LATENT SEMANTIC INDEXING BY SELF-ORGANIZING MAP

Mikko Kurimo and Chafic Mokbel

IDIAP CP-592, Rue du Simplon 4, CH-1920 Martigny, Switzerland
Email: Mikko.Kurimo@idiap.ch

ABSTRACT

An important problem for the information retrieval from spoken documents is how to extract those relevant documents which are poorly decoded by the speech recognizer. In this paper we propose a stochastic index for the documents based on the Latent Semantic Analysis (LSA) of the decoded document contents. The original LSA approach uses Singular Value Decomposition to reduce the dimensionality of the documents. As an alternative, we propose a computationally more feasible solution using Random Mapping (RM) and Self-Organizing Maps (SOM). The motivation for clustering the documents by SOM is to reduce the effect of recognition errors and to extract new characteristic index terms. Experimental indexing results are presented using relevance judgments for the retrieval results of test queries and using a document perplexity defined in this paper to measure the power of the index models.

1. INTRODUCTION

In this paper we present methods for indexing speech data which has been automatically transcribed by a speech recognizer. The goal is to be able to retrieve the relevant speech sections just by providing a natural language query. For indexing the difference between the automatically generated speech transcriptions and conventional text sources is that the word error rate (WER) in the transcriptions can be quite high and vary considerably depending on speakers and acoustic conditions. Thus, an important problem in information retrieval from spoken documents is how to extract those relevant documents which are poorly decoded by the speech recognizer.

A typical application of spoken data indexing is building a retrieval database from *broadcast news* recordings [1]. The first step is normally to separate the recognizable speech from music and other sounds that might exist between different sections or inside them. Although most of the news-readers speak rather clearly the recognition of the extracted speech sections remains still a difficult task. The vocabulary is very large, and includes many names from foreign languages. Since people already use the best available state-of-art large-vocabulary speech recognizers, decreasing the word error rate significantly to obtain a better index is very hard. The use of domain specific and adaptive *language models* (LM) might become helpful, because the amount of available up-to-date news data is increasing rapidly. The successful indexing of the decoded documents requires that the most important words characterizing each section can be extracted. Since there is generally no other information besides the recording itself, the index terms should both be recognized correctly and separated from the less important words.

Artificial Neural Networks (ANNs) are often used, if there is large collection of data available, but no exact parametric function is known for the models. ANNs play an active role both in the state-of-art speech recognition and text retrieval. For speech recognition, the best systems use either mixture Gaussian hidden Markov models (HMMs) or hybrid HMM/ANN systems [13, 5]. In hybrid systems ANNs can, for example, compute posterior probabilities for HMMs. For text data collections ANNs have been successfully used to order the data based on its semantic structures and to illustrate clusters using a low-dimensional display [9].

Latent Semantic Analysis (LSA) [7] is used for modeling text data based on semantic structures found by analyzing the co-occurence matrix of words and documents. The models project the data into lower dimensional subspaces by finding the most important structures and removing noise. LSA is often associated with Principle Component Analysis (PCA) or *Singular Value Decomposition* (SVD) by which the LSA is normally generated. In document indexing LSA is applied to find out the essential index terms to which the documents should be related.

This paper is describes a novel way to compute a LSA based index for spoken documents which can be applied even for large data collections. The paper describes also the French ASR system that has been used to decode the documents for indexing purposes. A quantitative measure based on perplexity is also proposed. At the end we give some preliminary results together with analysis and discussion for further improvements.

2. LSA, PROBLEMS AND NEW APPROACHES

LSA has traditionally been based on the idea that data is efficiently compressed by extracting orthogonal components directed so that each new component minimizes the projection error remaining from previous components. For indexing, the document collections are usually presented as a matrix A where each row corresponds to one document and each column the existence of a certain word. This representation looses the information of the word positions in the document, but it is mainly intended to answer to the indexation question, i.e. in which documents the certain word is used.

The word-document co-occurence matrix is decomposed $A = USV^T$ by SVD to find out the singular values and vectors. By choosing the n largest singular values from S we obtain a reduced space $A_n = U_n S_n V_n^T$ [7]. In this n-dimensional subspace the word w_i is coded as $x_i = u_i S_n / ||u_i S_n||$ by using the normalized row i of matrix $U_n S_n$. We can then get smoothed representations by clustering the words or the documents using the semantic dissimilarity measure $d(w_i, w_j) = x_i x_j^T$ [3].

