij} = K + (1 - K)\frac{freq_i(x_j)}{maxfreq_i}$$

where K is a constant that need to be set experimentally and $maxfreq_i$ is the maximum frequency of any term in document $\vec{x}_i$.

This value can be used as a component of the document vector $\vec{x}_i$, instead of a binary component. Since it is independent of relevance information, it can be used both in the predictive and the relevance feedback cases.

In the predictive case, for large N, i.e. large collections of documents, the term relevance weight (TRW) can be approximated by the *inverse document frequency*:

$$IDF_j = \log \frac{N}{n_j}$$

where N denotes the number of documents in the collection and n_j the number of documents in which the term x_j occurs.

The IDF weighting schema is widely used in IR to provide a measure of the discrimination power of a term in a document collection. This weight is based on Luhn's assumption and on the assumption that the discriminating power of a term is inversely proportional to the number of documents in which that term

occurs [8]. In particular, the inverse document frequency reflects the intuition that the larger the number of documents that are indexed by the same term, the less important the term becomes as a descriptor of any of them.

We can now combine the above two weighting schemas with the retrieval formulas reported in the previous section to obtain a complete specification of the model. While we know from the previous section how to deal with retrieval once we have relevance information, we need to be able to deal with retrieval in the absence of retrieval information. The solution proposed in [4] is to use use the following formula:

$$g(d_i) = \sum_{j=1}^{n} TF_{ij}(C + IDF_j)$$

where C is a constant that is set experimentally to tailor the weighting schema to different collections.

These statistically–based weighting techniques can be enhanced by *conflation* techniques which attempt to map individual word tokens to a single morphological form. IR has developed also sophisticated techniques for *stemming* (see for example [11]), which are used in most of the operational IR systems. Some experimental systems attempt to use phrases instead of individual terms (see for example [15]), but the automatic identification of phrases in free texts is a problematic task.

In the rest of the paper we will use the two words "word" and "term" interchangeably, although there a clear difference between the two. Term is a word mainly used in IR where it refers to a textual feature of a document. In this sense a "term" is a "word" that has been chosen for indexing a document. Word is instead mainly used in the speech recognition area where it refers to a single unit of language represented by one or more phonemes.

The different use of these words in the two communities, IR and speech recognition, would make the paper very confusing if we were to keep distinguishing between them.

3 The SIRE Information Retrieval system

The system used in the context of the work reported in this paper is a retrieval toolkit called *SIRE* (System for Information Retrieval Experimentation) developed "in-house" at Glasgow University by Mark Sanderson. SIRE is a collection of small independent modules, each conducting one part of the indexing, retrieval

and evaluation tasks required for classic retrieval experimentation. The modules are linked in a pipeline architecture communicating through a common token based language. SIRE was initially used in research examining the relationship between word sense ambiguity, disambiguation, and retrieval effectiveness [17]. It proved to be a flexible tool as it not only provided retrieval functionality but a number of its core modules were used to build a word sense disambiguator as well. It was also used in the experiments for the Glasgow IR group submissions to TREC-4 and TREC-5 and is currently being used in a number of research efforts within the group.

SIRE is implemented on the UNIX operating system which, with its scripting and pre-emptive multi-tasking is eminently suitable for supporting the modular nature of SIRE.

SIRE was chosen as the IR platform for the experiments reported in this paper because it implemented a probabilistic IR model based on the "TF-IDF" weighting schema reported in Section 2. Moreover, it was relatively easy to modify the code to take into account the characteristics of the new data.

A detailed description of the functionalities of SIRE is outside the scope of this paper. The system is currently public available for research purposes. The interested reader should contact Mark Sanderson for a copy of a short unpublished paper describing the system [16] and for the location of SIRE's binary files. The system has been successfully used by many students of the Advance Information Systems M.Sc. of Glasgow University for their practical work.

4 The Abbot speech recognition system

Abbot is a speaker independent continuous speech recognition system developed by the Connectionist Speech Group at Cambridge University and now jointly supported by Cambridge and Sheffield Universities with commercialisation by SoftSound.

The Abbot system grew out of work on recurrent neural networks at Cambridge. It was further developed under the ESPRIT project "Auditory Connectionist Techniques for Speech" and then the ESPRIT project "WERNICKE: A Neural Network Based, Speaker Independent, Large Vocabulary, Continuous Speech Recognition System". Currently further development is being funded by the Framework 4 projects "SPRACH: Speech Recognition algorithms for connectionist hybrids" and "THISL: Thematic Indexing of Spoken Language".

The system is designed to recognise British English and American English clearly spoken in a quiet acoustic environment. The system is based on a model that is

a combination of a connectionist and a Hidden Markow model.

Most, if not all, automatic speech recognition systems explicitly or implicitly compute a score (distance, probability, etc.) indicating how well a novel utterance matches a model of the hypothesised utterance. A fundamental problem in speech recognition is how this score may be computed, given that speech is a non-stationary stochastic process. In the interest of reducing the computational complexity, the standard approach used in the most prevalent systems factors the hypothesis score into a local acoustic score and a local transition score. In the hidden Markov models (HMM) framework, the observation word models the local (in time) acoustic signal as a stationary process, while the transition probabilities are used to account for the time-varying nature of speech.

The Abbot system uses an extension to the standard HMM framework which addresses the issue of the observation probability computation. Specifically, an artificial recurrent neural network (RNN) is used to compute the observation probabilities within the HMM framework. This provides two enhancements to standard HMMs:

- the observation model is no longer local;
- the RNN architecture provides a non-parametric model of the acoustic signal.

The result is a speech recognition system able to model long-term acoustic context without strong assumptions on the distribution of the observations. Abbot has been successfully applied to a 20,000 word, speaker-independent, continuous speech recognition task, showing good levels of performance.

An in depth treatment of the characteristic of Abbot is outside the scope of this paper. The interested reader is directed to the extensive bibliography on the Abbot system[1] and in particular the document titled "The use of recurrent neural networks in continuous speech recognition" by Tony Robinson, Mike Hochberg and Steve Renals, not yet published but available online[2]. A demo version of the Abbot system is available to the public free of charge. *AbbotDemo*[3] is a packaged demonstration of the Abbot system. The demonstration system has a vocabulary of $10,000$ words, anything spoken outside this vocabulary can not be recognised (and therefore will be recognised as another word or string of words).

[1]See http://svr-www.eng.cam.ac.uk/~ajr/GroupPubs/publications.html.
[2]See http://svr-www.eng.cam.ac.uk/~ajr/rnn4csr94/rnn4csr94.html.
[3]Available at http://svr-www.eng.cam.ac.uk/~ajr/abbot.html.

5 Spoken document retrieval at TREC-6

TREC (Text REtrieval Conference) is a workshop series sponsored by the National Institute of Standards and Technology (NIST) and the Defence Advanced Research Projects Agency (DARPA) that promotes IR research by providing appropriate test collections, uniform scoring procedures, and a forum for organisations interested in comparing their results. TREC tracks allow participants to focus on particular subproblems of the retrieval task[4].

The annual TREC is an event in which organisations with an interest in IR and information routing take part in a coordinated series of experiments using the same experimental data and queries. The results of these individual experiments are then presented at the workshop where tentative comparisons are made. In order to preserve the desired, pre-competitive nature of these conferences, the organisers have developed a set of guidelines constraining the dissemination and publication of TREC evaluation results. These guidelines are meant to preclude the publication of incomplete or inaccurate information that could damage the reputation of the conference or its participants and could discourage participation in future conferences. The guidelines apply to all TREC participants, regardless of the type of organisation or institution involved. A signed agreement is required of each organisation participating in the TREC evaluations. Any organisation that is found to have violated the terms or spirit of the agreement may be denied participation in future TRECs.

TREC-6 (the 1997 TREC event) included, for the first time, a track on *Spoken Document Retrieval* (SDR). The remainder of this section outlines the SDR test paradigm and describes the data for the test.

The SDR track was designed to foster as much participation as possible in keeping with TREC's retrieval charter, rather than, say, the cleanest experimental design or the simplest track specification. While sites with both speech recognition and information retrieval systems were strongly encouraged to participate in the full SDR evaluation, baseline speech and retrieval components were offered so those sites with only one of these technologies can also participate. Speech and retrieval sites were encouraged to team up to produce full SDR systems for the evaluation. As this is a TREC track, all participants were required to produce retrieval output in a particular format.

