Cluster-based Model Fusion for Spontaneous Speech Retrieval

Muath Alzghool and Diana Inkpen
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5
{diana,alzghool}@site.uottawa.ca

ABSTRACT
In this paper we present a new method for combining the results of different models in order to improve the performance on a difficult task: Information Retrieval from spontaneous speech. Our technique is based on clustering the training topics according to their tf-idf (term frequency-inverse document frequency) properties, and selecting the best models for each cluster. When the system runs on a test topic, the cluster of the topic needs to be determined and the combination of models of this cluster is used. We report significant improvement on the Malach test collection used at CLEF-CLSR 2007. We also include a comparison of the results of our method on automatic speech transcripts versus manual meta-data.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval.

General Terms
Algorithms, Experimentation.

Keywords
Searching spontaneous speech transcriptions, model fusion.

1. INTRODUCTION
Conversational speech such as recordings of interviews or teleconferences is difficult to search through. The transcripts produced with Automatic Speech Recognition (ASR) systems tend to contain many recognition errors, leading to low Information Retrieval (IR) performance[11] unlike the retrieval from broadcast speech, where the lower word error rate did not harm the retrieval [5].

Users tend to express their queries in various ways: sometimes they use more general terms, sometimes more specific terms, or a combination of both. IR systems need to be able to accommodate this variety of user needs; there is also variation among the collections (if it is a special collection like the one we use or a general collection the news collection). Some retrieval models or weighting schemes perform better when the queries are general, others perform better when the queries are more specific, and others when a combination is available. In this paper we are looking for a system that will perform well in all these cases.

There are two solutions to this problem. One solution is to fuse the retrieval results of many available weighting schemes with a reasonable weight for each scheme chosen on the training data. Alzghool et al. [1] proposed a model-fusion technique to fuse the results of 15 weighting schemes. This system outperformed the other systems tested on the Malach collection. This system has a drawback with regard to the running time, because it is takes a long time to run 15 weighting schemes for each query, and then to fuse the results.

In this paper we propose a second solution, that selects a smaller number of weighting schemes according to the query type, and then it fuses the results from those weighting schemes. The experiments that we will present show that having not more than 7 weighing schemes for each query type works better or as well as the fusion of the 15 weighting schemes.

We explore the idea of combining the results of different retrieval strategies, according to characteristics of the user's query. We propose a novel data fusion technique for combining the results of different IR models. We choose a feature based on tf-idf (term frequency-inverse document frequency) that allows us to cluster the training queries/topics from the collection. Then we select the best weighting schemes for each cluster as a combination of the best scheme for each of the topics from the cluster. Later on we use this feature to classify the test topics into the appropriate clusters and run the corresponding combination of weighting schemes.

We applied our data fusion techniques to the Malach collection [10] used in the Cross-Language Speech Retrieval (CLSR) task at Cross-Language Evaluation Forum (CLEF) 2007. See Section 5 for a brief description of the collection.

The remainder of this paper is organized as follows: Section 2 is pointing to the most important work in model fusion, Section 3 describes the two IR systems that we used to provide candidate weighting schemes for our model fusion technique. Section 4 describes how we cluster the topics according to tf-idf feature. Section 5 describes the model fusion. Section 6 outlines the CLEF CL-SR test collection. Section 7 presents our experimental results. We discuss in Section 8.1 a comparison and analysis of manual summaries and keywords vs. automatic transcripts and Section 8.2 discuss how the results could be improved. Finally, Section 9 presents conclusions and future work.

SSCS'08, July 24, 2008, Singapore.
ACM 978-1-60558-164-4/08/07.

2. RELATED WORK

Model fusion combines the results from multiple retrieval models. Since different models may have different strengths, combining information extracted by multiple retrieval models can bring performance improvements. Fusion of retrieval results from different models for improving retrieval performance has been reported in works like [3, 6, 8, 9, 15, 16]. Retrieval results from different systems [15] or retrieval results using different document representations [6] were fused together for performance improvement. There were also several approaches for the multi-model fusion (e.g. summation, maximum of, minimum of) investigated [15]. In general, a linear combination of the retrieval results was found to be the simplest and most effective way for fusing multiple information sources to improve retrieval performance.

3. SYSTEM DESCRIPTION

The weighting schemes for our fusion system were provided by two IR systems: SMART [4, 14] and Terrier [2, 12] .