With very large document collections like broadcast news recorded over a long time the *dimensionality of matrix A* becomes too large to handle. However, the matrix is sparse, because only a small subset of the very large vocabulary is actually used in one document, and so it can be represented as a list of just the non-zero elements. There exist efficient methods to compute the SVD for sparse matrices such as the Single Vector Lanczos iteration [4]. However, traditional LSA becomes still very difficult due to its computational complexity. The dimension can be quickly reduced by filtering out words that are not supposed to be important for LSA, like very common or very rare words. Vocabulary reduction will inevitably reduce indexing accuracy, since it is not obvious to judge the importance of the words, without analyzing the documents – some rare words can be very good to characterize a certain document. For practical purposes the index should also be easy to adapt for new data.

Because the co-occurrence matrix is sparse, there exist also some easy ways to represent it in lower dimensions. Instead of the true eigenvectors we can quickly generate random and approximately orthogonal vectors for the words and present the documents as an average vector for words it contains. In fact, by just using 100 – 200 dimensional random vectors, we can get quite a good approximation with considerably lower computational complexity [8]. The method is called *Random Mapping* (RM) [11]. By using this approximation it becomes feasible to use a very large vocabulary without dropping any words and also to expand the matrix later by adding new documents and words.

In practice, a second important problem, especially with spoken documents, is that the *documents are short and important words quite rare*. To still get meaningful distributions of the index words in the models, a careful smoothing is needed [3]. This is generally done by clustering similar documents together and using the average document vector of each cluster to represent the cluster members. The cluster vectors will also generate a smoothed representation of the documents, since they integrate the content of several semantically close documents into one model. The clusters can be interpreted as automatically selected topics based on the given document collection.

To avoid *quantization error* between the document and its nearest cluster, we can select a set of nearest clusters (or even all the clusters) to compute the smoothed mapping. For example, we can consider their weighted average based on distance, so that nearby clusters will have the strongest effect. This generalization matches well the broadcast news example, since one section can be relevant to several topics.

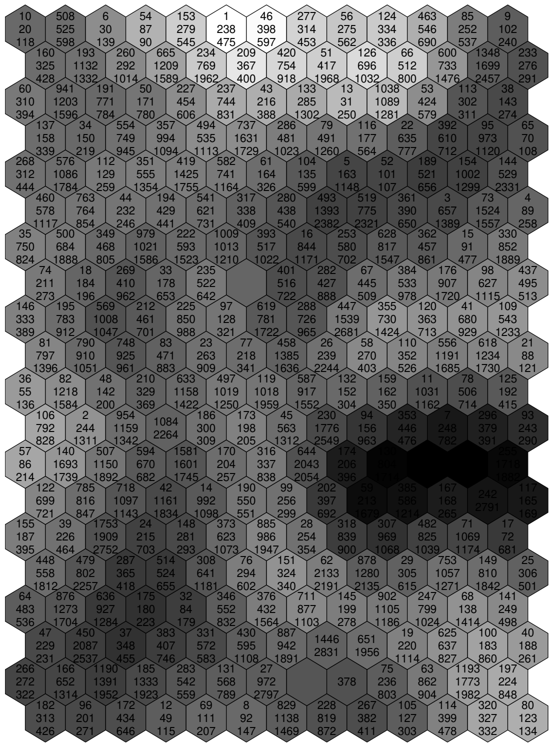

Figure 1: An example of visualizing a document collection. Each cell corresponds to one cluster (node) in the SOM grid. The lighter is the color of the cell, the shorter is the distance to the selected test document. The vectors of neighboring cells are usually also near each other in the original high-dimensional vector space. The numbers inside a cell are the ids of some of the documents closest to that cell. For visualization of the cluster contents, we can also label the clusters by the best matching index words.