The track offered two modes of participation: SDR for those with speech recognisers and Q(uasi)SDR for those without. The latter is intended as a startup for those in the IR community without immediate access to speech recognition

[4]More details about TREC can be found on the TREC home page at NIST: http://www-nlpir.nist.gov/trec.

expertise. Those in the speech community may use any IR system available to them including public available systems such as NIST's ZPRISE system.

Note that this track was an "experiment". From the IR point of view, the test was simple and small in scale. But this was unavoidable because from a speech point of view the scale was far from small.

The main administration of the track was by John Garofolo (speech) and Ellen Voorhees (IR) both of the NIST.

6 The SDR TREC-6 data set

The SDR track uses stories taken from the Linguistic Data Consortium (LDC) 1996 Broadcast News corpora. This data was used in the November 1996 HUB-4 DARPA Speech Recognition Evaluation.

The data set is divided in a training set and a test set to be used according to the established TREC experimental methodology [7].

The *training set* is a set of about 1000 stories (about 50 hours) with 6 known-item search topics, i.e. topics constructed to retrieve a single known document. A story, i.e. document, is generally defined as a continuous stretch of news material with the same content or theme (e.g. tornado in the Caribbean, fraud scandal at Megabank), which will have been established by hand segmentation of the news programmes. Note, however, that some stories such as news summaries may contain topically varying material. Also note that a story is likely to involve more than one speaker, background music, noise etc.

The *test set* consists in a similar set of about 1500 stories representing about 50 hours of recorded material (though after the removal of commercials, there is a bit less speech), with a set of 50 known-item search topics.

There are 4 forms of the story data supplied for both the training and the test data sets:

sph Sphere formatted speech files: digitised recordings of the broadcasts. File labelled sph (format used in DARPA speech evaluations).

dtt Detailed TREC Transcriptions: hand-generated transcriptions used in speech recogniser training and for speech recognition scoring that use the established DARPA broadcast news SGML-tagged annotation/transcription convention, hence not conveniently readable as simple text. These are the LDC-generated transcripts with absolute section (story) IDs added. File labelled dtt.

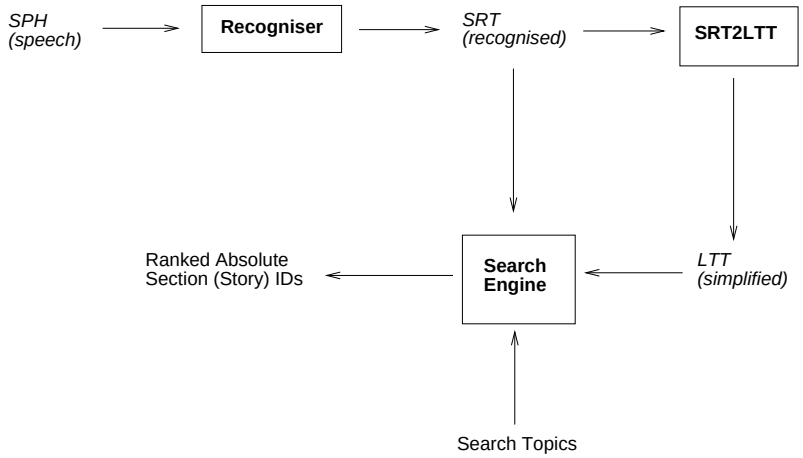

Figure 1: SDR track data flow diagram

ltt Lexical TREC Transcriptions: DTT's with most SGML tags removed and hence conveniently readable as text. File labelled ltt.

srt Speech Recogniser Transcriptions: automatically-generated transcriptions (therefore likely to contain recognition errors) produced by a particular recogniser when applied to sph. The file format is identical to LTT except that each word is bracketed by an SGML tag pair that indicates the time at which the word occurs. SRTs generated by a volunteer speech site are used for Baseline testing (these are called the "BSRTs"). File labelled srt.

The motivation for the Baseline Speech Recogniser Transcriptions is to allow retrieval people interested in SDR, but not able to get together in time with anyone with a recogniser, to use output generated by a real recogniser. These people form the QSDR Group, engaged with quasi-speech.

All of the above file types were cross-linked by SGML-tagged time markers for story beginnings and ends. See Figure 1 for a detailed account of the relationships between these formats and see the Appendix for samples of each format.

In addition, the following files were provided:

sil Speaker Information Log: used to cross reference information about the speakers in the transcripts. This were used primarily by speech sites in calibration. File labelled sil.

ndx Index: containing only Episode and Section tags used by the speech recognition systems to produce SGML-tagged output for the evaluation. File labelled ndx.

The required retrieval *runs* are as follows.

For SDR Group participants only:

S1 Speech run: a full SDR run with their own recogniser on the SPHERE-formatted digitised broadcast news recordings.

In addition to being evaluated on its retrieval effectiveness, this run will also be evaluated on its Speech Recognition performance using Error Rate measures of the kind normally used for DARPA CSR tests. Each SDR speech recognition system's output will be scored and tabulated. The same scoring protocol will be applied to the Baseline Speech Recogniser Transcriptions used in B1 as well.

For all participants:

B1 Baseline run: a retrieval run using the the Baseline Speech Recogniser Transcriptions as input. This is the "speech run" for QSDR participants. However, SDR Group participants are also required to do this run to enable speech-based retrieval comparisons for all track participants.

R1 Reference run: a retrieval run using the reference (hand-transcribed) ltt as input. This run enables retrieval-based comparisons across all track participants.

Participants must use the same retrieval strategy for these 3 runs (that is, term weighting method, stop word list, use of phrases, retrieval model, etc must remain constant). While it would be nice if SDR participants could could carry their general approach to handling speech data for retrieval (e.g. keep only items with high acoustic scores) across to the Baseline run, this is unlikely to be possible in practice because the necessary information will not come with the BSRT.

The three runs just listed are obligatory. Participants may optionally submit a second speech run and a second baseline run, S2, B2, to test the effects of variations in their own system parameter settings.

These required runs support retrieval performance comparisons as follows:

- between members of the SDR Group (i.e. the "real" spoken document retrieval case), as a black box comparison not distinguishing the relative contributions for any given system between the recognition strategy and the retrieval strategy.

- between members of the QSDR Group, to compare retrieval strategies for the one shared recognition strategy, i.e. the one that delivered the Baseline Speech Recogniser Transcriptions.
- between all participants, SDR and QSDR, to compare retrieval strategies, via the Baseline Speech Recogniser Transcriptions.
- for each participant, between spoken document retrieval and text retrieval using the Lexical TREC Transcriptions, to calibrate the former against the latter for the participant's own retrieval strategy.
- for all participants, on text retrieval with the Lexical TREC Transcriptions, to compare retrieval strategies.

Together these comparisons should show, via the Lexical TREC Transcription text runs, what the level of performance would be for the given documents and topics with a perfect speech recogniser and the teams' various retrieval strategies, and via the other runs, what the effects of the various recognisers are. The slightly heavy detail follows from allowing for QSDR participants as well as SDR ones.

7 Experimenting probabilistic retrieval of spoken documents

In this section we report a detailed account of the strategies we used for our first experiments with retrieval of spoken documents.