SMART was originally developed at Cornell University in the 1960s. SMART is based on the vector space model of IR. We use the standard notation from SMART: the weighting scheme for the documents, followed by dot, followed by the weighting scheme for the query, where the schemes are abbreviated by the type of normalization (n means no normalization, c cosine, t idf, l log, etc.). We used the nnc.ntc, ntc.ntc, lnc.ntc, ntn.ntn, lnn.ntn, ltn.ntn, lsn.ntn weighting schemes [4, 14]. We chose these schemes because they performed well on the training data.

Terrier was originally developed at the University of Glasgow. It is based on Divergence from Randomness models (DFR) where IR is seen as a probabilistic process [2, 12]. We experimented with all the weighting schemes implemented in Terrier (BB2, BM25, DFR_BM25, DFRee, DLH13, DLH, IFB2, In_expB2, In_expC2, InL2, PL2, LemurTF_IDF, and TF_IDF).

4. FEATURES AND CLUSTERING

One important issue is what query features to consider when clustering the training topics. Once we decided on what are the clusters into which we arrange the training topics, we will select the best weighting schemes for each cluster and fuse them. When the system runs on a test topic, it will determine into which cluster to categorize the new topic, and it will run the combination of weighting schemes that was previously determined for that cluster.

Looking at the experimental results for the weighting schemes mentioned in section 3 on the training topics, we noticed a variation between the weighting schemes performance according to topics. Each of the weighting schemes performs better than other weighting scheme on some topics; moreover, the best weighting scheme in terms of Mean Average Precision (MAP score) on the training queries (DFree) is not the same as the best weighting scheme on the test topics (nnc.ntc). These two observations guide us to propose a new model fusion method that performs better than every single weighting scheme across all the topics. We hypothesize that there are clusters of topics so that each cluster prefers some specific weighting schemes. To prove this we have to find features that will allow us to cluster the training topics.

Our proposed feature is based on the tf-idf values of the terms in the queries. This feature has four parts that weight each term in the query[1]:

The term frequency in the collection, that is in all documents, (tf_c), which can be calculated by formula 1, where TF is the term frequency in the document collection (how many times the term occurs in the collection), DF is the document frequency (how many documents the term occurs in), and $Max_{tfc}(q)$ is the maximum tf_c for any term in the query. We divide by $Max_{tfc}(q)$ to normalize the values, and we multiply by the $\log(Max_{tfc}(q))$ to increase the weight for terms that appeared more frequently in the document. The intuition behind this part is that the more often the term appears in the document, the more important the term is.

$$tf_c = \frac{TF/DF}{MAX_{tfc}(q)} * \log(MAX_{tfc}(q)) \quad (1)$$

The inverse document frequency (idf), which can be calculated by formula 2, where N is the total number of documents in the collection, and DF is the document frequency. The intuition behind this part is to include the discrimination power of each term, i.e., a term that appears in fewer documents is more discriminant.

$$idf = \log(\frac{N}{DF}) \quad (2)$$

Term frequency of the term in the query (tf_q): which can be calculated by formula 3, where tf is the term frequency in the topic, and $MAX_{tfq}(q)$ is the maximum tf in the topics. We divide by $MAX_{tfq}(q)$ to normalize the value. The intuition behind the term frequency part is that the more often the term appears in the topic, the more important the term is.

$$tf_q = \frac{tf}{MAX_{tfq}(q)} \quad (3)$$

The length normalization part, which can be calculated by formula 4, represents the total number of terms in the topic. We use this part in order to get an average value for all the terms in the topic.

$$len = \frac{1}{\sum tf} \quad (4)$$

Then the feature weight (FW) of the query is calculated by formula 5, which is the summation for each part of the feature for each term in the topic.

$$FW = \sum_{for\ each\ term\ in\ topic} tf_c * idf * tf_q * len \quad (5)$$

After calculating the feature weight FW for each training topic, it is time to cluster the topics. We use one of the most popular clustering techniques, the K-Mean method. For this method we have to decide how many clusters we are looking for. Therefore

[1] We experimented with several formulas and this one was the best on the training data.

we tried different numbers of clusters, and we chose 15 because the output of the clustering method gave us clusters with a dmaximum size of 7, which is a reasonable number of weighting schemes to fuse, assuming that each cluster prefers 7 weighing schemes at most.