For automatically decoded documents we must somehow take into account that documents are not completely described by the decoded words. Some *relevant words are lost* or substituted by fully irrelevant ones. Clustering has the advantage of mapping the decoded documents based on their whole content and in that way minimizing the effect of incorrect individual terms. In classical clustering methods such as K-means each cluster vector is the average of vectors only in that particular cluster. This adapts the clusters well to the fine structure of the data, but can make the smoothing sometimes inefficient. The more training vectors affect to each cluster, the smoother is the representation and the more will the clusters reflect the major structures of the data. If we do the *clustering by a SOM*, each training vector affects all clusters around the best one at the same time, which makes it also easier to train large cluster sets. As learning proceeds in a SOM the density of the cluster vectors will eventually reflect the density of the training vector space. This will provide strongest smoothing on sparse of areas and highest accuracy on dense areas. If we train the SOM as a two-dimensional grid, the automatic ordering will provide a visualization of the structures in the data (see Figure 1). If the display is suitably labeled, we can see the dominant clusters and directions and get immediately a conception of the area where a chosen document is situated [9].

3. EVALUATION OF INDEXING AND RETRIEVAL METHODS

The correct evaluation of a spoken document index is a difficult task itself. To find out, for example, whether a new document coding variation improves the index, we need to code the vectors, do the clustering and then create the new index. And to evaluate the index we must make test queries and rank the answers and still there is no direct way to compare the obtained document ranking lists. In the TREC evaluation organized by NIST a lot of work has been done to use *human experts* to prepare a set of test queries and select the relevant documents from a collection of broadcast news sections, respective to each query [10].

The relevance of the retrieved documents can be compared by two relative measures. The *recall* is the proportion of the relevant documents which are obtained, and the *precision* is the proportion of the obtained documents which are relevant. A meaningful comparison for ranked retrieval lists is finally to check the precision at different levels of recall or, as in this paper, by computing the *average precision* (AP) over all relevant documents. In addition to AP, we use another related measure which is the *average R-precision* (RP) defined by the precision of the top R documents, where R is the total number of the relevant documents.

If there were a direct and fully automatic way, even very approximative, to do quantitative comparisons between indexes, it would certainly be very useful as it would allow us to test the methods quickly for different new databases. In this work we have used a new concept called the *average document perplexity* to give a numerical measure of how well an index describes a document set. We will present these perplexities for different LSA variations in two databases. For the other database we have the recall-precision values as well, because this was used in latest TREC evaluation [10].

The idea of perplexity has been used in speech recognition for a long time to quantify the relative difficulty of a recognition task. One interpretation of this measure could be that the higher the perplexity, the larger is each time the set of words from which we have to select the correct one. The lower the perplexity is, the smaller is the vocabulary and the stronger are the LM constraints restricting the possible word combinations. So perplexity is a measure of the strength or *predictive power* of the LMs and it is also widely used to compare LMs when it is too expensive to compute every time the actual WER for whole speech recognition system [6]. The perplexity for the words $w_1, \ldots, w_T$ in the test set can be defined as $PP = \exp(-\sum_{i=1}^{T} \ln \Pr(w_i|\text{LM})\ /T)$.

For document models we define the perplexity using the vector space representation of words and documents so that instead of $\Pr(w_i|\text{LM})$s we have the probabilities given by the LSA model for the test document. The LSA probabilities are computed using the normalized matches between the vectors of the index terms (words or stems) and the vector of the test document (or its smoothed version). A high word match means that the word is very likely to exist in the test document and the more unlikely words there are in the test document, the higher the perplexity. Thus a higher average document perplexity means also that the models have less predictive power for the tested documents and the index might be worse. However, perplexity is by no means a substitute for the actual retrieval test and, as it is well known from speech recognition experiments, even significant improvements in perplexity do not necessarily imply improvements in the actual WER.

4. EXPERIMENTS

4.1. The system for indexing French news

A *hybrid HMM/MLP recognition system* similar to the baseline of [2] was used to decode a database (BREF) of 7.5 hours of French news. The news are selected extracts from the French newspaper, Le Monde, and read by speakers selected from a set of 80-speakers. For this SI-LVCSR task we obtained 29 % WER. The main difference between this database and true broadcast news is that the recordings are made in controlled conditions with a low noise level.

The features used in speech recognition are 12 RASTA-PLP coefficients with their Δ values and Δ energy and $\Delta\Delta$ energy. A new feature vector is computed every 10 ms using a 30 ms window. For each frame the feature vectors of 9 frames acoustic context are fed to a MLP (234 inputs, 1000 hidden units and 36 outputs corresponding to phoneme classes). The MLP was trained as suggested in [2]. The HMMs are tied to the MLP outputs using duration modeling by cloned states. The associations between phonemes and words are specified in a large pronunciation dictionary with also multiple pronunciations for some words.