7.1 The PFT weighting schema

One of the characteristics of the Abbot speech recognition system is the fact that it associates a measure of uncertainty to each word it recognises, as can be seen from the following example of a srt file produced as output. See the appendix for a comparison with the srt file of the Baseline testing and note the absence of the probability values attached to words.

```
<Episode Filename=a960521.sph Program="ABC_Nightline"
Scribe="obert_markoff" Dat e="960521:2330" Version=4 Version_Date=961011>
.
.
.
<Section S_time=0.000 E_time=61.320 Type=Filler ID="a960521.1" >
```

```
<Word S_time=1.76 E_time=2 Prob=-1.873> IT'S </Word>
<Word S_time=2 E_time=2.048 Prob=-0.9346> A </Word>
<Word S_time=2.048 E_time=2.656 Prob=2.025> QUESTION </Word>
<Word S_time=2.656 E_time=2.832 Prob=-0.6394> THAT </Word>
<Word S_time=2.832 E_time=2.992 Prob=-0.3682> WILL </Word>
<Word S_time=2.992 E_time=3.36 Prob=1.188> MAKE </Word>
<Word S_time=3.408 E_time=3.488 Prob=-0.9622> A </Word>
<Word S_time=3.488 E_time=3.872 Prob=2.335> LOT </Word>
<Word S_time=3.872 E_time=3.984 Prob=0.4647> OF </Word>
<Word S_time=3.984 E_time=4.672 Prob=5.322> AMERICANS </Word>
<Word S_time=4.672 E_time=4.864 Prob=-0.4521> THINK </Word>
<Word S_time=6.882 E_time=6.994 Prob=-2.392> TO </Word>
<Word S_time=6.994 E_time=7.234 Prob=-1.807> HAVE </Word>
<Word S_time=7.234 E_time=7.346 Prob=-3.124> TO </Word>
<Word S_time=7.91 E_time=8.086 Prob=-0.2239> YOU </Word>
<Word S_time=8.086 E_time=8.294 Prob=0.1139> SAY </Word>
<Word S_time=8.294 E_time=8.454 Prob=-2.961> TO </Word>
<Word S_time=8.454 E_time=8.95 Prob=-3.794> ONE </Word>
.
.
.
</Section >
```

These measures of uncertainty are incorrectly called probabilities. However they are not probabilities, as an explanation of the way these are computed helps clarifying:

1. For a given time segment, the neural network at the heart of Abbot provides a set of posterior probabilities for each phoneme. These are the "acoustic probabilities".

2. To facilitate the decoding, the acoustic probabilities are converted into scaled likelihoods by dividing by the prior probability of the phoneme.

3. During decoding, a search is performed using the acoustic probabilities and the language model to find the most likely sequence of words for that utterance.

4. As each word is defined as a sequence of phonemes, the score for that word is obtained by summing the scores of the individual phones which constitute that word. (Summing because Abbot works with log probabilities).

Although they are not probabilities, we can still consider them as weights expressing the confidence given by Abbot in the correct recognition of words. This gave us the idea of combine these weights with the probabilistic model underlying

SIRE. As already explained in Section 2, the probabilistic model used by SIRE assigned to every index term extracted from the text of a document a weight that is a combination of two different discrimination measures: the IDF and the TF. The IDF of a term is a collection wide weight, since it is calculated taking into account the distribution of the term inside the all collection. The TF of a term is instead a document wide weight, since it is calculated taking into account the distribution of a term within a document. The TF is of particular interest in our discussion. the TF of a term is usually calculated as a normalised sum of the number of occurrences of that term in the document. If the occurrence of a term is a binary event, then

$$occ.(x_j) = \begin{cases} 1 & \text{if } x_j \text{ occurs in } d_i \\ 0 & \text{otherwise} \end{cases}$$

Therefore, in its simplest definition, the frequency of occurrence of a term is defined as follows:

$$freq_i(x_j) = \sum_{d_i} occ.(x_j)$$

We decided to use the probabilities Abbot assigns to words as a way of devising a more general definition of occurrence. We decided to use the following definition of occurrence:

$$occ'.(x_j) = \begin{cases} Prob(x_j) & \text{if } x_j \text{ occurs in } d_i \\ 0 & \text{otherwise} \end{cases}$$

Therefore the frequency of occurrence of a term is now defined as:

$$freq_i(x_j) = \sum_{d_i} Prob(x_j)$$

This definition of frequency is the one used to redefine TF as follows:

$$PTF_{ij} = freq_i(x_j)$$

We called this new definition of TF PFT (Probabilistic Term Frequency).

The above definition is quite intuitive. While TF measures the importance of a term in the context of a document as a function of the number of occurrences of

the term, PTF weights the number of occurrences of a term with the confidence assigned every time to the recognition of the occurrence of the term. In fact, it is intuitive that the PTF of a term should be higher in the case of the term being recognised as present in the document with high confidence values, than in the case of being recognised with low confidence values. In the latter case, in some instances, the term may have been mistaken for another term and may not even be present in the document.

In some of the experiments that follow we tried to see if a PTF-IDF weighting schema gives better performance than the classical TF-IDF. The actual formula for the PTF used in these experiments is, for reasons already explained in Section 2, the following:

$$PTF_{ij} = K + (1 - K)\frac{freq_i(x_j)}{maxfreq_i}$$

7.2 Generating a weighting schema by merging different transcriptions

In the previous section we have taken advantage of a particular feature of the transcription we had available, the probabilities assigned by Abbot to words in the transcription. We used these probabilities to generate a new weighting schema for the words in the transcription. However, a few questions that we posed ourself were: are these probabilities reliable? Is there any other strategy that we could use to generate confidence (or uncertainty) values to assigned to recognised words?

Another, perhaps naive, strategy that we decided to test was again due to our particular situation. We had two different speech recognition transcript for the same speech data. A first analysis of the two transcripts showed large differences in recognition:

BSRT:

```
<Section S_time=0.000 E_time=61.320 Type=Filler ID="a960523.1" >

I will talk about blacks and winds we eventually go wrong a
of the tough question who he hid ...

</Section>
```

Abbot:

```
<Section S_time=0.000 E_time=61.320 Type=Filler ID="a960523.1" >

we talked about blanks and whites we eventually get around
to the tough question his own unions say well ....

</Section>
```

DTT:

```
<Section S_time=0.000 E_time=61.320 Type=Filler ID="a960523.1" >

when we talk about blacks and whites we eventually get around
to the tough question some of you are ...

</Section>
```

The first two are respectively the BSRT and the Abbot transcriptions, while the last one is the DTT transcription, that is the real text of the speech. It is easy to spot the errors made by the two speech recognition systems. One interesting fact is that there are many cases of words correctly recognised by one system and wrongly by the other. For example, the word "blacks" has been correctly recognised by BSRT and wrongly by Abbot, while the word "white" has been correctly recognised by Abbot and wrongly by BSRT. If one of these two words would have been used in a query, the IR system could not avoid retrieving only the document in which the word has been recognised correctly.

The above findings suggested to us that a merging of the two speech recognition transcripts. In this case the correct recognition of one system could compensate for the wrong ones of the other system. Moreover, using the classical TF-IDF weighting schema, if a word has been correctly recognised by both systems, then it will have a larger frequency of occurrence and this will increase its weight in the context of the document. On the other hand, a word that has been wrongly recognised by one of the speech recognition systems will have a small frequency of occurrence (unless it has been consistently recognised wrongly, a case that we suppose does not happen frequently) and therefore a lower weight in the context of the document. We called *Merged* this weighting schema.

8 Results and discussion

The following two sections report the results of our experimental investigation into the effectiveness of the two weighting strategies outlined above. We performed a first set of experiments on the training data set in order to tune some of the

parameters involved in the retrieval strategies. Some more experiments were then performed on the test data set. Given the different sizes of the two data sets, the results obtained from the test data sets are statistically more significant than those obtained from the training data set.

Most of the results reported in the following two sections are presented in *recall and precision graphs* (P/R graphs). These graphs are obtained by depicting the precision figures at standard levels of recall using a techniques described in [18]. Here it may be useful to remind the definitions of recall and precision:

$$Recall\ (R) = \frac{\mid A \cap B \mid}{\mid A \mid}$$

$$Precision\ (P) = \frac{\mid A \cap B \mid}{\mid B \mid}$$

Where A is the set of relevant documents and B is the set of the retrieved documents. Therefore, $\mid A \cap B \mid$ is the number of relevant and retrieved documents, $\mid B \mid$ is the number of retrieved documents, and $\mid A \mid$ is the number of relevant documents.

8.1 Results from the training data set

A first set of experiments was performed using the training data set with the purpose of determining the best possible combination of IR system and speech recognition system.

Together with the Speech and Hearing Research Group of the Department of Computing Science of the University of Sheffield (the group who made available to us the Abbot generated transcripts), we had available the following three IR systems: PRISE, MG and SIRE. The latter system has already been described in Section 3.

