IDIAP RESEARCH REPORT

IDIAP

Martigny - Valais - Suisse

Thematic Indexing of Spoken Documents by Using Self-Organizing Maps

Mikko Kurimo

IDIAP–RR 00-05

February 2000

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

IDIAP Research Report 00-05

Thematic Indexing of Spoken Documents by Using Self-Organizing Maps

Mikko Kurimo

February 2000

Abstract. A method is presented to provide a useful searchable index for spoken audio documents. The task differs from the traditional (text) document indexing, because large audio databases are decoded by automatic speech recognition and decoding errors occur frequently. The idea in this paper is to take advantage of the large size of the database and select the best index terms for each document with the help of the other documents close to it using a semantic vector space. First, the audio stream is converted into a text stream by a speech recognizer. Then the text of each story is represented by a document vector which is the normalized sum of the word vectors in the story. A large collection of document vectors is used to train a self-organizing map to find the clusters and latent semantic structures in the collection. Because the news stories are quite short and include speech recognition errors, the idea of smoothing the document vectors using the thematic clusters determined by the self-organizing map is introduced to get a better index. The application in this paper is the indexing and retrieval of broadcast news on radio and TV. Test results are given using the evaluation data from the TREC spoken document retrieval task.

Nomenclature

ANN	artificial neural network
AP	average precision
ASR	automatic speech recognition
B1	speech decoding by a reference ASR
HMM	hidden Markov model
IR	information retrieval
KNN	K nearest neighbor
LM	language model
LSA	latent semantic analysis
LSI	latent semantic indexing
LVCSR	large vocabulary continuous speech recognition
P10	precision at the recall level 0.10
PLP	perceptual linear predictive analysis
QE	query expansion
R1	reference (human) speech decoding
RM	random mapping
RP	R-precision
S1	speech decoding by THISL ASR
SDR	spoken document retrieval
SOM	self-organizing map
SVD	singular value decomposition
TER	term error rate
THISL	thematic indexing of spoken language
TREC	text retrieval conference
WER	word error rate
A	term-document matrix
A_k	rank k approximation of A
x_i	code vector for word w_i
n	size of vocabulary
m	number of documents
l	dimension of random vectors
k	rank of SVD
$\|\|A\|\|_F^2$	Froebius norm of matrix A
K	number of document clusters used in smoothing
C_i	the ith nearest document cluster
$g(t, d)$	smoothed projection from term t to document d
$CW(t, d)$	Okapi weight for term t in document d
λ	LSI weight
S	significance threshold for term selection
W^{idf}	word weight by inverse document frequency
W^{ent}	word weight by entropy in document collection
f_i^d	frequency of documents including word w_i
f_{ij}^w	frequency of word w_i in the document j

1 Introduction

As larger and larger audio sources become easy to access, the problem of indexing it for automatic information retrieval is getting extremely relevant. Often no written transcripts of the audio are available, and thus the indexing has to be based solely on the automatic decoding.

One example of an important and large audio source is the broadcast news. There is a continuous flow of new multimodal news data coming through many parallel channels in TV and radio in the form of video, text and pure audio. The automatic and quick management of what is happening is crucial in many business areas, not to mention the broadcasting companies themselves. The management of large archives of the past news, as well, is a strongly related and important task.

The application used in this paper is the indexing and retrieval of broadcast news from radio and TV. However, the presented methods are as well applicable for indexing other spoken audio sources. Many methods are similar as what have been used for indexing pure text sources, but there are some special characteristic features of spoken audio, and also of broadcast news, that should be taken into account.

In simple terms, the problem of indexing a document collection can be expressed as selecting the index terms and setting the pointers from them to all the relevant documents. This is most conveniently solved backwards, i.e., by scanning through the documents and extracting the relevant index terms. For a spoken document, there are several alternative ways to do this extraction. One can recognize and index the audio in the phonetic level [Ng and Zue, 1998], spot for certain keywords, or try to decode the whole speech flow into text by a very large vocabulary continuous speech recognizer (LVCSR) [Abberley et al., 1999a, Allan et al., 1998, Johnson et al., 1999]. All these approaches have their advantages and disadvantages and the selection should be made considering the nature of the indexing task at hand [Abberley et al., 1999a]. In the THISL project [Abberley et al., 1999a], which is the framework of this paper as well, the text-based speech decoding and indexing approach was chosen [Renals et al., 1998, Abberley et al., 1999a]. This means that almost all of the decoded words will contribute to the index. The main advantages of this decoding approach are that language modeling can be applied to improve the speech recognition and that allowing redundancy in index terms reduces the effect of recognition errors. Naturally, all words are not equally good for indexing, but this can be taking into account by weighting the index terms [Robertson and Jones, 1976]. When the retrieved documents are ranked, the index terms that occur frequently in the current document but are rare in general, are given more weight than the generally frequent words in the whole collection.

The proposed thematic indexing combines the baseline THISL indexing [Abberley et al., 1999a] and the latent semantic analysis (LSA) [Deerwester et al., 1990] of the document collection. LSA includes methods to represent the words and documents according to an automatically extracted semantic basis on which the indexing can be based. The aim of this combination is both to utilize all the index terms found in the document, and to give extra weight to those terms that are semantically close [1] to the document, and to pick up extra index terms according to this semantic closeness. LSA is motivated by the dimensionality and the noise reduction obtained by projecting the documents and index terms into a semantic vector space. This is also the motivation to smooth the vectors by the probabilistic clustering in the semantic space. In the clustering the documents by the self-organizing map (SOM) [Kohonen, 1997] the idea is to create a topology preserving mapping to find out the latent semantic topics (document clusters and cluster hierarchies in the LSA space). In addition to the indexing, LSA is very useful for improving the language models (LMs) of speech recognition, because it provides a way to take into account long-span characteristics, for which the traditional N-gram methods are inefficient [Bellegarda, 1997].

The second chapter of this paper explains briefly the baseline THISL broadcast news indexing system and brings out the most relevant points for the thematic indexing point of view. The third chapter presents the motivations for the current thematic indexing by LSA and probabilistic clustering. The fourth chapter explains how random mapping (RM) and SOM are applied here for dimensionality reduction and smoothing of the semantic document vectors. One chapter is devoted to the experiments and presentation of the evaluation metrics used to analyze the results. At the end of the paper there are brief discussions and conclusions.

[1] Semantic distance between a term and a document means here the distance using a suitable distance metric between the corresponding vectors in the space extracted by LSA.

2 Baseline system

This section gives a brief overview of the relevant features of the baseline speech retrieval system [Renals et al., 1998] on which the current LSI system is built. The main components of the baseline system are the HMM/ANN hybrid LVCSR and the indexing and IR.

The basis of the whole spoken document retrieval system is the speech recognition. The documents created for indexing and retrieval are formed solely from the text outputted by the speech recognizer. If the speech stream is not pre-segmented into stories [Garofolo et al., 1999], the documents can be automatically defined by merging the basic blocks which can be either overlapping word sequences of constant length in words or constant duration in seconds [Robinson et al., 1999].