News texts from the same newspaper as was used for the speech data, but on subsequent months, were used to compute the trigram LMs. The LMs, the HMMs and the pronunciation dictionary are all integrated to decode the best-matching sentence hypothesis for each sequence of MLP outputs corresponding to a section of speech. In this work we used for indexing only the decoding by the best-matching hypothesis. In fact, it would probably be very useful to pass more information from the speech recognizer, like n-best hypothesis and assign an indexing weight for each word by making use of confidence measures in the different hypothesis.

The indexing was done by creating an inverted file first using only the index words corresponding to decoded words (as in [1]), but then adding also the best matching index words according to the LSA. The number of words taken from the LSA list was determined by assuming LSA scores were normally distributed and selecting all the best scores above the 99 % significance level. The indexing was made stochastically so that the index words were weighted by the the LSA scores scaled to within [0,1]. The first index words selected directly from the actual decoding got the weight 1.0.

The documents were coded by taking a weighted average of the vectors of their words (excluding stop words, i.e. some very common words that have no value for indexing). The word vectors were 200-dimensional normalized random vectors as explained in section 2. For word weights we did not use the frequency in the current document, but took the general weights reflecting the importance of a word to the whole document collection. Importance can be defined using the mutual information or its simpler approximation, the *inverse document frequency* [12]. We took the latter approach. The same word weighting was applied as well for weighting the scores of the index terms. Instead of actually using words as index terms or to build the document terms, we used the word stems i.e. the words are mapped into their root form by suppressing suffixes like in [10].

After the LSA index is made, it can be used similarly as the default THISL index [10]. Queries are processed by eliminating stop words and mapping other words into their stems. To find the best matches, the documents are scored based on the number of matches between the query terms and the document using the index. The scores are normalized using weights for document length and the term frequency in the collection [12]. The highest ranked sections are listed with scores and pointers to the audio recordings.

4.2. Tests on the TREC test material

Index	AP	RP	PP	PP test
RM	0.33	0.34	2.6	(2.7)
RMSOM0	0.33	0.35	2.2	
RMSOM	0.34	0.36	2.1	
SVD	0.35	0.34	1.7	(1.8)
SVDSOM0	0.37	0.34	1.8	
SVDSOM	0.38	0.34	1.8	
THISL default	0.37	0.37		
“perfect”	0.43	0.41		

Table 1: Some recall-precision comparisons (AP is the average precision and RP the R-precision) between the LSA versions and with the baseline (THISL [10]) system for TREC data. The average document perplexity (PP) is provided for the LSA versions. The “PP test” is a simulation of independent test data made for each test document by ignoring the contribution of the document itself for LSA. The results here are for the S1-decodings (36 % WER).

Index	AP	RP	PP	PP test
RM	0.23	0.25	2.7	(2.7)
RMSOM0	0.25	0.25	2.2	
RMSOM	0.25	0.25	2.1	
SVD	0.24	0.23	1.6	(1.8)
SVDSOM0	0.26	0.27	1.7	
SVDSOM	0.25	0.26	1.7	
THISL default	0.29	0.29		
“perfect”	0.43	0.41		

Table 2: The same as Table 1, but using a 49 % WER speech recognizer (B2).

Since the relevance judgments were only available for the TREC database (American English broadcast news), we selected that database for quantitative and qualitative comparisons of our LSA indexing results. The indexing tests were made by using a similar index as in 4.1., but for the decoding we used two different speech recognizers with different WER levels. We selected the 36 % average WER one (S1) provided by Sheffield University [10] and the 49 % WER one (B2) provided by NIST.

In Tables 1 and 2 RM is a 200-dimensional random mapping; SVD mapping used 125 most important singular values (and vectors); SOM clustering has 260 units and SOM0 is SOM trained with 0-neighborhood

	"perfect"	S1 decoding		B2 decoding	
	decoding	def.	LSA	def.	LSA
ranked	841	893	1883	925	1425
recall	0.92	0.91	0.96	0.89	0.90
P.10	0.65	0.62	0.65	0.51	0.49
AP	0.43	0.37	0.38	0.29	0.26
RP	0.41	0.37	0.34	0.29	0.27

Table 3: Some finer details for the comparison between the reference systems and the best LSA system for Tables 1 and 2. "ranked" = average number of documents ranked per query, "recall" = total recall, "P.10" = precision at recall level 0.10. The total number of documents in the collection is 2864.