The *PRISE* system is an experimental prototype of a full-text IR system developed by the Natural Language Processing and Information Retrieval Group at NIST. The NIST PRISE system treats documents and queries as lists of words and responds to a query with a list of documents ranked in order of their statistical similarity to the query. A basic version of the PRISE search engine (without the interface and without any client/server mechanism) has been available for research use for several years (contact Donna Harman, donna.harman@nist.gov for more information). The experiments with PRISE were performed by the Sheffield group.

Query	*Target*	*DTT/PRISE*	*BSRT/PRISE*	*Abbot/PRISE*
1	k960524.2	1	13	2
	k960524.17	2	1	1
2	j960522b.23	1	1	1
	g960522.21	2	3	4
3	f960615.12	3	2	2
4	j960522b.15	3	38	9
5	e960510b.14	1	1	2
6	e960510a.10	1	1	1

Table 2: Performance of PRISE system.

Query	*Target*	*DTT/MG*	*BSRT/MG*	*Abbot/MG*
1	k960524.2	2	8	2
	k960524.17	1	1	1
2	j960522b.23	5	1	5
	g960522.21	8	6	8
3	f960615.12	8	4	8
4	j960522b.15	7	26	21
5	e960510b.14	3	2	2
6	e960510a.10	1	1	1

Table 3: Performance of MG system.

The *MG* (Managing Gigabytes) system is a collection of programs which comprise a full-text retrieval system. It is "full-text" in the sense that every word in the text is indexed and the query operates only on this index to do the searching. MG is public domain[5]. MG is covered by a GNU public licence. The MG system is an embodiment of ideas developed primarily by: Tim C. Bell, University of Canterbury; Alistair Moffat, University of Melbourne; Ian Witten, University of Waikato; and Justin Zobel, RMIT. The system is described in [19].

We also had availability of two speech recognised transcripts and the hand transcribed data. The hand transcribed data (named DTT) correspond to the (presumably) perfect recognition of the spoken documents, as performed by a human. The two speech recognition systems could only try to get recognition performance as good as the DTT data. The two speech recognition systems were: the baseline speech recognition system (BSRT, the srt transcripts provided by NIST) and the already described Abbot system.

Tables 2, 3 and 4 report the results of a first experimentation into the performance of the IR systems with the hand transcribed data (DTT), the Baseline speech recogniser transcripts (BSRT), and the Abbot speech recogniser transcripts (Ab-

[5]The MG software can be ftped from: ftp://munnari.oz.au/pub/mg.

Query	*Target*	*DTT/SIRE*	*BSRT/SIRE*	*Abbot/SIRE*
1	k960524.2	1	13	2
	k960524.17	1	1	1
2	j960522b.23	1	1	1
	g960522.21	2	3	3
3	f960615.12	3	2	1
4	j960522b.15	3	38	10
5	e960510b.14	1	1	2
6	e960510a.10	1	1	1

Table 4: Performance of SIRE system.

bot). The tables report for each of the 6 queries available in the training data set the target documents and their positions in the ranking obtained by using different combinations of transcriptions and IR systems. Despite the limits of the conclusions that can be drawn from such a small experimental data set, it can be observed that there are considerable differences in the performance obtained.

For the DTT data, SIRE seems to be giving the best results, although these are similar to those given by PRISE, while MG gives bad results in particular for queries 2, 3 and 4. For the BSRT data, instead, we had a completely different result: MG performed better than PRISE and SIRE, with PRISE and SIRE performing at exactly the same level. The difference in performance is particularly evident for queries 4 and 1, where PRISE and SIRE seem to have considerable problems. This is probably due to the fact that some words that were wrongly recognised by the Baseline speech recognition system were given higher weights by PRISE and SIRE and not by MG. For the Abbot data we have performance figures more resembling the one obtained for the DTT data. PRISE and SIRE performed almost at the same level (with SIRE a little ahead) and MG falling behind on the same queries it performed badly with the DTT data.

It is difficult to explain these results. Our only possible explanation is related to the differences in the ways the three systems index the data. The three systems use different weighting schemas, different stemming algorithms, and have different stoplists (actually, MG does not have a stoplist). It would be necessary to look deeply into the weights assigned to every term in documents to be able to give a precise answer. Since this was outside the purpose of our set of experiments, we just took the result as motivating our choice of using SIRE for the following set of experiments[6].

The next set of experiments was targeted at finding the best possible combina-

[6]We would probably have chosen to use SIRE all the same. In fact, this was the system we knew the best, since it was developed in Glasgow by one of us. The above results were nonetheless reassuring

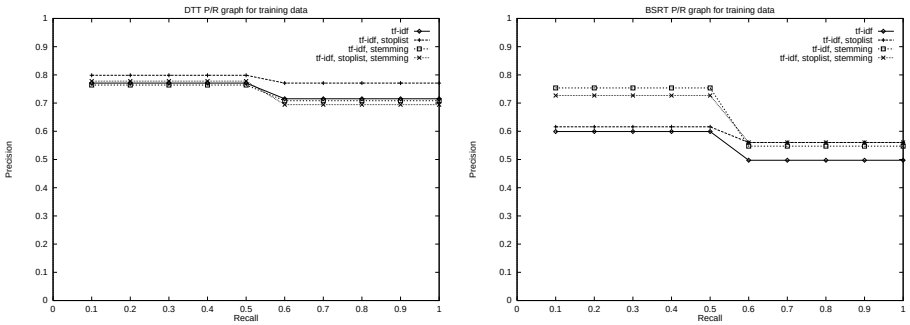

Figure 2: Performance figures for DTT and BSRT on the training data.

tion of indexing parameters for the use of SIRE. The flexible architecture of the SIRE system enabled us to implement various forms of indexing switching on or off the use of a stoplist and the use of a stemming procedure. Figure 2 report the performance of SIRE using DTT and BSRT from the training data. Different combinations of indexing are tested. The term "TF-IDF" refers to the weighting schema explained in detail in Section 2. At first sight, these results seem contradictory, since for DTT the best performance are obtained by using TF-IDF and stoplist, while for BSRT the best performance are achieved using TF-IDF and stemming. However, this result has a plausible interesting explanation.

The performance of SIRE on DTT data is what we would expect from the use of a classical probabilistic IR system on a textual collection. There has been a long debate in the past about the advantages and disadvantages of using stemming [5]. For some collections and/or for some IR systems stemming seems to improve performance, while for other collections and IR system performance seem to get worse. The difference, however, is never too large. This is our case.

The figures for SIRE on BSRT data show that stemming improve the performance of the IR system, in particular for low levels of recall. An explanation for this fact can be obtained by looking at the kind of errors a speech recognition system usually make. Lets look at the following example obtained from the training data.

Hand generated transcript (DTT):

```
<Section S_time=0.000 E_time=61.320 Type=Filler ID="a960521.1" >

it's a question that will make a lot of americans think damn you think
you're white you're not you're black it's a question that will make a
lot of americans angry in order for you to be black for the rest of
your life what would it take to compensate you for that how much do
you want how much do i want how much would it take we continue our
series america in black and white tonight how much is white skin worth
this is a. b. c. news nightline reporting from washington ted koppel
```

```
</Section >
```

Baseline speech recognition system generated transcript (BSRT):

```
<Section S_time=0.000 E_time=61.320 Type=Filler ID="a960521.1" >

it's a question little Laventhol law of Americans at an M.  more room
Missouri awarding of the No I applaud a Newton it's a question that
will make a lot of Americans and Ingrid an enormous refute of the
black but rose to Bulent whom men or what it did to compensate if on
the issue will of lost or one or more from them we continue was
serious America in light and want no denying women on G.  E.  is white
scheme were you then it had her hay half that this he has C.  News
Nightline who thought him half his his office

</Section>
```

Abbot speech recognition system generated transcript (Abbot):

```
<Section S_time=0.000 E_time=61.320 Type=Filler ID="a960521.1" >

it's a question that will make a lot of americans think to have to you
say to one run and do not know it is lef it's a question that will
make a lot of americans angry in order for you to be black for the
rest of your life what would it take to compensate here and that what
she wants i'm a strong one how much would the terrace where he hit her
we continue our series america in black and white tonight and not much
is white skin words these sleazy c.  news nightline reporting from
wanting him that toppled

</Section >
```

A comparison of the above speech recognition systems generated transcripts with the hand generated one shows how far we are from perfect speech recognition performance. It also shows how some of the mistakes can be understood at phonetic level. Take for example the word "angry" in the DTT, recognised as "Ingrid" by the BSRT, or the word "scheme" in the DTT recognised as "skin" by the BSRT, or again the word "worth" in the DTT recognised as "words" by Abbot. Given these kind of errors, it is conceivable that stemming could help alleviate some of them. In fact, stemming reduces a set of words with common stem to a unique word, the stem. In doing so stemming may smooth some of the errors the speech recognition system makes, in particular in the last phonemes. This is just a first and intuitive explanation of our results. We will need to perform some further experiments in order to confirm this theory. We intend to do that in the future.