Several different speech recognizers have been created in the framework of the THISL project covering British English [Robinson et al., 1999], American English [Renals et al., 1998, Abberley et al., 1999b], and French [Andersen, 1998]. The original motivation for developing the current LSI system was to make the indexing possible for the French news data, where a very high WER was expected. One of the reasons for the high WER were the lack of training data for acoustic and language models. In the TREC evaluation data (North American business news) used in this paper , we see, however, that the LSI system is not only applicable to high WER data. The results indicate small improvements in the IR precision also for better ASR outputs and even for the manually produced reference transcripts.

The speech recognizer used for decoding the TREC SDR evaluation is the Abbot LVCSR system developed at the Universities of Cambridge and Sheffield [Robinson et al., 1996] and further developed by SoftSound. It is a hybrid HMM/ANN [Bourlard and Morgan, 1994] system using a set of recurrent ANNs to compute phone posterior probabilities based on PLP features and integrating these probabilities with the statistical HMM framework. The decoding [Abberley et al., 1999b] is performed using the Chronos decoder [Robinson and Christie, 1998] with a 64K word pronunciation dictionary and large trigram LMs. The decoding (referred by S1 in section 5) performed by the THISL project partners, took approximately 3 x realtime on standard hardware.

Even having the best and highly sophisticated speech recognition system does not guarantee the success in a speech retrieval evaluation. In fact, the TREC SDR results from 1998 [Garofolo et al., 1999] and 1999 show that documents decoded by a simpler ASR can achieve better IR precisions, because of a more successful IR system. Some IR systems systems are even fairly robust for different ASR outputs ranging between 20 – 40 % WER.

The THISL IR system (*thislIR*) used as a baseline for the LSI experiments, is a so-called bag-of-words model where almost all the decoded words in a document are used as index terms. Important additional features are: The most common words (so-called stop words) are filtered away, the words are stemmed [Porter, 1980] to get rid of the inflected forms, and finally the remaining index terms are weighted relative to their frequency in the current document versus the whole collection as in [Abberley et al., 1999b]. An extra feature in the *thislIR* that was found very useful for the baseline system, is the query expansion (QE) [Xu and Croft, 1996, Abberley et al., 1999a]. It will probably be as useful for the LSI based system as well, because it brings in associations retrieved from external text databases that link certain index terms together. However, this was not yet included to the experiments presented in this paper, because after QE the differences of indexing methods are not so easily seen by the IR comparisons.

3 Latent semantic analysis and indexing

One way to automatically create a semantic [2] representation of a document is to consider the distribution of words in it [Salton, 1971]. Of course, this is only one way to interpret the semantics and the word count is a very coarse semantic model, because it excludes all the information about which words occur close each other and about the order in which the words appear. For some purposes, however,

[2]In this paper, the semantics of words and documents refer to features obtained by analyzing the distribution of words in documents.

as the indexing of the documents based on their contents, this approximative semantic representation can still be very useful.

The dimensionality of the word count vectors makes them quite difficult to handle, because the vectors are as long as the size of the whole vocabulary, which can easily be several tens or even hundreds of thousands. Another difficulty is the word noise in short documents. The noise comes both from the use of synonyms and homonyms and from the randomness of the choice of words. In decoded documents the decoding errors add the word noise, as well. Because the word count vectors are very sparse, there are several methods to easily reduce the dimensionality and the noise as, for example, the random mapping (RM) and the probabilistic clustering which are described in the next section, and the singular value decomposition (SVD).

The SVD transformation of the sparse word count vectors creates an orthogonal vector basis, where the words and documents can be projected. The basis vectors could be loosely called as “eigendocuments”, because they try to represent the typical directions in the high dimensional word distribution space. The term-document matrix A is decomposed by SVD as $A = USV^T$, to find the singular values and vectors. By choosing the k largest singular values from S we obtain a reduced space where A is approximated by the estimate A_k [Deerwester et al., 1990]

$$A_k = U_k S_k V_k^T \ . \tag{1}$$

From the characteristics of SVD it follows (Eckart and Young, see [Golub and Reinsch, 1971]) that with any first k of these basis vectors, we can represent the original vector set in the maximal accuracy in k-dimensional space according to the L_2 norm. In this k-dimensional subspace the word w_i can be coded as

$$x_i = u_i S_k / \|u_i S_k\| \tag{2}$$

by using the normalized row i of matrix $U_k S_k$. To measure the closeness of the words (for smoothing or clustering, e.g.) we can then use the simple semantic dissimilarity measure [Bellegarda, 1997]

$$d(w_i, w_j) = x_i x_j^T \ . \tag{3}$$

Because the latent semantic word and document coding (2) is based on the most important correlations in the whole document collection, the remaining (and thus reduced) correlations are probably only local and can be assumed to be noise. In this sense we can call these basis vectors a semantic basis of this document collection. This analysis to reveal the semantic directions is often called latent semantic analysis (LSA) [Deerwester et al., 1990]. Of course, there are other methods than SVD, as well, to define these latent semantics. For example, the probabilistic LSA [Hofmann, 1999] and the method based on SOMs described in the next section.

The document vectors can be made either directly from the SVD decomposition or by summing the word vectors for each document. The sum of the semantic word vectors will point to the average semantics in the word set. Thus, the effect of the erroneous words will be reduced, because it is unlikely for the words resulting from recognition errors to have any common latent semantic properties, at least, if only short-span LMs are used in ASR. This is because ASR mostly confuses words that just sound similar, but their meaning can be completely different.

After normalizing the lengths of the semantic document vectors obtained by the SVD transformation, we can measure the semantic closeness of two vectors simply by equation (3). In the same way the index term vectors can be compared to the document vectors to judge how close they are to these documents in the semantic space. This way of using the dot-products of the vectors to determine the relevance of the corresponding index terms to the documents is used in latent semantic indexing (LSI). In the original high-dimensional space, respectively, this kind of comparison of directions would equal to the basic bag-of-words index, because there the words are truly orthogonal and the dot product between a word and a document directly gives the word’s relative frequency.

The optimality of the semantic basis found by SVD can be criticized, because the L_2 norm is clearly not the optimal norm to compare the word counts. For example, there is a significant difference between having ten word occurrences against nine and one against zero. Furthermore, these comparisons do not apply similarly for rare and common words. Despite that, however, in some applications the SVD based LSI works quite well and the different word weighting functions can be as well combined with LSA to improve the performance, in practise. For certain "well-behaving" document collections it can also be proved that LSI increases IR performance by capturing the existing semantics. In [Papadimitriou et al., 1998] this is done by showing that nearly orthogonal document vectors will be assigned to different topics and nearly parallel vectors to the same topic.

3.1 Random mapping in LSI

A successful dimensionality reduction in high-dimensional sparse vectors does not need very complicated methods. For example, for a document classification application [Kaski, 1998] it was experimentally shown that using a RM of dimensionality $d >= 90$ already gives as good results as using SVD of $d = 50$. In this example the task was the separation of document topics and it was observed that the performance using SVD $d >= 50$ was close enough to the performance using the original document vectors ($d = 5781$). In applications where the computational complexity restricts severely the use of SVD, a simpler dimensionality reduction may be the most convenient way to avoid other approximations, and thus, achieving the best results. And in some applications, it is necessary to frequently add new document vectors or new words to the vocabulary, which is very simple for RM, but not for SVD. The theoretical motivation behind the RM is the result by Johnson and Lindenstrauss [Johnson and Lindenstrauss, 1984] that if points are mapped to a random subspace of suitably high dimension, then the distances between the points are approximately preserved.