5. MODEL FUSION

Our model fusion formula is a modified version of the method proposed by [15]; their method, called combMNZ, sums up all the scores of a document multiplied by the number of non-zero scores of the document, as in formula 6:

$$combMNZ = \sum_{i \in IR\ schemes} score_i * n \qquad (6)$$

where $score_i$ is the similarity score of the document for the weighting scheme i which retrieved this document, and n is the number of non-zero scores of the document.

Since there are different weighting schemes from different systems, these schemes will generate different ranges of similarity scores, so it is necessary to normalize the similarity scores of the document. Lee [8] proposed a normalization method by utilizing the maximum and minimum scores for each weighting scheme as defined by formula 7.

$$NormalizedScore = \frac{score - MinScore}{MaxScore - MinScore} \qquad (7)$$

For each cluster of topics, as described in section 4, there are some weighting schemes preferred by the cluster, and these weighting schemes have different MAP scores. For that reason we adapt combMNZ to carry a weight for each weighting scheme in the cluster. Our cluster-based fusion model uses a fusion formula that we call WCombMNZ represented by formula 8.

$$WCombMNZ = \sum_{i \in IR\ schemes} W_{ik} * Normalizedscore_i * n \qquad (8)$$

where W_{ik} is a precalculated weight associated with each weighting scheme's results in the cluster k, n is the number of non-zero scores of the document, and the $NormalizedScore_i$ is calculated by formula 7 as described before.

The weight (W_{ik}) for each weighting scheme is calculated based on the MAP score for each cluster on the training data, reflecting how much its cluster prefers this weighting scheme, using formula 9.

$$W_{ik} = \begin{cases} 1 & \text{if the weighting scheme k has the max MAP for at least two topics in the cluster} \\ 1 & \text{if the weighting scheme k has the max median MAP for all the topics in the cluster k} \\ 0.1 & \text{otherwise} \end{cases} \qquad (9)$$

Basically, our model fusion with this particular weights allow the best weighting scheme to contribute the most, and the others to support it by two contributions: the first one is a small factor (0.1) of the normalized score of the document, and the second one helps re-rank the document proportional to the number n of the non-zero scores of the document. The intuition behind using 1 as a weight for some weighting schemes is that in some clusters there is more than one topic that prefers a particular weighting scheme; this is a strong indication that in these cluster this weighting scheme is one of the best in the cluster. Another case is when each topic in the cluster prefers a different scheme. In this case we select the weighting scheme with the maximum median MAP score among the topics in the cluster to have the weight one. The reason for selecting the median, not the mean, is because the median is less sensitive to the extreme MAP scores and a better indicator for smaller sample size, while the mean is often used with larger sample.

Our cluster-based model fusion differs from other works in the literature in that we fuse the retrieval results based on clusters of weighting schemes, and in the way we weight each weighting scheme for each cluster.

6. THE CLEF CL-SR TEST COLLECTION

This section describes the data that we used. The Malach collection contains 8104 “documents” which are manually-determined topically-coherent segments taken from 272 interviews with Holocaust survivors, witnesses and rescuers, totaling 589 hours of speech. Two ASR transcripts are available for this data, in this work we use the ASRTEXT2006B field provided by IBM research with a word error rate of 25%. Additional metadata fields for each document include: two sets of 20 automatically assigned keywords determined using two different kNN classifiers (AK1 and AK2), a set of a varying number of manually-assigned keywords (MK), and a manual 3-sentence summary written by an expert in the field. A set of 63 training topics and 33 test topics were generated for this task. The topics provided with the collection were created in English from actual user requests. Topics were structured using the standard TREC format of Title, Description and Narrative fields. For cross-language experiments, the topics were translated into Czech, German, French, and Spanish by native speakers. Relevance judgments were generated using search-guided procedure and standard pooling methods. See [10] for full details of the collection design.

7. EXPERIMENTAL RESULTS

We applied the K-mean clustering method on the 63 training topics. 15 clusters were produced based on tf-idf values, as described in section 4. For each cluster, from 7 runs produced by SMART and 13 runs produced by Terrier, the best weighting scheme for each topic was selected based on its MAP score. The weight for each weighting schemes for each cluster was calculated based on the MAP score, as described in section 5. After that we applied the data fusion method for the best weighting schemes of the particular cluster, as described in section 5. Maximum 7 weighting scheme was fused for each cluster because there were maximum 7 topics in each cluster. Then each test topic was classified based on its tf-idf value into one of the 15 clusters previously-produced based on the training data and the data fusion formula for the cluster was applied.