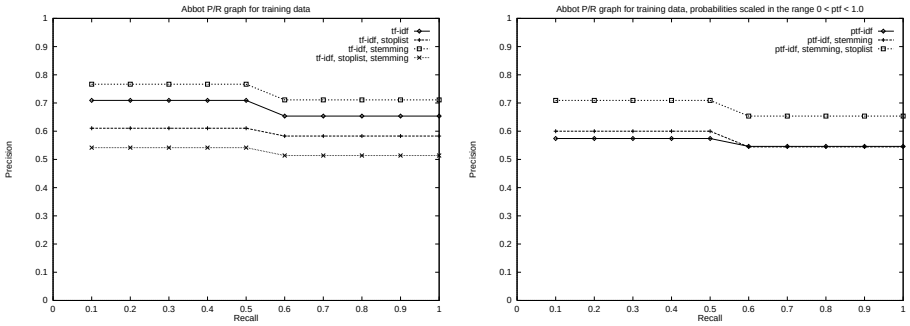

Figure 3: Performance figures for Abbot on the training data using TF and PTF weighting schemas.

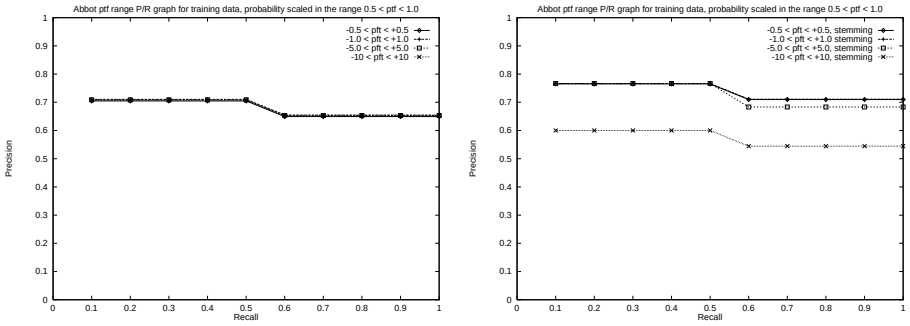

Figure 4: Performance figures for Abbot on the training data using pft.

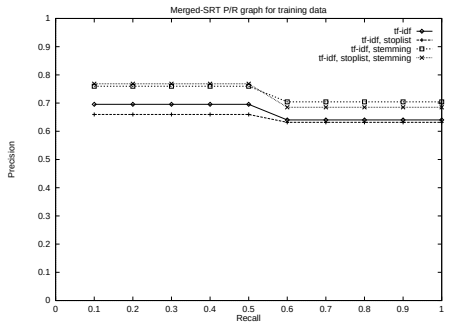

Figure 5: Performance figures for the Merged strategy on the training data.

Similar results with regards to the effect of stemming can be observed in Figures 3 and 4. These figures, however, report the results of another series of experiments aimed at using the probabilities values that Abbot attaches to recognised words (see Section 7.1). A problem with these probabilities is that, despite their name, they are not actually probabilities. Leaving aside consideration about the way they are evaluated, one of the major concerns is that they do not range in the usual interval $[0, 1]$. Moreover, a sketchy study of their distribution shows that it resembles a normal distribution (the classic distribution of errors) with average value $\mu(Prob.) = 1.861$ and standard deviation $\sigma(Prob.) = 2.412$. Using the properties of the normal distribution we know that 95% of these probability values are in the range $-2.963 \leq Prob. \leq 6.686$, while 99% will be in the range $-5.373 \leq Prob. \leq 9.098$.

On the left of Figure 3 we have the performance of Abbot without using the probabilities values, while on the right we have the performance of Abbot using them after having scaled them into the range $[0, 1]$, the same range used by SIRE for the TF values. However, it was apparent that using the full range $[0, 1]$ would jeopardise performance since a PTF value close to 0 assigned to a word would have had great consequences on the retrieval of the documents using that word. We therefore decided to scale the values of the probabilities assigned by Abbot in the range $[0.5, 1]$ also trying to cut off some of the extreme values. Figure 4, shows the performance of Abbot using the probabilities (called PTF) scaled in the range $[0.5, 1]$ and cut off threshold at different values. The performance of Abbot with and without stemming are also reported. The figures show that using the probabilities scaled in the range $[0.5, 1]$ and cleaned from some extreme values, in conjunction with stemming, gives performance that is marginally better, but more consistent, than those achieved by without the use of the probabilities. This result motivated us towards the use of this particular feature of Abbot.

Figure 5 reports the performance of the Merged strategy. Again, it can be seen that stemming helps improving the performance, although it seems that the use of a stoplist has marginal advantages in particular for high levels of recall. However,

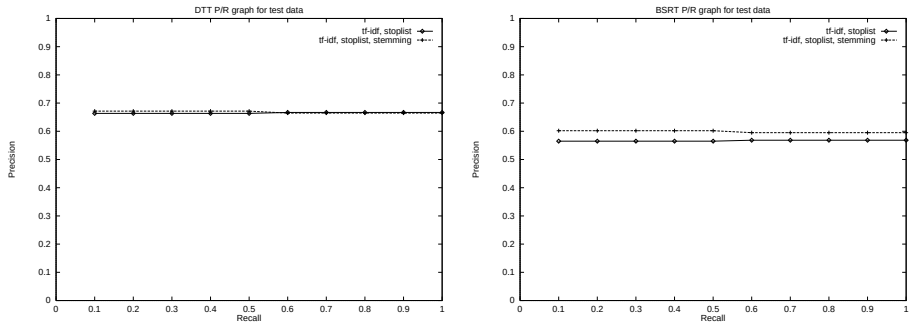

Figure 6: Performance figures for DTT and BSRT on the test data.

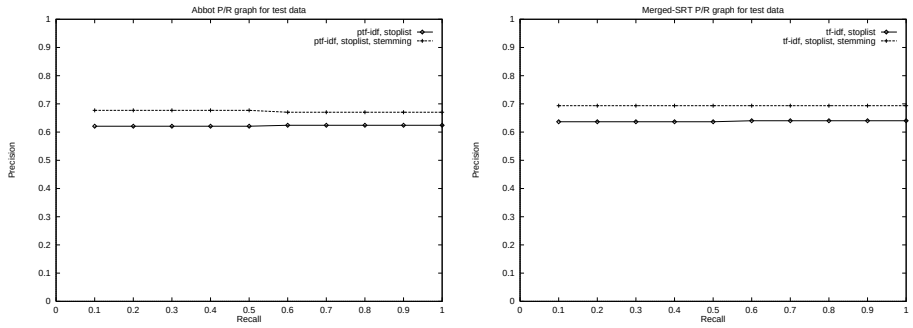

Figure 7: Performance figures for Abbot and Merged on the test data.

the difference between using or not of a stoplist is not statistically significant.

8.2 Results from the test data set

In this section we report the results obtained using the test data. We did not perform many of the experiments already performed on the training data, although the larger size of the test data could have helped confirming them. We instead decided to carry on a comparison of the effectiveness of SIRE on the four data sets: DTT, BSRT, Abbot, and Merged. We tested the performance of SIRE using a stoplist and with or without stemming.

Figures 6 and 7 show the performance of SIRE on the different transcripts. It is interesting to note that the use of stemming consistently improves the performance. This result reinforce the analysis and the conclusions reported in the previous section with regards to the effect of stemming. As already noticed, the effect of stemming is stronger on the speech recognised data that on the hand transcribed.