There are several approaches to make the SVD run faster for big databases. For example, the Single Vector Lancsos [Berry, 1992] algorithm can be used to compute iterative approximations of SVD by taking advantage of the sparseness of the original high-dimensional term-document matrix. However, even this algorithm can get quite slow as the dimensions increase and then an acceptable solution in a feasible time is not always guaranteed.

In this work it was observed that we can as well combine RM and SVD for the LSI. Thus we can, on the other hand, compute even the normal unoptimized SVD easily and quickly for rather large document collections and, on the other hand, easily re-map the random mapped documents into a semantically meaningful basis vectors, as in normal SVD. The SVD becomes easy, because instead of an nxm matrix, where both the m (number of documents) and n (size of vocabulary) are of tens or hundreds of thousands, we only have an lxm matrix to transform. The dimension of the random vectors $l << n$ can well be just a couple of hundreds. The "trick" here is that we still get the normal, not an approximative SVD, but for an already *approximated term-document matrix*. Thus, the quality of the obtained latent semantic basis depends only on the rank of the SVD and the dimensionality of the RM, but not on any other approximations.

Random mapping (or random projection) technique can be considered as a good way of speeding up LSI, since the final representation is still very close to that given directly by the LSI [Papadimitriou et al., 1998]. In fact, the accuracy and the speed are easily controlled by adjusting the number of the random dimensions. Of course, RM is not the only way to approximate the term-document matrix for making the LSI feasible for large document collections. Another commonly used straight-forward way is to sample only a subset of the documents or of the terms or of both. The disadvantage of the sampling is that we should select the sampled documents or terms carefully to maintain a good accuracy and it is hard to predict the IR results concerning the documents and terms which are lost in sampling. The mathematical analysis of different sampling algorithms seems to be difficult [Papadimitriou et al., 1998], but at least a given computation speed-up and accuracy would probably be easier to obtain by RM than by sampling in random.

In [Papadimitriou et al., 1998] the accuracy of the LSI after RM is compared to the accuracy of the direct LSI using the measure [3] $||A - A_k||_F^2$, i.e., how much SVD with rank k recovers from the original term-document matrix A. It is shown that the matrix B_{2k} obtained by the LSI with rank $2k$

[3] The Froebius norm $||A||_F^2$ is the sum of the squares of all the matrix elements.

after RM recovers almost as much as A_k

$$||A - B_{2k}||_F^2 <= ||A - A_k||_F^2 + 2\epsilon||A||_F^2 \ . \tag{4}$$

This approximation holds for high probability, if the dimension of the RM is large enough, i.e. $l = O(\frac{\log n}{\epsilon})$ [Papadimitriou et al., 1998]. The speed-up will then be from $O(mnc)$, which is for the direct sparse SVD of c non-zero elements per row, to $O(mc\log n + m\log n^2)$ which comes from the RM ($O(mcl)$) and the non-sparse SVD ($O(ml^2)$).

When SVD is made from the random mapped term-document matrix approximation, the result gives the semantic subspace coding (2) for each random dimension, but not directly for the words. The semantic word vectors are composed from the projections of the original random dimensions to these new code vectors. The document vectors are weighted sums of the new word vectors and the smoothing can then be applied (either for word vectors, for document vectors, or for both) as in normal SVD (see Section 4.1).

4 Self-organizing maps for LSI

The motivation of using SOM for LSI is twofold. First, it offers a natural way to *smooth* the latent semantic document and word vectors in order to more reliably reflect the semantic characteristics and to reduce noise. The second motivation comes from the need to *visualize* the relations between the semantic topics in the document collection. For example, for the IR it is useful to see which topics are present in the database in general, what are the topics of the retrieved best documents, and which other topics are semantically close to them.

4.1 SOM for smoothing

Smoothing of the word and document vectors is important for applications with a lot of word noise coming from, e.g., short and high-WER document decodings. The LSA as well suffers from the word noise, because it usually has to be made using the noisy data and a database which is not large enough to provide a good statistical accuracy for the semantic representations. A practical motivation for smoothing spoken documents is that if a document is very short, it does not directly provide many relevant index terms. Smoothing can also ease the computational load of indexing, because the indexing information already computed for close-by documents or document clusters can be exploited for a new document.

A straight-forward way of smoothing is to average the K nearest neighbor documents (KNN) for each document. However, this is too slow for large document collections, if no major optimizations are made to reduce its complexity ($\mathcal{O}(m^2k)$ for k dimensional vectors). A clustering of the document vectors approximates this KNN smoothing, since the cluster centroids will act as averages of the neighboring documents. To get the mapping more continuous, the smoothed vector can also be computed by the weighted average of the K nearest clusters. Another motivation for this is the fact that as the clusters probably learn to represent well some often occurring document types of the collection, a single document can often be relevant for several categories or document topics. Thus, the smoothing by all the relevant topics should better preserve the main content of the document. The clustering is often, as well, a considerably faster operation than all the full KNN searches in the whole input data. For a SOM of s units the complexity of the smoothing is only $\mathcal{O}(mks)$ and of the training of the SOM $\mathcal{O}(ks^2)$, or even less, after some efficient approximations [Kohonen et al., 1999].

In practise, one of the main motivations to cluster by SOM instead of by using other methods such as the K-means, is the SOM's convenience of use for large data sets. It is quite robust for selecting the number of units, their initialization, the learning rate, and the amount of iterations. If there are too many units, the excess of units will most probably learn to model variation around the largest clusters, because, in general, the whole point density of the SOM units will be a function of the input density [Kohonen, 1997]. Because the training starts with a large learning area around the samples, even a random initialization will work, since all the units start quickly to follow the given input. The learning rate and the number of iterations naturally affect the quality of the resulting SOM, but the practise has shown that the differences are not very significant unless too fast learning is forced so that the SOM has no time to organize properly. The specification of the size and decay of the learning neighborhood offers a good tool to control the level of the smoothing and the mapping accuracy of the result.

SOM is used in smoothing to find the main latent topics of the collection by clustering the documents and ordering the clusters in the semantic space. Instead of indexing the document vectors by finding directly the closest and most relevant index terms, we first find the closest semantic clusters (comparing the document vector to the cluster means) and then select the index terms that are closest to these topics. The final LSA score of a document d computed for the index term t, actually approximates the probability:

$$\Pr(d|t) = \Pr(t|d)\Pr(d)/\Pr(t)\ , \tag{5}$$

where the probability of each term $\Pr(t|d)$ can be computed as the average of the K (best-matching) clusters $C_1, \ldots, C_K$ weighted by their similarity to the current document

$$\Pr(t|d) = \sum_{i=1}^{K} \Pr(t|C_i)\Pr(C_i|d)\ . \tag{6}$$

Thus, the smoothed projection $g(t,d)$ is the weighted average of projections of t to the K nearest clusters of d, where the weights are proportional to the normalized projections between d and the clusters:

$$g(t,d) = \sum_{i=1}^{K} p(t,C_i)p(C_i,d) / \sum_{i=1}^{K} p(C_i,d)\ . \tag{7}$$

The projection $p()$ here, is the dot-product similarity measure (3) in the semantic space normalized between $[0,1]$.