We conducted three types of experiments, based on the fields which were indexed. In the first one, the automatic transcripts (ASRTEXT2006B), and two automatic keywords (AK1 and AK2) were used for indexing the documents; we call this experiment Auto. In the second experiment, we indexed the manual keywords and the manual summaries for each document; we named this experiment Manual. In the last experiment we indexed the automatic transcripts, the two automatic keywords fields, the manual summaries, and the manual keywords, we call this experiment Auto+Manual. The title and description fields from each topic are used as query. Table 1 shows some statistics about each experiment. One interesting observation is that the number of terms (distinct words) in the manual fields is about half of the number of terms in the automatic fields. The number of tokens (total number of words) in the manual fields is about 16% of the number of tokens in the automatic fields. The average term frequencies are 39, 125, and 125 for Manual, Auto, and Auto+Manual, respectively. This ratio is very high, about four times more in the Auto fields. We also note that combining Auto and Manual brings about 14% of the terms to the Auto+Manual list of terms, which means that there is more information in the combined fields.

Table 1. Some statistics about the number of terms and the number of tokens for the three experiments.

	Number of terms	Number of tokens
Auto	13,605	1,711,684
Manual	7,131	278,717
Auto + Manual	15,884	1,990,401

Experiments on the 63 training topics using 20 weighting schemes form SMART and Terrier showed a variation between the weighting schemes performance according to topics. Each of the weighting schemes performs better than other weighting schemes on some topics. This observation guided us to propose the new model fusion technique described in section 5. Figure 1 illustrates this observation, by showing how many topics preferred by each weighting scheme.

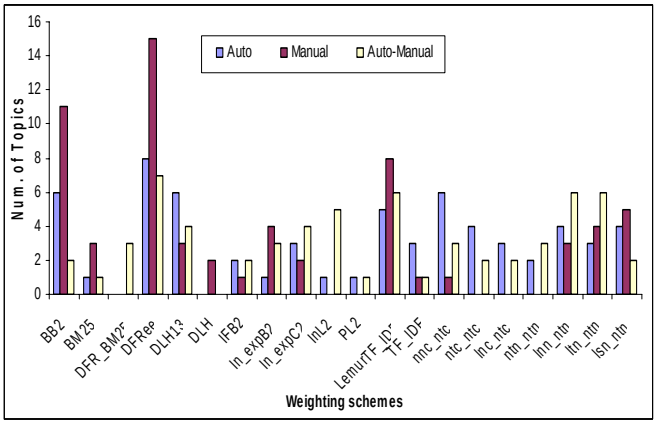

Figure 1. Variation between the weighting schemes performance according to topics, on training data.

Our experiments showed a strong relation between the feature weight (based on tf-idf) for each topic and the performance of the topics (measured as MAP score). When the value of the feature weight increased among the clusters, the maximum summation of the MAP score increased as well. Figure 2 shows the relation between the clusters and the maximum summation of the MAP score; as we see the histogram is negatively skewed, which means the smaller values are to the left and the larger values are to the right, so the maximum summation of the MAP score is increasing, and the feature weight between clusters is increasing as well, i.e. topics in cluster 1 have lower feature weights than topics in cluster 5. This proves our claim that the proposed tf-idf feature is a very good feature to cluster the topics.

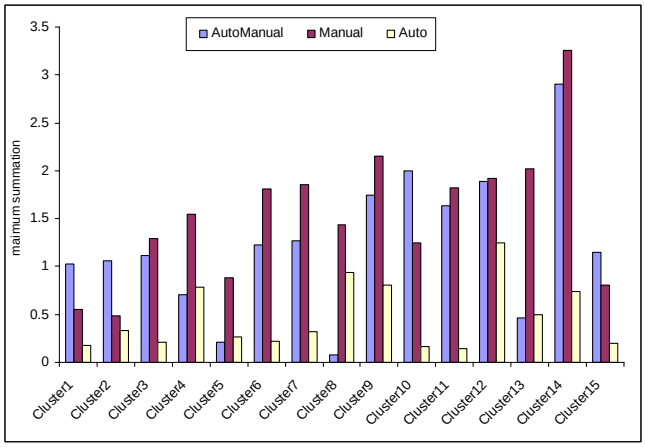

Figure 2. The relation between the clusters and the maximum summation of the MAP score.