(to equal an adaptive version of the classical K-means clustering). In the clustered mappings (RMSOM, SVD-SOM) the smoothed model is made using the weighted average of 10 best-matching clusters (as explained in section 2). For the non-clustered methods (RM, SVD) we made the smoothed model from the weighted average of 20 best-matching actual document vectors.

In the "perfect" decoding we have made the index using the correct transcription, i.e. with no (automatic) speech recognition errors. The THISL default [10] is with the default parameters and without the query expansion. In the query expansion not only the index terms related to the query are checked, but also terms that are commonly associated with query terms. The common associations can be formed using any large text database related to the subjects of the indexed database. The query expansion has been shown [10] to improve the recall-precision values of the THISL default and it would probably do that for the LSA index as well.

4.3. Tests on material with no decoding errors

To see how the different LSA methods behave on other data, we made some indexes for the French newspaper data. This example is rather different from the previous one, so we can't make any general conclusions just based on these two experiments. Firstly, this experiment tests the methods on (ASR-)error-free data, so it actually is a text retrieval test only. Secondly, no relevance judgments were available, so only the document perplexities could be computed. These perplexities are a bit different from the previous, as well, because they were computed using a separate equally large document collection, so not the training data. Thirdly, no stemming was made and the language in the data is French. And finally, heavier approximations were tested in the document coding, both for RM and SVD. In RM we tested 100-dimensional random vectors and in SVD we used again 125-dimensional vectors, but reduced the original vocabulary dimension by dropping words that appeared only a few times in the collection. The Figure 2 shows the relative perplexities for 10 different LSA variations.

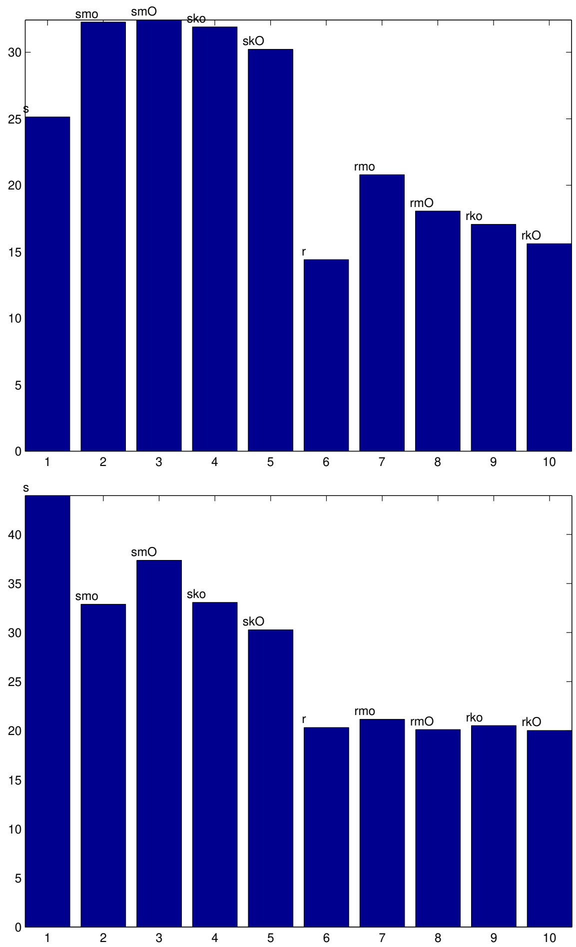

Figure 2: The perplexities of 10 different LSA indexing methods. The upper picture shows perplexities when 50 best cluster models ($k = 50$) are used for the smoothing and the lower picture when the match is approximated by only the best cluster model ($k = 1$). The explanation of the method ids: 's' = SVD, 'm' = SOM, 'o' = 260 clusters, 'O' = 2000 clusters and 'k' = SOM0.