The indexing with the help of the clusters makes the computation of the term-document projections also a bit faster. The complexity decreases from $\mathcal{O}(mnk)$ to $\mathcal{O}((m+n)sk)$ as we only need to project the terms and documents to the clusters and not to each other.

The index is made stochastically which means that all selected index terms for a document will get a weight describing the relevance of the association. In addition to the normal frequency weight (see Section 2), the relevance weight includes the smoothed projection (7) from the term to the document. The total index weight w_{td} is determined as a convex combination [4] of the Okapi term weighting function $CW(t,d)$ [Renals et al., 1998] and of the smoothed vector projection $g(t,d)$ (7)

$$w_{td} = (1-\lambda)CW(t,d) + \lambda g(t,d)\ , \tag{8}$$

where the global LSI weight $\lambda \in [0,1]$ depends on the database. This combination can be interpreted as balancing the importance between the smoothing and the decoding. Thus, with $\lambda = 0$ this would be equal to the basic ranking [Renals et al., 1998] with no semantic weighting. To restrict the size of the created index file, only those new index terms are selected, whose projection to the document would be above the 99 % significance level in a Gaussian normalization of the projections [Kurimo and Mokbel, 1999].

As well as the semantic document vectors, the semantic word vectors can be smoothed by an SOM. The motivation is to represent more reliably rare words which are generally more affected by the word noise. Because the rare words are used only in a few documents, even a single substitution by a synonym or a decoding error can significantly change the semantic vector of the term in a document collection. The rare words can also be more difficult to decode, because of the low LM probabilities and lack of acoustic training data. But, if the words are clustered in the semantic space, the centers of the clusters will be more robust to word noise. This could be interpreted as a probabilistic grouping of index term "synonyms", i.e., clustering words that have similar existence patterns in the collection.

[4] The weight CW must be here normalized for the same range as the weight g.

4.2 SOM for visualization

The purpose of the visualization of indexing and IR results is to gain knowledge of the content and structures of the document collection and to help the user to compose better queries. This is important, because even the best LSI, smoothing, and query expansion methods can only find associations which are given in the available data. Since the IR system cannot read user's thoughts, it is sometimes more efficient, in practise, to provide some structural information about the documents in addition to just the IR results, and to let the human mind to determine *the relevant question* for the problem at hand. The visualization can give an overview of the existing topics, of their hierarchies and show the best index terms to describe them. A valuable information is also to see where and how the obtained IR results are mapped.

Having the semantic clusters and topics extracted by SOM provides an easy way to make a 2D map view of the document collection. In the SOM, the documents that are close to each other in the input space are mapped into clusters close to each other in the map, as well. Because the topics are presented by the clusters, the topics that are semantically close will also be close in the map. Thus ideally, the nearby areas in the map concern similar topics. Several other different collection characteristics can also be displayed [5] on the map [Simula et al., 1999]. Because the 2D map plane must fold quite a lot in the high dimensional input space to achieve a good mapping accuracy, it is helpful to color the map to better see which clusters really are close to each other in the input space. Perhaps the most widely used method to show the structures of SOM by coloring is the unified distance matrix (U-matrix) [Ultsch, 1999]. In the U-matrix the colors indicate height levels as in topographical maps for geography. The height levels are, however, defined relative to the neighboring units, so that the further the neighbors are from each other in the semantic space, the higher is the "mountain" between them (see Figure 1, for an example [6]). The "valleys" in the map show topics that are close to each other and the units on the high mountains are either "glue" to keep the map continuous, i.e., they are between some topics and do not describe well any of them, or just topics far away from the others.

In the LSI point of view it is interesting to select some characteristic descriptors as labels to show the contents of the map clusters (see Figure 1). Several methods exist to extract the labels automatically [Lagus and Kaski, 1999, Rauber and Merkl, 1999, Hofmann, 1999]. In this work the following method was developed to best monitor the index:

1. Do the probabilistic indexing (equation 8)
2. Find the Woronoi regions (i.e., list the mapped documents) for all clusters
3. In each cluster, sum the indexing weights for terms in the Woronoi region
4. Show the top ranked index terms for each cluster as topic labels

For viewing the whole map at once (the top level of the hierarchy), where all the labels cannot be shown, a selection method as in [Lagus and Kaski, 1999] can be used. Naturally, to label larger areas, it is also possible to just use the method described above extending the sums over the merged Woronoi regions. Of course, the use of the SOM's neighborhood function for weighting the neighboring clusters would probably find more accurate positions for the labels. As many documents probably belong to several different topics, it might be better to sum also over the second or third order Woronoi regions (i.e., lists where the documents would be mapped if the best match were ignored) with appropriate weights. However, this would probably not change much the order of top ranked descriptors.

In Figures 1 and 2 the labels shown are just the stems of the winning index terms for every 9th unit. A more sophisticated label selection would give more insight of the index, but even these rough

[5]See URL http://www.cis.hut.fi/projects/somtoolbox for freely available software implementation of the SOM algorithm with several visualization techniques.

[6]See URLs http://www.idiap.ch/ kurimo/scfig1.ps.gz and http://www.idiap.ch/ kurimo/scfig2.ps.gz for better pictures with colors.