Figure 8 show a comparison of the performance of SIRE on the four sets of data. It is surprising to see that both Abbot and Merged produce results that are as

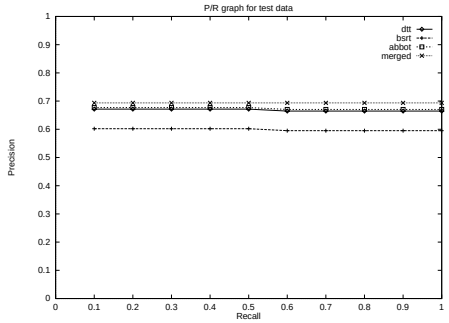

Figure 8: Comparison of the P/R performance for on the test data.

Data	*Mean Run Length*	*#1 Hits*	*Hits in top ten*
DTT	8.58	24	42
BSRT	18.34	22	40
Abbot	45.02	29	40
Merged	17.37	30	42

Table 5: Comparison of the performance as number of hits.

good than the DTT. In particular, we were surprised at the performance of the merged strategy, a strategy so simple but still consistently better than BSRT and Abbot. Knowing that both Abbot and Merged are far from perfect recognition of the speech data, as their differences with DTT show, we are tempted to conclude that SIRE is immune from some of these recognition errors. This is particularly true in the case of the use of stemming.

However, Table 5 gives another view of these results. In the second column it shows how far down we have to go in the document ranking to find the relevant document (the mean run length). In the third and fourth column it shows, respectively, the number of queries for which we have the relevant document as first document in the ranking (the so called #1 hits), and the number of queries for which we have the relevant document in the first 10 documents in the ranking (hits in top ten). As it can be seen, while SIRE performs on average far better with the DTT than with any other transcription, the #1 hits and the hits in top ten figures seem to suggest that this difference is not that high. Looking at the ranking position of the relevant documents for Abbot, for example, we can see that while SIRE performs quite well for most queries for some queries it performs really bad. This is the case of query number 15, for example, for which SIRE put the relevant document at the ranking position 1428, that is almost at the bottom of the ranking. This explain the high mean run length for Abbot. A similar situation can be found for BSRT and Merged. On the light of the data reported in Table 5 our results seem less surprising.

8.3 Official SDR results for TREC-6

Very recently we received from NIST the official results of our two submission to the SDR track, one with the PTF weighting schema and one with the merged strategy. We report here for completeness the full text of the message detailing the results.

```
Dear SDR Track Participant,

Below is the evaluation of (one of) your SDR track submission(s).
Topic 21 is completely ignored in the evaluation since its target
document was one of the ones omitted in the baseline recognizer transcripts.
All of the empty documents and the documents omitted from the baseline
recognizer transcripts were removed from all results files before
evaluation, resulting in a collection of 1451 documents.  For the two
topics that had two correct targets, the evaluation used whichever
document was retrieved first.

The format of the evaluation is as follows:

* The rank at which the known item was found for each topic for each
different run type (reference transcript, baseline recognizer, and own
recognizer).  A known item that was not retrieved (by rank 1000) was
assigned a rank of 2000.  The run tag for a run type not submitted is
'---'.

* Mean rank when found === the mean rank at which the known item was
found averaged over the topics for which the target item was found
(the smaller the number, the more effective the run)

* Mean Reciprocal === the mean of the reciprocal of the rank at which
the known item was found over the 49 topics.  For this computation, an
item that was not retrieved was assigned a reciprocal of 0.  The
benefits of this measure include: it minimizes the difference between,
say, retrieving a known item at rank 750 and retrieving it at rank
900; it incorporates a penalty for not retrieving a known item; it is
bounded between 1 and 0, inclusive; and, for those of us that are
creatures of habit, a larger value implies better performance.  Note
also that if there were exactly one relevant document per topic, the
mean reciprocal is also the average precision of the run since average
precision is the precision averaged over all relevant documents.

* A histogram showing the number of topics for which the known item
was found in a given range.  The ranges are overlapping, so all topics
that are included in the <= 5 bin are also included in the <= 10 bin,
etc.

Following the evaluation of an individual submission is a summary of
all the submissions.  This summary includes a count of the number of
```

distinct runs of each type; the minimum, maximum, and median values for the mean rank when found and mean reciprocal rank by run type; and a table giving the minimum, maximum, and median rank at which a known item was found for each topic and run type.

System:	gla6R1	gla6B1	gla6S1
1	1	1	1
2	7	4	7
3	227	200	240
4	1	1	1
5	1	7	14
6	1	2	1
7	1	2	1
8	6	28	7
9	4	2	12
10	1	13	3
11	1	2	1
12	2	2	3
13	1	1	1
14	1	2	2
15	1	1	1
16	2	1	1
17	2	3	2
18	1	5	3
19	2	2	1
20	1	2	1
22	1	1	1
23	3	110	329
24	2	1	1
25	2	1	2
26	1	1	1
27	1	1	1
28	1	1	1
29	1	1	1
30	55	61	35
31	2	1	1
32	1	1	1
33	3	12	2
34	1	1	1
35	1	1	1
36	4	2	1
37	12	3	6
38	3	38	7
39	2	5	6
40	3	1	1
41	2	1	1
42	2	273	264
43	1	1	1
44	1	3	1
45	1	1	1
46	1	1	1

47	19	58	54
48	1	1	1
49	2	8	17
50	1	1	9
Mean rank:	8.04	17.80	21.47
Mean recip:	0.6898	0.6059	0.6560
Known items found at rank:			
<= 5	43	38	35
<= 10	45	40	41
<= 20	47	42	44
<= 100	48	46	46
Not found:	0	0	0
System:	gla6R1	gla6B1	gla6S2
1	1	1	1
2	7	4	3
3	227	200	196
4	1	1	1
5	1	7	11
6	1	2	1
7	1	2	1
8	6	28	14
9	4	2	3
10	1	13	1
11	1	2	2
12	2	2	2
13	1	1	1
14	1	2	1
15	1	1	1
16	2	1	1
17	2	3	1
18	1	5	4
19	2	2	2
20	1	2	2
22	1	1	1
23	3	110	167
24	2	1	1
25	2	1	1
26	1	1	1
27	1	1	1
28	1	1	1
29	1	1	1
30	55	61	35
31	2	1	1
32	1	1	1
33	3	12	2
34	1	1	1
35	1	1	1
36	4	2	1
37	12	3	3

38	3	38	18
39	2	5	5
40	3	1	1
41	2	1	1
42	2	273	279
43	1	1	1
44	1	3	1
45	1	1	1
46	1	1	1
47	19	58	45
48	1	1	1
49	2	8	5
50	1	1	3
Mean rank:	8.04	17.80	16.94
Mean recip:	0.6898	0.6059	0.6891
Known items found at rank:			
<= 5	43	38	41
<= 10	45	40	41
<= 20	47	42	44
<= 100	48	46	46
Not found:	0	0	0

Number of distinct reference transcript runs: 15
Number of distinct baseline recognizer runs: 17
Number of distinct own recognizer runs: 13

	Reference Transcript			Baseline Recognizer			Own Recognizer		
	Minimum	Median	Maximum	Minimum	Median	Maximum	Minimum	Median	Maximum
Ave rank	3.06	8.04	18.12	10.11	17.96	36.06	6.94	18.06	229.20
Ave recip	0.5022	0.7685	0.8416	0.4287	0.6360	0.7235	0.0046	0.6560	0.8242

	Reference Transcript			Baseline Recognizer			Own Recognizer		
Topic	Minimum	Median	Maximum	Minimum	Median	Maximum	Minimum	Median	Maximum
1	1	1	8	1	1	1	1	1	21
2	1	3	34	1	1	36	1	3	2000
3	26	236	2000	41	349	2000	57	201	716
4	1	1	13	1	1	12	1	2	171
5	1	1	64	2	7	2000	1	3	2000
6	1	1	82	1	2	74	1	2	79
7	1	1	91	1	1	6	1	1	163
8	2	6	18	2	35	125	3	17	2000
9	1	5	23	1	7	2000	1	6	2000
10	1	1	2000	1	3	2000	1	3	2000
11	1	1	2000	1	3	2000	1	4	2000
12	1	3	20	1	3	24	1	3	162
13	1	1	6	1	1	3	1	1	448
14	1	1	2	1	1	12	1	1	54
15	1	1	2	1	1	2	1	1	2000
16	1	1	2000	1	1	2000	1	2	2000
17	1	1	7	1	1	22	1	1	2000