Performance results for each single run and fused runs are presented in Table 2. The results are presented in the format MAP score, R-Precision, and number of relevant documents retrieved. In the table, % change is given with respect to the run that was best on a single model on the training data and the one on the test data.

We can conclude that cluster-based model fusion helps to improve the MAP score on the held-out test data. The improvement is statistically significant comparing to all individual weighting schemes, based on a one-tailed Wilcoxon signed rank test with ($p < 0.05$), except for nnc.ntc, In_expB2, and nnc.ntc for Auto, Manual, and Auto+Manual, respectively, for which the improvement was only statistically significant with $p<0.1$. Moreover, the results was significantly better ($p<0.05$) comparing to the best weighting schemes on the training data (Dfree). It is very important to compare with the best system on the training data because the researchers often select the system based on the training data. The best improvement using the cluster-based model fusion was on the Auto experiments with 9%, 22% relative changes comparing to the best system on the test data and the training data, respectively. Also, there is an improvement in the number of relevant documents retrieved (Recall) and R-Precision for all the experiments (see in Table 2). This supports our claim that data fusion improves the recall by bringing some new documents that were not retrieved by all the runs. Moreover, the improvement on MAP score means that the

data fusion method gives a better ranking for the documents in the list. One very important observation is that the best weighting scheme on the training data is not the best weighting scheme on the test data. For example for the Auto experiment, DFree was the best on the training data, and nnc.ntc on the test data. In general, the data fusion helps, because the performance on the test data is not always good for weighting schemes that obtain good results on the training data, but combining models allows the best-performing weighting schemes for each cluster to be taken into consideration.

Experiments show that our cluster based model fusion performs better than the individual weighting schemes for different levels of recalls. Figures 3, 4, and 5 show Precision-Recall graphs for 11 levels of recall for the three experiments: Auto, Manual, and Auto+Manual, respectively, in order to compare our model fusion method with the best weighting scheme on the training data and on the test data.

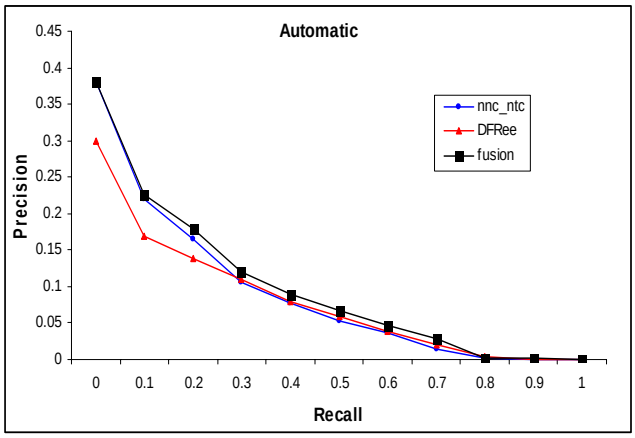

Figure 3 Precision-Recall graph for 11 levels of recall for Auto experiment.

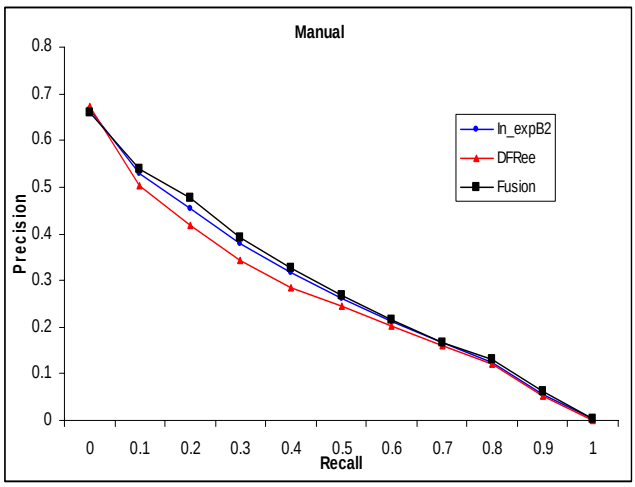

Figure 4 Precision-Recall graph for 11 levels of recall for Manual experiment.