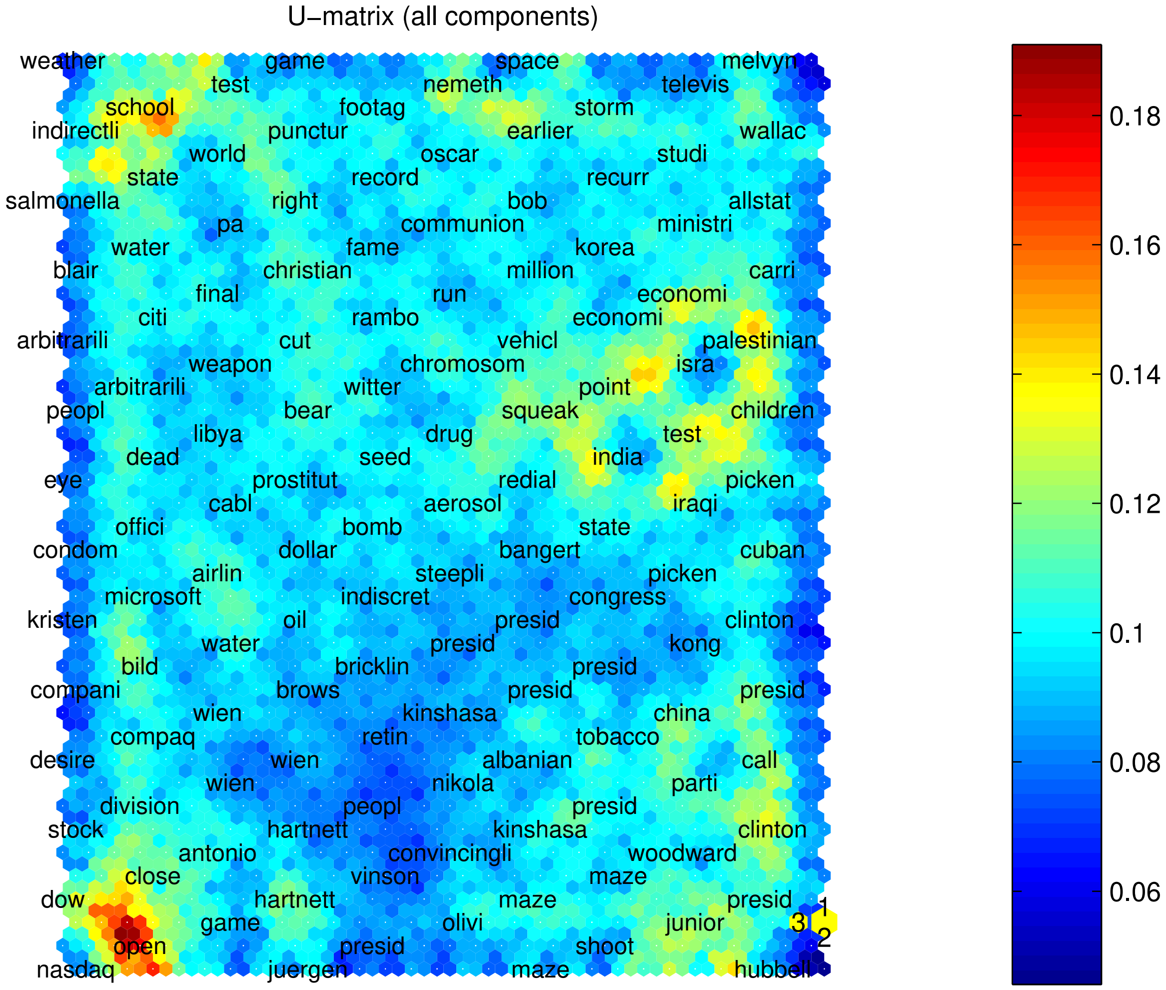

Figure 1: An example of visualizing an indexed document collection by a labeled U-matrix. There are 1200 cells corresponding to the 1200 clusters (node) of the SOM grid. The semantic vectors of neighboring cells in this 2D map are, in general, near each other also in the original high-dimensional vector space, but because the map is somewhat folded, the distances are better shown by the colors. The colder the color between the cells, the closer the neighboring cells are in the original space. The label of the cluster is selected as the stem of the index term that gets the highest total indexing weight for the documents in the cluster. For clarity of the figure only the label of every 9th cluster is shown. The numbers 1,2,3 show where in the collection map the three best-matching documents (for the given query) get mapped (see Figure 2 to zoom).

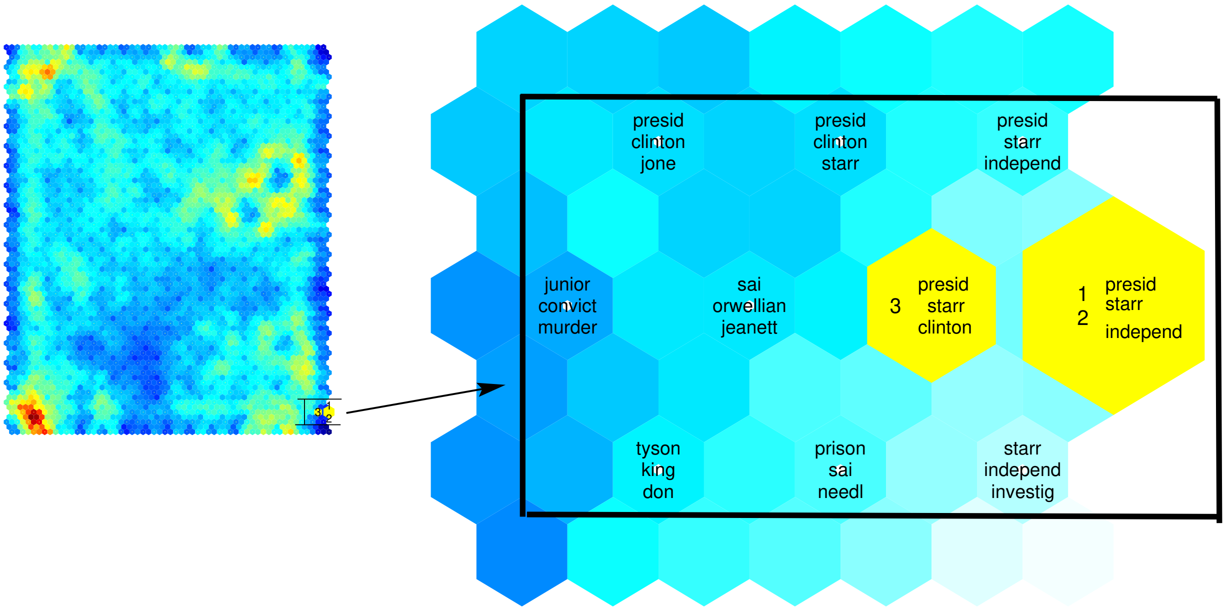

Figure 2: Displaying the latent document topics in a 2D map of hexagons. The original query was "Lewensky" (sic.). The closest map cells for the 3 best documents are shown with magnified hexagons. The topic labeling used in Figure 1 is here extended to the three best index terms.

stems already give some hints about the organization of the topics. For example, on the lower left corner, there are things related to stock markets and a bit higher up there are some company names indicating some more specific business news. Also groups related to the president and to some foreign affairs, like Israel and the Palestinian, can be seen. The Figure 2 is a detail of the Figure 1. There we see better the clusters in the close neighborhood of the 3 best matches for the query that has been made.

The hit histogram, i.e., how many documents get mapped into each area, can be added to the same display, e.g., by using dots of variable sizes [Simula et al., 1999]. Instead of U-matrix the colors in the map can optionally show the distribution of a chosen SOM component plain, i.e. how relevant the topics are just for a certain semantic dimension. Other useful distributions to show are the semantic distances from the map units to a certain query, index term or document [Kurimo, 1999].

When the document collection and the map are very big (e.g., a million nodes for a collection of millions of documents [Kohonen et al., 1999]), it is not convenient to show the whole map at once. The WEBSOM demo [7] shows an example of how to use several display hierarchies [Honkela et al., 1996, Kohonen, 1997]. There an explorer can select an interesting area from any level and zoom in or out to see the nearby topics and finally to examine the selected cluster by viewing the associated documents.

5 Experiments

5.1 Evaluation measures for SDR

All the results reported here are based on the test queries of the spoken document retrieval (SDR) tasks in the TREC-7 [Garofolo et al., 1999] and the TREC-8. Other broadcast news collections decoded by speech recognition (in French and in English) have also been indexed by the described system, but the relevance judgments of human experts were only available for the TREC evaluation tests.