18	1	1	1	1	7	68	1	4	2000
19	1	1	10	1	2	42	1	1	2000
20	1	1	5	1	2	3	1	2	2000
22	1	1	3	1	1	2	1	1	2000
23	2	3	11	22	80	294	3	167	2000
24	1	1	2	1	1	2	1	1	2000
25	1	1	5	1	1	3	1	2	2000
26	1	1	1	1	1	1	1	1	2000
27	1	1	21	1	2	19	1	1	2000
28	1	1	5	1	1	3	1	1	2000
29	1	1	2000	1	1	2000	1	1	659
30	3	34	481	1	61	693	1	67	2000
31	1	1	2	1	1	2	1	1	2000
32	1	1	2	1	1	1	1	1	172
33	1	1	8	1	3	46	1	2	2000
34	1	1	10	1	1	9	1	1	288
35	1	1	5	1	1	3	1	1	2000
36	1	4	22	1	1	18	1	3	2000
37	1	3	80	1	1	69	1	6	2000
38	1	1	8	2	22	80	1	3	2000
39	1	2	65	2	7	2000	1	6	2000
40	1	1	4	1	1	4	1	1	2000
41	1	2	2000	1	1	2000	1	1	2000
42	1	2	49	100	314	2000	3	214	2000
43	1	1	1	1	1	128	1	1	2000
44	1	1	20	1	3	2000	1	1	2000
45	1	1	2000	1	1	2000	1	1	2000
46	1	1	2	1	1	2	1	1	381
47	1	10	19	16	42	58	3	26	333
48	1	1	217	1	1	2	1	1	425
49	1	2	42	1	1	144	1	15	2000
50	1	1	2	1	1	6	1	3	240

We will not discuss these results in detail in this paper. We will just note that:

- a few errors in the test set data were found after we performed our experimentation and send our submissions; this explain why the number of documents and queries is different in the official results (the faulty documents and queries have been removed);
- our R1 run (using the DTT data) is right on the median value;
- our B1 run (using the BSRT data) is slightly above the median value;
- our S1 run (using the PTF strategy with Abbot data) is below the median value, although it is easy to see that this is due to some very bad performance for some queries (eg. query 5, 8, 23, 49, for example);

- our S2 run (using the merged strategy) is above the median value and better that the B1 run, as we expected.

Some of these results somehow contradict some of our previous conclusions. We were expecting, for example that S1 would have better than B1 and almost at the same level than S2. We were also not expecting such good results in the R1 run, for example. We will study these results carefully before rushing to any conclusion.

9 Related work

We are not ashamed to recognise our lack of background knowledge in the SDR area. Despite our large experience in IR, with particular regards to probabilistic IR [3] and natural language processing applied to IR [16], we have never approached the SDR area. Our encouraging results will motivate us to study other approaches to this area and compare their findings with ours. For the time being we are not yet able to talk about and compare our work with related experience with confidence.

10 Conclusions and future works

This was our first experience in dealing with retrieval of spoken documents. Although we lack the necessary know-how and tools to perform speech recognition, we have a considerable knowledge of IR, in particular of probabilistic IR. We decided to use in the best possible way our available knowledge and ask the help of some other group to deal with the speech recognition. Our initial results are very encouraging and we now fell fit to work in this area and to start acquiring the necessary skills to be able to deal with the recognition and retrieval of spoken documents "in house".

Acknowledgements

The authors would like to thank the Speech and Hearing Research Group of the Department of Computing Science of the University of Sheffield (UK) for providing the recognised documents. Thanks in particular to Dave Abberley for providing us insights into how Abbot works.

Appendix: Examples of data formats

The following is a set of examples of the data formats used in the SDR TREC-6 training and test data set. All transcription files are SGML-tagged.

Sphere waveform - sph

Sphere-formatted digitized recording of a broadcast, used as input to speech recognition systems. Waveform format is 16-bit linear PCM, 16kHz. sample rate, MSB/LSB byte order.

```
NIST_1A
   1024
sample_count -i 27444801
sample_rate -i 16000
channel_count -i 1
sample_byte_format -s2 10
sample_n_bytes -i 2
sample_coding -s3 pcm
sample_min -i -27065
sample_max -i 27159
sample_checksum -i 31575
database_id -s7 Hub4_96
broadcast_id NPR_MKP_960913_1830_1900
sample_sig_bits -i 16
end_head
(digitized 16-bit waveform follows header)
.
.
.
```

Detailed TREC Transcription - dtt

LDC-produced Broadcast News transcription with absolute Section (story) IDs added.

```
<Episode Filename=file4.wav Program="NPR_Marketplace" Scribe="NIST_Reconciled"
Date="960913:1830" Version=1 Version_Date=961213>
.
.
.
```

```
<Section Type=Filler S_time=75.438250 E_time=81.214313 ID="k960913.3">
<Segment Speaker="David_Brancaccio" Mode=Planned Fidelity=High
 S_time=75.500000 E_time=81.214313>
<Expand E_form="it is">it's</Expand> friday september thirteenth
<Expand E_form="i am">i'm</Expand> david brancaccio and
<Expand E_form="here is">here's</Expand> some of
<Expand E_form="what is">what's</Expand> happening in business and the world
<Sync Time=80.883875>
</Segment>
</Section>
<Section Type=Story S_time=81.214313 E_time=207.317250 ID="k960913.4"
 Topic="Archer Daniels Midland Price-Fixing Probe">
<Segment Speaker="David_Brancaccio" Mode=Planned Fidelity=High
S_time=81.214313 E_time=107.508688>
agricultural products giant archer daniels midland is often described as
<Sync Time=85.404938>
politically well connected {breath}
<Background Type=Music Time=86.806875 Level=Low>
any connections notwithstanding the federal government is
<Sync Time=89.822500>
pursuing a probe into whether the company conspired to fix the price of a key
additive
<Background Type=Music Time=94.247437 Level=Off>
for livestock feed {breath}
.
.
.
</Segment>
.
.
.
</Section>
.
.
.
</Episode>
```

Lexical TREC Transcription - ltt

Detailed TREC Transcription with all SGML tags removed except for Episode and Section. This format is used for speech recognition scoring and can be used in SR or IR training.

```
<Episode Filename=k960913.wav Program="NPR_Marketplace"
Scribe="NIST_Reconciled" Date="960913:1830" Version=1 Version_Date=961213>
.
```

```
.
.
<Section Type=Filler S_time=75.438250 E_time=81.214313 ID=k960913.3>
it's friday september thirteenth i'm david brancaccio and here's some
of what's happening in business and the world
</Section >
<Section Type=Story S_time=81.214313 E_time=207.317250 ID=k960913.4
Topic="Archer Daniels Midland Price-Fixing Probe">
agricultural products giant archer daniels midland is often described
as politically well connected any connections notwithstanding the
federal government is pursuing a probe into whether the company
conspired to fix the price of a key additive for livestock feed
.
.
.
</Section >
.
.
.
</Episode>
```

Note that the Section (story) tags contain a short, human-generated description string designated as a "Topic". These are generated in the dtt's for the convenience of humans but may not be indexed or used in any way for the test.