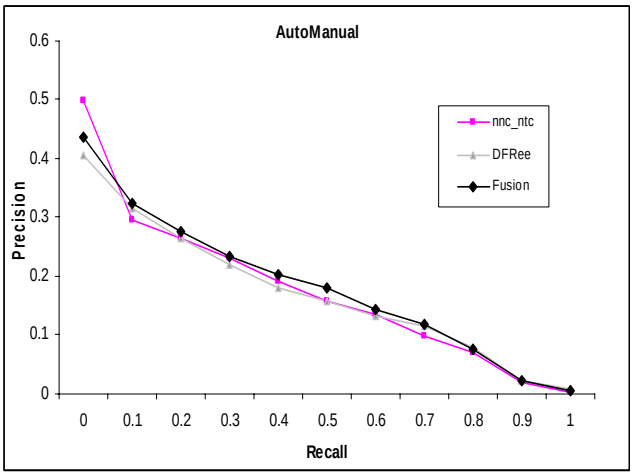

Figure 5 Precision-Recall graph for 11 levels of recall for Auto+Manual experiment.

We can compare our new method with the results of other IR systems on the same test set (using the 33 English test queries and the automatic transcripts – the required run for the CLSR task at CLEF 2007). For this setting we obtained a MAP score of 0.0849. This result was approximately the same as the best system proposed by Alzghool and Inkpen [1] (the MAP score was 0.855). It can be considered better because this system has a drawback with regard to the running time, because it is takes a long time to run 15 weighting schemes for each query, and then to fuse the results. Moreover, our system is better than the other 4 systems that participated in the task [13], as reported in Table 3.

Table 3. Results for our system and the 5 teams that participated in the CLSR task at CLEF 2007, on the English test queries.

Submitted run	MAP score
our system	0.0849
UO	0.0855
DCU	0.0787
BLLIP	0.0785
UC	0.0571
UVA	0.0444

8. DISCUSSION

8.1. Manual Summaries and Keywords versus Automatic Transcripts

Experiments on manual keywords and manual summaries (Manual) available in the test collection showed high improvements over automatic transcripts and automatic keywords (Auto). The MAP score jumped from 0.0849 to 0.2801 on the test data. Also, if we indexed the Manual fields and the Automatic fields together (Auto+Manual), the MAP score jumped to 0.1671 but it is far from the results on the Manual. This was also the case in the systems that participated in CLEF-CLSR. We are looking for a justification of why the difference is so big between the results of the Auto experiment and the Manual experiment, and why when we merge the Auto with Manual we do not reach the performance of the Manual fields. Since there are no manual transcripts available for the segments, we cannot know how the word error rate (WER) affects the retrieval.

In our view there are four factors that may affect the retrieval. The first factor is related to the nature of the summary and manual keywords; these fields were generated by experts, for example the manual summary is three-sentences long on average and answers four main questions: who? what? when? and where?. So, the summary is a very concise representation of the segments.

The second factor is how the automatic transcript or the manual summary covers the search terms from the training and test topics. To find out the effect of this factor we count the missing terms for each experiment in the training and test topics for title and description field. The results are shown in Table 4. We noticed that the number of the missing terms is approximately the same for Manual and Auto, and for Auto+Manual is approximately half the missing number of terms from Manual or Auto. So we cannot consider the missing term as the factor which affect the large difference in MAP score between Auto and Manual.

The third factor could be related to the ability of the search terms to discriminate among the documents. The classic discrimination measure is the idf value for the search terms. Therefore we compute the average idf for the training and test topics; the values are shown in Table 4. We can see that the average idf for Auto and AutoManual is less than for Manual. So, the topics ability to discriminate the documents in the Manual experiments is higher than for Auto or Auto+Manual.

The fourth factor is the average term frequency. It is much larger in Auto and AutoManual (125) than in Manual (39), as previously concluded from Table 1.

The last two factors are another reason to select the tf-idf as a feature to cluster the topics for our model fusion method, as we proposed in this paper.

Since the manual summaries and the automatic transcripts complement each other, each one brings new terms to the document structure as shown also in Table 1. Mixing the two fields is supposed to improve the retrieval, in theory. From the results, it is clear that simple merging them does not help. A better way to combine or fuse the two fields was addressed by [7].