[7] See URL http://websom.hut.fi for a demo.

The comparison of the spoken document indexes is not a straight-forward task. The WER of speech recognition varies a lot and it is not clear how much this affects to the correctness of the index. A better measure could be the TER (index term error rate) [Renals et al., 1998], but for IR, the significance of different terms to different documents varies a lot, as well. Perplexity of the index [Kurimo and Mokbel, 1999] can be used to measure the predictive performance of the models, as in speech recognition [Chen et al., 1998]. This involves, however, a transformation of the LSI scores into probabilities, which is not straight-forward and makes the comparison of different systems difficult [Hofmann, 1998].

A standard way to compare IR results is to use the *recall-precision curve* (see Figure 3). An index is considered to be better than another, if the precision of the retrieval results in each recall level is higher than by the other method. The *recall* is the proportion of relevant documents which are retrieved and the *precision* the proportion of retrieved documents which are relevant. Widely used scalar performance indicators obtained from this recall-precision curve are the *average precision* (AP) over all standard recall levels and the precision (RP) at the level R, where the number of retrieved documents equals to the total number of relevant documents. In Table 1 we also give the precision at the lowest standard recall level 0.10 (P10), because the top of the document ranking is is often the most relevant for practical IR applications since people usually rather revise their queries than scan through all the given answers.

5.2 Results

Two broadcast news databases with standardized evaluation queries were used for testing the proposed indexing system. The databases are the evaluation sets for TREC-7 and TREC-8 SDR tasks. The TREC-7 task has approximately 100 hours of news segmented into 3000 stories and the TREC-8 550 hours in 22000 stories. The relevance judgments by human experts are provided for the results of 23 and 50 test queries, for TREC-7 and TREC-8, respectively.

The speech recognition was done using the THISL speech recognizer, which is a specialized version of the Abbot HMM/ANN hybrid [Renals et al., 1998] (S1). Results are also given for the reference ASR decodings provided by TREC (B1) and for the reference transcripts with no ASR errors (R1). The baseline method for indexing the decoded documents was the thislIR-0.2 [Renals et al., 1998], which uses the same stemming, stop list and Okapi term weighting function as the LSI system, but indexes the documents using just the stems found in the decoding (as $\lambda = 0$ in equation 8).

Table 1 presents the results for the TREC test sets by the baseline *thislIR* (see Section 2) and the LSI+SOM method (see Section 3). We also tested how robust the LSI+SOM index is for the key parameter values. The results of these parameter variations are in Tables 2 – 5. The statistical significance of the differences in results is briefly discussed in Section 6.

		thislIR			LSI+SOM		
	WER %	AP %	RP %	P10 %	AP %	RP %	P10 %
TREC-7/S1	35.9	37.4	37	62	38.1	38	63
TREC-7/R1	-	43.4	41	65	42.9	43	64
TREC-8/S1	32.0	40.0	41	67	42.3	43	71
TREC-8/B1	27.5	40.4	41	69	42.4	43	71
TREC-8/R1	-	43.8	44	66	45.4	46	67

Table 1: Results for the indexing systems in different broadcast news sets and decodings (see section 6). Precisions at the lowest level (0.10), at level R, and in average are given (P10 %, RP % and AP %, respectively).

Table 2 concentrates on the combinations of the indexing weights given by the following two sources (see equation 8): The frequency weights from the Okapi criterion CW and the LSA scores g. First we tested different λ values. The smaller test (TREC-7) gives better APs for the smaller λs, but the

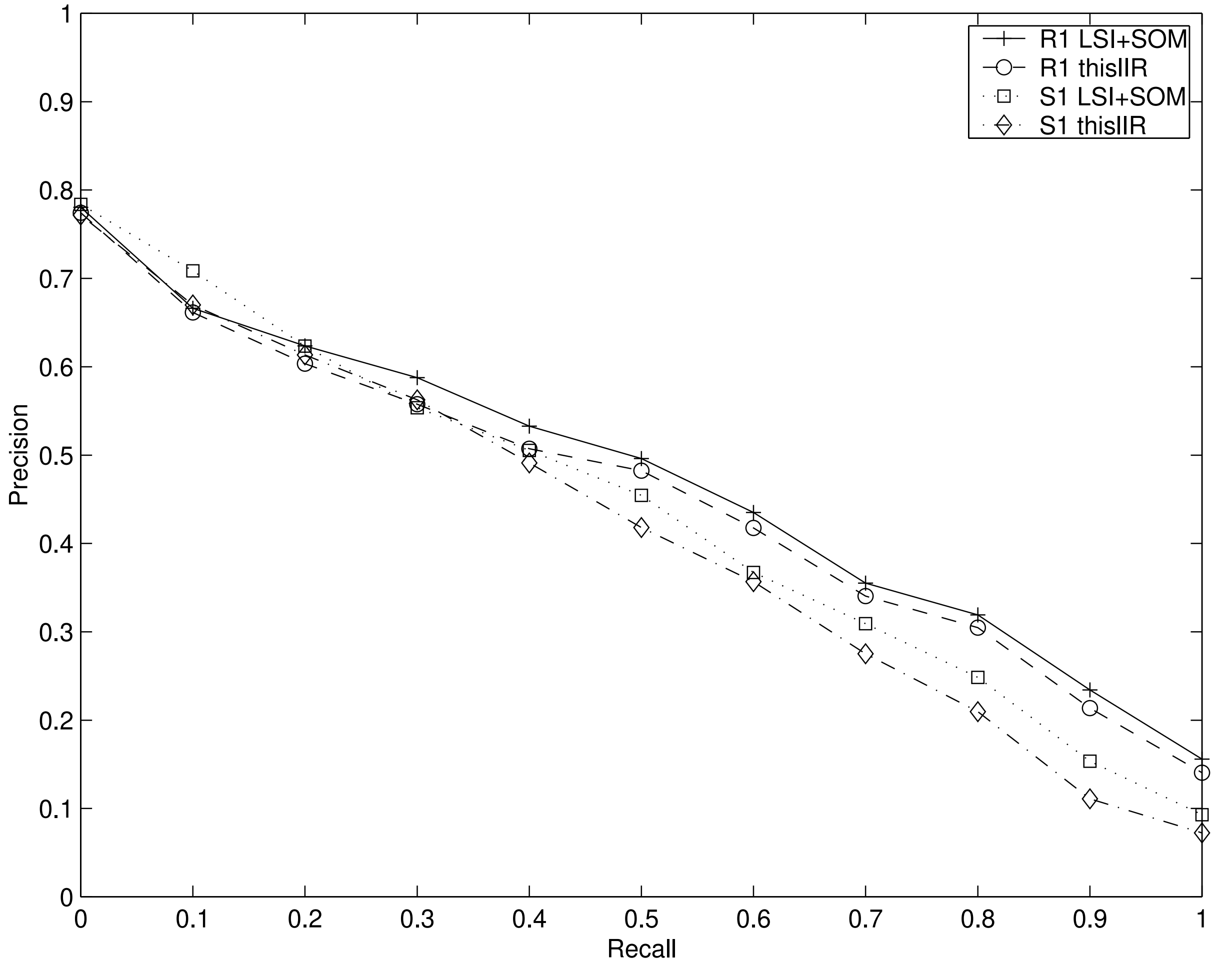

Figure 3: The recall-precision curves for the proposed LSI+SOM and the baseline *thislIR* using reference transcriptions (R1) and THISL decoding (S1).

bigger test shows no changes. The next test was to lower the significance threshold S for the new index terms to be accepted according to the LSA score. This does not seem to have any other effect except that the index files grow very large as more and more terms are taken into the index.