Speech Recogniser Transcription - srt

Output of speech recogniser which will be used as input for retrieval. If word times are not desired, the SRT2LTT filtered version below can be used. The input for the speech recognition systems will be a set of these files sans words and the corresponding sphere-formatted waveform files.

```
<Episode Filename=k960913.wav Program="NPR_Marketplace"
Scribe="NIST_Reconciled" Date="960913:1830" Version=1 Version_Date=961213>
.
.
.
<Section Type=Filler S_time=75.438250 E_time=81.214313 ID=k960913.3>
<Word S_time=75.52 E_time=75.87>HIS</Word>
<Word S_time=75.87 E_time=75.36>FRIDAY'S</Word>
<Word S_time=76.36 E_time=76.82>SEPTEMBER</Word>
<Word S_time=76.82 E_time=77.47>THIRTEENTH</Word>
<Word S_time=77.47 E_time=77.65>I'M</Word>
<Word S_time=77.65 E_time=77.88>DAVID</Word>
<Word S_time=77.88 E_time=78.12>BRAN</Word>
```

```
<Word S_time=78.12 E_time=78.48>CAT</Word>
<Word S_time=78.48 E_time=78.56>SHE</Word>
<Word S_time=78.56 E_time=78.66>TOE</Word>
<Word S_time=78.66 E_time=78.89>HERE'S</Word>
<Word S_time=78.89 E_time=79.04>SOME</Word>
<Word S_time=79.04 E_time=79.12>OF</Word>
<Word S_time=79.12 E_time=79.30>WHAT'S</Word>
<Word S_time=79.30 E_time=79.73>HAPPENING</Word>
<Word S_time=79.73 E_time=79.84>IN</Word>
<Word S_time=79.84 E_time=80.28>BUSINESS</Word>
<Word S_time=80.28 E_time=80.38>IN</Word>
<Word S_time=80.38 E_time=80.44>THE</Word>
<Word S_time=80.44 E_time=80.82>WORLD</Word>
</Section >
<Section Type=Story S_time=81.214313 E_time=207.317250 ID=k960913.4
 Topic="Archer Daniels Midland Price-Fixing Probe">
<Word S_time=81.34 E_time=82.05>AGRICULTURAL</Word>
<Word S_time=82.05 E_time=82.49>PRODUCE</Word>
<Word S_time=82.49 E_time=82.91>GIANT</Word>
<Word S_time=82.91 E_time=83.27>ARCHER</Word>
<Word S_time=83.27 E_time=83.78>DANIELS</Word>
<Word S_time=83.78 E_time=84.20>MIDDLE</Word>
<Word S_time=84.20 E_time=84.33>IS</Word>
<Word S_time=84.33 E_time=84.59>OFTEN</Word>
<Word S_time=84.59 E_time=85.21>DESCRIBED</Word>
<Word S_time=85.21 E_time=85.35>AS</Word>
<Word S_time=85.35 E_time=85.85>POLITICALLY</Word>
<Word S_time=85.85 E_time=86.17>WELL</Word>
<Word S_time=86.17 E_time=86.95>CONNECTED</Word>
<Word S_time=86.96 E_time=87.19>ANY</Word>
<Word S_time=87.19 E_time=87.82>CONNECTIONS</Word>
<Word S_time=87.82 E_time=87.98>NOT</Word>
<Word S_time=87.98 E_time=88.15>WASH</Word>
<Word S_time=88.15 E_time=88.50>STANDING</Word>
<Word S_time=88.72 E_time=88.80>THE</Word>
<Word S_time=88.80 E_time=89.00>FED</Word>
<Word S_time=89.00 E_time=89.21>RAIL</Word>
<Word S_time=89.21 E_time=89.69>GOVERNMENT</Word>
<Word S_time=89.69 E_time=89.80>IS</Word>
<Word S_time=89.80 E_time=90.31>PURSUING</Word>
<Word S_time=90.31 E_time=90.39>A</Word>
<Word S_time=90.39 E_time=90.76>PRO</Word>
<Word S_time=90.76 E_time=90.93>INTO</Word>
<Word S_time=90.93 E_time=91.19>WEATHER</Word>
<Word S_time=91.19 E_time=91.32>THE</Word>
<Word S_time=91.32 E_time=91.74>COMPANY</Word>
<Word S_time=91.74 E_time=92.34>CONSPIRED</Word>
<Word S_time=92.34 E_time=92.42>TWO</Word>
<Word S_time=92.42 E_time=92.75>AFFECT</Word>
<Word S_time=92.75 E_time=92.85>THE</Word>
<Word S_time=92.85 E_time=93.29>PRICE</Word>
```

```
<Word S_time=93.29 E_time=93.37>OF</Word>
<Word S_time=93.37 E_time=93.46>THE</Word>
<Word S_time=93.46 E_time=93.76>KEY</Word>
<Word S_time=93.76 E_time=94.20>ALTITUDE</Word>
<Word S_time=94.27 E_time=94.41>FOR</Word>
<Word S_time=94.41 E_time=94.71>LIVE</Word>
<Word S_time=94.75 E_time=94.95>STOP</Word>
<Word S_time=95.11 E_time=95.46>FEET</Word>
.
.
.
</Section >
.
.
.
</Episode>
```

Word-Time-stripped Speech Recogniser Transcription in Lexical TREC Transcription form - ltt

This simplified form of the speech recogniser transcription can be used for retrieval if word times are not desired.

```
<Episode Filename=k960913.wav Program="NPR_Marketplace"
Scribe="NIST_Reconciled" Date="960913:1830" Version=1 Version_Date=961213>
.
.
.
<Section Type=Filler S_time=75.438250 E_time=81.214313 ID=k960913.3>
HIS FRIDAY'S SEPTEMBER THIRTEENTH I'M DAVID BRAN CAT SHE TOE HERE'S
SOME OF WHAT'S HAPPENING IN BUSINESS IN THE WORLD
</Section >
<Section Type=Story S_time=81.214313 E_time=207.317250 ID=k960913.4
Topic="Archer Daniels Midland Price-Fixing Probe">
AGRICULTURAL PRODUCE GIANT ARCHER DANIELS MIDDLE IS OFTEN DESCRIBED AS
POLITICALLY WELL CONNECTED ANY CONNECTIONS NOT WASH STANDING THE FED RAIL
GOVERNMENT IS PURSUING A PRO INTO WEATHER THE COMPANY CONSPIRED TWO AFFECT
THE PRICE OF THE KEY ALTITUDE FOR LIVE STOP FEET
.
.
.
</Section >
.
.
.
</Episode>
```

References

[1] W.S. Cooper. Some inconsistencies and misnomers in probabilistic information retrieval. *ACM Transactions on Information Systems*, 13(1):100–111, 1995.

[2] D.R. Cox. *Analysis of Binary Data*. Methuen, London, UK, 1970.

[3] F. Crestani, M. Lalmas, I. Campbell, and C.J. van Risbergen. Is this document relevant? ...probably. A survey of probabilistic models in information retrieval. *ACM Computing Surveys*. In print.

[4] W.B. Croft and D.J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35:285–295, 1979.

[5] W.B. Frakes. Stemming algorithms. In W.B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: data structures and algorithms*, chapter 8. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1992.

[6] D. Harman. Relevance feedback and other query modification techniques. In W.B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: data structures and algorithms*, chapter 11. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1992.

[7] D. Harman. Overview of the first TREC conference. In *Proceedings of ACM SIGIR*, pages 36–47, Pittsburgh, PA, USA, June 1993.

[8] H.P. Luhn. A statistical approach to mechanized encoding and searching of library Information. *IBM Journal of Research and Development*, 1:309:317, 1957.

[9] M.E. Maron and J.L. Kuhns. On relevance, probabilistic indexing and retrieval. *Journal of the ACM*, 7:216–244, 1960.

[10] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Mateo, California, 1988.

[11] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[12] S.E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304, December 1977.

[13] S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, May 1976.

[14] G. Salton. *Automatic information organization and retrieval.* Mc Graw Hill, New York, 1968.

[15] G. Salton and M.J. McGill. *Introduction to modern Information Retrieval.* McGraw-Hill, New York, 1983.

[16] M. Sanderson. System for information retrieval experiments (SIRE). Unpublished paper, November 1996.

[17] M. Sanderson. *Word Sense Disambiguation and Information Retrieval.* PhD Thesis, Department of Computing Science, University of Glasgow, Glasgow, Scotland, UK, 1996.

[18] C.J. van Rijsbergen. *Information Retrieval.* Butterworths, London, second edition, 1979.

[19] I.H. Witten, A. Moffat, and T.C. Bell. *Magaging Gygabytes: compressing and indexing documents and images.* Van Nostrand Reinhold, New York, USA, 1994.