Table 4. The average idf values, and number of missing search terms from title and description fields, for training (681 terms) and test (356 terms) topics

	IDF Training	IDF Test	Missing Training	Missing Test
Auto	1.22	1.08	28	8
Manual	1.75	1.74	27	9
Auto+Manual	1.22	1.05	10	5

8.2 How to Improve the Results?

One question that arises from our experiments is the following: is there any room for improvements over the results that we obtained? If yes, what are the possible ways to do this?

To answer these questions, we build an upper bound approximation for our experiments, by taking the MAP score for the best weighting scheme for each topic, then compute the average over all the topics. Results show that the upper bounds are 0.1016, 0.3082, and 0.22 for Auto, Manual, and Auto+Manual, respectively, on the test data. So, if we succeed to build a system that selects the best weighting scheme for each topic, we will get the above results. We could get more than that if our system fuses the best weighting scheme with others that perform well for that topic. So, one way to improve our system is by clustering topics in such a way that all the topics in one cluster have the same preferred weighting scheme. This would require discovering other features for clustering.

Another way to improve our system is to include more weighting schemes form different IR models, for example based on language modeling which has a very successful way to deal with missing terms from the query by using different smoothing techniques.

9. CONCLUSIONS

In this paper we explored the idea of clustering topics in order to determine the best combination of weighting schemes for each cluster. The clusters contain words with similar levels of specificity, because they are formed according the average tf-idf of the words. We showed that the improvement achieved on the training data carries on to the test data. We also explored the term distribution in manual meta-data versus automatic transcripts, in order to explain the loss of performance when using only automatic speech transcripts of spoken interviews.

In future work we plan to investigate more methods of data fusion, to remove or correct some of the speech recognition errors, and to use speech lattices for indexing (when they become available).

10. REFERENCES

[1] Alzghool, M. and Inkpen, D. 2007.Model Fusion Experiments for the Cross Language Speech Retrieval Task at CLEF 2007. In Proceedings of the Working Notes of the CLEF- 2007 Evaluation (Budapest-Hungary, 2007).

[2] Amati, G. and Van Rijsbergen, C. J. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM, New York, 2002.

[3] Bartell, B. T., Cottrell, G. W. and Belew, R. K. 1994. Automatic Combination of Multiple Ranked Retrieval Systems. In Proceedings of the Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (Dublin, Ireland, 1994). ACM/Springer, 1994.

[4] Buckley, C., Salton, G. and Allan, J. 1992. Automatic Retrieval With Locality Information Using SMART. In Proceedings of the Text REtrieval Conference (TREC-1) (1992), 59-72, 1992.

[5] Garofolo, J. S., Auzanne, C. G. P. and Voorhees, E. M. 2000.The TREC Spoken Document Retrieval Track: A Success Story. In Proceedings of the RIAO: Content-Based Multimedia Information Access (PARIS, April 12-14, 2000),

[6] Jones, G. J. F., Foote, J. T., Sparck, J. and Young, S. J. 1996. Retrieving spoken documents by combining multiple index sources. In Proceedings of the Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval (Zurich, Switzerland, 1996). ACM, 1996.

[7] Jones, G. J. F., Zhang, K., Newman, E. and Lam-Adesina, A. M. 2007 . Examining the Contributions of Automatic Speech Transcriptions and Metadata Sources for Searching Spontaneous Conversational Speech. In Proceedings of the SIGIR 2007 workshop: Search in Spontaneous Conversational Speech workshop (Amsterdam, 27 July 2007, 2007),

[8] Lee, J. H. 1995. Combining multiple evidence from different properties of weighting schemes. In Proceedings of the Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval (Seattle, Washington, United States, 1995). ACM, 1995.

[9] Ng, K. 2000. Information fusion for spoken document retrieval. In Proceedings of the Proceedings of the Acoustics, Speech, and Signal Processing, 2000. on IEEE International Conference - Volume 04 (Istanbul, Turkey, 2000). IEEE Computer Society, 2000.

[10] Oard, D. W., Soergel, D., Doermann, D., Huang, X., Murray, G. C., Wang, J., Ramabhadran, B., Franz, M. and Gustman, S. 2004. Building an information retrieval test collection for spontaneous conversational speech. In Proceedings of the Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (Sheffield, United Kingdom, 2004). ACM, 2004.