The Okapi parameters (K and b) can be tuned for the optimal performance in each task. For the LSI+SOM experiments we used the default values ($K = 2$ and $b = 0.7$). In Table 2 we tested how much we can improve by tuning them for the best performance. For these two tasks it seems, however, that the default values are quite good, because the tuning does not give large improvements. In the test called "post-Okapi", the Okapi term weighting was applied to the result of equation 8 rather than just to the term frequency CW.

Table 3 gives results for variations of the document vector smoothing. K_d is the number of closest reference vectors (best-matching SOM kernels) used for smoothing. SOM_d is the size of the document SOM. These parameters change very little the measured AP. If we discard the SOM and just use the K_d closest other document vectors of the collection (KNN), the results get worse, however.

Table 4 tests the smoothing of the word vectors. The idea is the same as for document vectors: The K_w closest reference vectors (best-matching SOM kernels) are selected in the semantic space and their sum, weighted by the distance, is used as the new smoothed semantic word vector (see Section 4.1). SOM_w is the size of the word SOM. This smoothing does not seem to affect much the APs.

Index variations:	TREC-7/S1	TREC-8/B1
$\lambda = 0.1$, $S = 99.9\%$	38.1	42.4
$\lambda = 0.05$	38.3	42.3
$\lambda = 0.2$	37.9	42.3
$S = 99\%$	38.1	42.4
$S = 95\%$	38.1	-
post-Okapi	37.4	40.9
tuned Okapi parameters	38.9	42.8

Table 2: Testing different ways to weight and combine the LSA score and the traditional (Okapi) frequency weight (see section 6). The default parameters (used in the baseline LSI+SOM system) are given on the first row. The results are the average precisions (AP %) for the test queries.

Index variations:	TREC-7/S1	TREC-8/B1
$K_d = 10$, $SOM_d = 600$	38.1	42.4
$K_d = 3$	38.1	42.4
$K_d = 20$	38.1	42.4
$SOM_d = 1200$	38.2	42.4
$SOM_d = 2000$	38.2	42.4
KNN (instead of SOM)	37.2	41.0

Table 3: Average precisions (AP %) of results for the variations of the smoothing of the document vectors (see section 6). The default parameters (used in the baseline LSI+SOM system) are given on the first row.

The Table 5 is probably the most interesting of the parameter robustness tests. Here, we varied the construction and dimensionality of the original word and document vectors before the SOMs are trained and used for smoothing the vectors. First the entropy based word weighting [Bellegarda, 1999] W_i^{ent} was substituted by a simple inverse document frequency weight:

$$W_i^{idf} = 1 - \frac{\log f_i^d}{\log m} , \tag{9}$$

where the document frequency f_i^d is the number of documents where the word w_i was observed and m is the total number of documents. The entropy weight, respectively, is computed from the normalized entropy of the word in the document collection, where word frequency f_{ij}^w is the frequency of word w_i

Index variations:	TREC-7/S1	TREC-8/B1
No WordSOM	38.1	42.4
$SOM_w = 1200$, $K_w = 10$	38.3	42.4
$SOM_w = 1200$, $K_w = 3$	38.2	42.4
$SOM_w = 1200$, $K_w = 20$	38.2	42.4
$SOM_w = 2000$, $K_w = 10$	38.1	42.4

Table 4: Average precisions (AP %) of results for the variations of word SOM used for smoothing the semantic word vectors. The default (used for LSI+SOM in Table 1) did not use any word vector smoothing.

in the document j:

$$W_i^{ent} = 1 + \frac{\sum (f_{ij}^w / \sum f_{ij}^w) \log (f_{ij}^w / \sum f_{ij}^w)}{\log m} \,. \tag{10}$$

The entropy weighting is theoretically more appealing (the mutual information between the document and the word [Siegler and Witbrock, 1999]), but here, using the simpler approximation by W_i^{idf} does not change much the AP. Interesting is that also the RM and SVD dimensions can be quite small without much effect in the results. Even if the SVD is completely skipped so that the SOMs are trained directly with RM vectors, we do not loose much in AP.

Word vector variations:	TREC-7/S1	TREC-8/B1
$RM = 200$, $SVD = 200$, W^{ent}	38.1	42.4
W^{idf}	38.1	42.1
$RM = 300$, $SVD = 200$	38.0	42.3
$RM = 200$, $SVD = 50$	38.1	42.4
$RM = 200$, no SVD	38.1	42.3
$RM = 100$, no SVD	38.2	42.4

Table 5: Average precisions (AP %) of results for different word and document vector dimensions and weighting. The default values (used in the baseline LSI+SOM system) are given on the first row. Word weights W^{idf} are based on the inverse document frequency and the default W^{ent} on the entropy.

6 Discussions

From the IR point of view, it is clear that the two evaluation sets used in this paper are not very large as there are only 3000 and 22000 documents, and 23 and 50 judged test queries, respectively. However, even for a near realtime ASR this amount of 100 and 550 speech hours is rather remarkable task, because each decoding run can take several months computation time. TREC does not provide any analysis of the statistical significance of the results. We tried to analyze the statistical significance using the Matched Pairs test [Gillick and Cox, 1989]. This test assumes that the result of each individual test query, for example AP, is independent and compares then whether there is any difference between the performance of two algorithms in these tests. For APs in Table 1, for example, this Matched Pairs scores *thislIR* and LSI+SOM to be significantly different at 95 % significance level for TREC-8/B1, but not for TREC-7/S1.

In the actual TREC-7 evaluation [Garofolo et al., 1999], the overall best APs were: $S1 = 51\%$, $B1 = 51\%$, $R1 = 57\%$; and in TREC-8: $S1 = 55\%$, $B1 = 55\%$, $R1 = 56\%$. The best systems in these TREC evaluations exploited all external text databases either to expand queries or documents to get better index terms than what would be possible just by decoding the given audio. These expansions have not yet been tried with the current LSI method. However, for the baseline *thislIR* (see Table 1), there is a QE version that achieved $S1 = 45\%$, $B1 = 42\%$, $R1 = 49\%$ in TREC-7 and $S1 = 53\%$, $B1 = 53\%$, $R1 = 56\%$ in TREC-8 being among the very best systems. It is expected that the queries expanded for the traditional indexes could be helpful for the current LSI as well. Another convenient way to exploit the external text data with the SOM based LSI would be just to train the SOM with a large (ASR-) error-free material and expand the speech documents to the semantically closest text documents or document clusters.