5. CONCLUSIONS

This paper describes a system for decoding spoken documents and indexing them based on the latent semantic analysis of the document contents. A new computationally simple approach is suggested for LSA in large document collections. To smooth the LSA models we apply clustering with SOMs. This provides an organized view over the contents of the document collection. Experiments are made using French and American news and for the latter we provide the relevance analysis results of the test queries. To measure the predictive power of the models we define a new document perplexity measure.

From Table 1 we see that the average precision improves by SVD and even further when we smooth the models by SOM. The closer comparison in Table 3 shows, e.g. that LSA retrieves many more documents than the reference, including also slightly more of relevant ones. By looking at the lowest standard recall level 0.10, which gives the precision of the highest ranked documents, LSA seems also to do quite well. For higher recall levels the precision of LSA drop below the default, because the cost of the higher total recall seems to be a vast increase of irrelevant documents.

The document perplexity for random mapped documents (Tables 1 and 2) decreases as stronger smoothing is applied, but the AP and RP indicators do not show any clear improvement. For SVD coding the AP and RP indicators show improvement for smoothing, but the perplexity doesn't change much. In the other test (Figure 2, upper part) the perplexity increases a little when stronger smoothing is applied (less clusters or wider clustering kernel imply stronger smoothing). From comparison of the upper and lower parts we see that the stronger approximation ($k = 1$) increases perplexity more for the non-clustered mappings and bigger cluster amounts, which is reasonable also from the model accuracy point of view. The fact that SVD coding is here worse in perplexity than RM is most probably due to the reduced vocabulary.

From computational point of view the random mapping is better than SVD, since it is much faster and there are no complexity problems as the number of documents and words increases. It is also convenient that we do not need to change the old document vectors as the database is updated. The clustering of models is favorable, since the indexing is faster with smaller total number of models and smaller number of selected best models (k). The SOM algorithm behaves well for large document collections, because it is not affected by the vocabulary size and only almost linearly by the number of documents.

The results presented in this paper are only preliminary and more experiments are required to get sound conclusions. We used almost only some ad hoc or default values for all the parameters controlling the document coding, clustering, smoothing and indexing, as well as for ranking the retrieved documents. Careful tuning of some or all the parameters can still improve the results significantly.

For further research we have left the integration of acoustic confidence measures and n-best hypothesis into the presented stochastic index, and testing the more sophisticated importance weights for the words and index terms. For the French databases the same stemming algorithm as for English has so far been used, but because the suffixes are different, we will probably have to implement a totally new algorithm. Further development of the ranking strategies might be useful for LSA, since we get significantly more matching documents and there are also more useful information included in the indexing weights. Another interesting aspect is the use of data visualization to help understand the structures in the database and to use suitable words in queries.

6. REFERENCES

[1] D. Abberley, S. Renals, and G. Cook. Retrieval of broadcast news documents with the THISL system. In *Proc. ICASSP*, pages 3781–3784, 1998.

[2] J. Andersen. Baseline system for hybrid speech recognition on French. COM 98-7, IDIAP, 1998.

[3] J.R. Bellegarda. A statistical language modeling approach integrating local and global constraints. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 262–269, 1997.

[4] M.W. Berry. Large-scale sparse singular value computations. *Int. J. Supercomp. Appl.*, 6(1):13–49, 1992.

[5] H. Bourlard and N. Morgan. *Connectionist Speech Recognition - A Hybrid Approach*. Kluwer Academic Publishers, 1994.

[6] S.F. Chen, D. Beeferman, and R. Rosenfeld. Evaluation metrics for language models. In *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[7] S. Deerwester, S. Dumais, G. Furdas, and K. Landauer. Indexing by latent semantic analysis. *J. Amer. Soc. Inform. Sci.*, 41:391–407, 1990.

[8] S. Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Proc. IJCNN*, pages 413–418, 1998.

[9] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1997. 2nd extended ed.

[10] S. Renals, D. Abberley, G. Cook, and T. Robinson. THISL spoken document retrieval. In *Proc. Text Retrieval Conf. (TREC-7)*, 1998.

[11] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biol. Cyb.*, 61(4):241–254, 1989.

[12] S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *J. Amer. Soc. Inform. Sci.*, 27(3):129–146, 1976.

[13] P.C. Woodland, M.J.F. Gales, and D. Pye. Improving environmental robustness in large vocabulary speech recognition. In *Proc. ICASSP*, 1996.