[11] Oard, D. W., Wang, J., Jones, G. J. F., White, R. W., Pecina, P., Soergel, D., Huang, X. and Shafran, I. 2007. Overview of the CLEF-2006 Cross-Language Speech Retrieval Track. CLEF2006, Springer, 2007.

[12] Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C. and Johnson, D. 2005. Terrier Information Retrieval Platform Springer, 2005.

[13] Pecina, P., Hoffmannova, P., Jones, G. J. F., Zhang, Y. and Oard, D. W. 2007. Overview of the CLEF-2007 Cross Language Speech Retrieval Track. In Proceedings of the Working Notes of the CLEF- 2007 Evaluation (Budapest-Hungary, 2007). CLEF2007, 2007.

[14] Salton, G. and Buckley, C. 1988. Term Weighting Approaches in Automatic Text Retrieval. (Information Processing and Management, 24, 5 1988), 513-523.

[15] Shaw, J. A. and Fox, E. A. 1994. Combination of Multiple Searches. National Institute of Standards and Technology Special Publication, 1994.

[16] Vogt, C. C. and Cottrell, G. W. 1999. Fusion Via a Linear Combination of Scores. (Information Retrieval 1, 3 1999), 151-173.

Table 2. Results (MAP scores, R-Precision, and number of relevant documents retrieved) for 20 weighting schemes from Smart and Terrier, and the results of the fusion methods, on the test data. In bold we marked the best weighting scheme on the test data, and underlined is the best weighting scheme on the training data (though we do not show the actual results on the training data, that we used for development).

Weighting scheme	Auto			Manual			AutoManual		
	MAP	R-Prec.	Rel.-Ret.	MAP	R-Prec.	Rel.-Ret.	MAP	R-Prec.	Rel.-Ret.
BB2	0.0441	0.0793	972	0.2699	0.3113	1826	0.0970	0.1280	1133
BM25	0.0567	0.0952	1120	0.2490	0.2911	1824	0.1404	0.1793	1381
DFR_BM25	0.0580	0.0984	1122	0.2558	0.2993	1818	0.1408	0.1815	1407
DFRee	0.0695	0.1179	1298	0.2527	0.2840	1822	0.1586	0.2056	1697
DLH13	0.0735	0.1162	1335	0.2560	0.2968	1825	0.1606	0.2078	1720
DLH	0.0719	0.1162	1325	0.2460	0.2904	1812	0.1606	0.2039	1707
IFB2	0.0605	0.1016	1080	0.2705	0.3025	1824	0.135	0.1831	1335
In_expB2	0.0657	0.1099	1259	**0.2727**	**0.3063**	**1826**	0.1537	0.2077	1581
In_expC2	0.0700	0.1144	1288	0.2704	0.3127	1826	0.1551	0.2103	1609
InL2	0.0629	0.1020	1259	0.2575	0.3000	1826	0.1521	0.1951	1570
PL2	0.0730	0.1172	1295	0.2510	0.2887	1803	0.1575	0.1991	1658
LemurTF_IDF	0.0517	0.0894	1146	0.2269	0.2674	1814	0.1319	0.1753	1425
TF_IDF	0.0651	0.1044	1302	0.2525	0.2965	1818	0.1452	0.1938	1627
nnc_ntc	**0.0779**	**0.1210**	**1270**	0.2190	0.2525	1760	**0.161**	**0.2047**	**1698**
ntc_ntc	0.0630	0.1097	1235	0.2154	0.2691	1776	0.1525	0.1994	1623
lnc_ntc	0.0722	0.1190	1269	0.2270	0.2863	1784	0.1585	0.2111	1667
ntn_ntn	0.0649	0.1161	1250	0.2140	0.2614	1792	0.1464	0.1951	1643
lnn_ntn	0.0658	0.1169	1284	0.2346	0.2808	1789	0.1527	0.2100	1684
ltn_ntn	0.0512	0.0924	1166	0.2167	0.2601	1785	0.1297	0.1810	1511
lsn_ntn	0.0426	0.0792	1028	0.1856	0.2312	1787	0.1140	0.1517	1376
Fusion	0.0849	0.1325	1353	0.2801	0.3191	1848	0.1671	0.2261	1720
%change (test)	**9%**	**10%**	**7%**	**3%**	**4%**	**1%**	**4%**	**10%**	**1%**
%change (train)	22%	12%	4%	10%	12%	1%	5%	9%	1%