Table 5 suggests that the average performance is very robust for most of the parameters. It is interesting to note that using the computationally more expensive KNN smoothing instead of SOM actually degrades the results, but leaving the SVD out, which makes the indexing even lighter, does not cause significant changes. This seems to suggest that the SOM smoothing is here a very essential part of the LSI.

In Table 1 we see that the IR results using the decoded speech are not very far from those of the (human) reference transcripts. This indicates that the state-of-art ASR is quite sufficient. However, it should be noted that the reference transcripts are not completely error free either, and that this result is only valid for broadcast *news*. Preliminary experiments in other broadcast material with more free speech and more difficult conditions have shown severe difficulties.

In addition to the decoded text output, the ASR can also provide more help for the indexing. The likelihood or confidence scores of the decoding hypotheses could be used to weight the index terms so that more uncertain terms would have lower weight in ranking. Properly weighted N-best hypothesis and whole word lattices could be used as well to prevent important words to be missed by the ASR. One important point is, however, that the most important words for indexing are often the rare ones which are sometimes difficult to recognize and might, thus, get low scores.

7 Conclusions

A novel method for latent semantic indexing (LSI) is described and tested for spoken audio. The motivation for developing this method was to gain robustness for recognition errors and word noise in short documents as well as to improve the speed and visualization of the LSI. This method includes random mapping (RM) for rapid and controlled dimensionality reduction, entropy based word weighting, probabilistic index weights by combined Okapi term weighting and semantic matching, and using self-organizing maps (SOMs) to smooth the document and word vectors. In addition to computing the index, the clustering of the documents into latent topic models by SOM provides an interesting way to visualize the results. The IR performance of the system has so far been tested quantitatively for two standard broadcast news IR evaluation databases and the results are highly encouraging.

Acknowledgments

This work was supported by ESPRIT Long Term Research Project THISL. I wish to thank the THISL partners, especially Dave Abberley from Sheffield University, for helping me to get the evaluation data and the speech decodings.

References

[Abberley et al., 1999a] Abberley, D., Kirby, D., Renals, S., and Robinson, T. (1999a). The THISL broadcast news retrieval system. In *ESCA ETRW workshop on Accessing Information in Spoken Audio*, Cambridge, UK.

[Abberley et al., 1999b] Abberley, D., Renals, S., Robinson, T., and Ellis, D. (1999b). The THISL SDR system at TREC-8. In *Proceedings of the 8th Text Retrieval Conference (TREC-8)*. (to be published).

[Allan et al., 1998] Allan, J., Callan, J., Croft, W., Ballesteros, L., Byrd, D., Swan, R., and Xu, J. (1998). INQUERY does battle with TREC-6. In *Proceedings of the Sixth Text Retrieval Conference (TREC-6)*, pages 169–206.

[Andersen, 1998] Andersen, J. (1998). Baseline system for hybrid speech recognition on french. COM 98-7, IDIAP.

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

[Bellegarda, 1999] Bellegarda, J. R. (1999). Speech recognition experiments using multi-span statistical language models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 717–720.

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

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

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

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

[Garofolo et al., 1999] Garofolo, J. S., Voorhees, E. M., Auzanne, C. G. P., and Stanford, V. M. (1999). Spoken document retrieval: 1998 evaluation and investigation of new metrics. In *ESCA ETRW workshop on Accessing Information in Spoken Audio*.

[Gillick and Cox, 1989] Gillick, L. and Cox, S. (1989). Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 532–535, Glasgow, Scotland.

[Golub and Reinsch, 1971] Golub, G. and Reinsch, C. (1971). *Handbook for Matrix Computation II, Linear Algebra*. Springer-Verlag, New York.

[Hofmann, 1998] Hofmann, T. (1998). Probabilistic latent semantic analysis. TR 98-042, International Computer Science Institute.

[Hofmann, 1999] Hofmann, T. (1999). Probabilistic topic maps: Navigating through large text collections. In *Proceedings of the Third Symposium on Intelligent Data Analysis (IDA'99)*, Amsterdam, Netherlands.

[Honkela et al., 1996] Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. (1996). Newsgroup exploration with WEBSOM method and browsing interface. Report A32, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland.

[Johnson et al., 1999] Johnson, S., Jourlin, P., Moore, G., Jones, K. S., and Woodland, P. (1999). The Cambridge university spoken document retrieval system. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 49–52.

[Johnson and Lindenstrauss, 1984] Johnson, W. and Lindenstrauss, J. (1984). Extensions of Lipshitz mapping into Hilbert space. *Contemp. Math.*, 26:189–206.

[Kaski, 1998] Kaski, S. (1998). Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, volume I, pages 413–418.

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

[Kohonen et al., 1999] Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., and Saarela, A. (1999). Self organization of a massive text document collection. In Oja, E. and Kaski, S., editors, *Kohonen Maps*, pages 171–182. Elsevier.

[Kurimo, 1999] Kurimo, M. (1999). Indexing audio documents by using latent semantic analysis and som. In Oja, E. and Kaski, S., editors, *Kohonen Maps*, pages 363–374. Elsevier.

[Kurimo and Mokbel, 1999] Kurimo, M. and Mokbel, C. (1999). Latent semantic indexing by self-organizing map. In *ESCA ETRW workshop on Accessing Information in Spoken Audio*, pages 25–30, Cambridge, UK.

[Lagus and Kaski, 1999] Lagus, K. and Kaski, S. (1999). Keyword selection method for characterizing text document maps. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'99)*, volume 1, pages 371–376. London: IEE.

[Ng and Zue, 1998] Ng, K. and Zue, V. W. (1998). Phonetic recognition for spoken document retrieval. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 325–328.

[Papadimitriou et al., 1998] Papadimitriou, C., Raghavan, P., Tamaki, H., and Vempala, S. (1998). Latent semantic indexing: A probabilistic analysi. In *Proc. 17th ACM Symposium on the Principles of Database Systems*, Seattle, USA. Invited for publication in Journal of Comp. and System Sciences.

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

[Rauber and Merkl, 1999] Rauber, A. and Merkl, D. (1999). Automatic labeling of self-organizing maps: Making a treasure-map reveal its secrets. In *Proceedings of the 3. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'99)*, Bejing, China.

[Renals et al., 1998] Renals, S., Abberley, D., Cook, G., and Robinson, T. (1998). THISL spoken document retrieval. In *Proceedings of the Seventh Text Retrieval Conference (TREC-7)*.

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

[Robinson et al., 1999] Robinson, T., Abberley, D., Kirby, D., and Renals, S. (1999). Recognition, indexing and retrieval of british broadcast news with the THISL system. In *Proceedings of 6th European Conference on Speech Communication and Technology*, Budabest, Hungary.

[Robinson and Christie, 1998] Robinson, T. and Christie, J. (1998). Time-first search for large vocabulary speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 829–832.

[Robinson et al., 1996] Robinson, T., Hochberg, M., and Renals, S. (1996). The use of recurrent networks in continuous speech recognition. In Lee, C. H., Paliwal, K. K., and Soong, F. K., editors, *Automatic Speech and Speaker Recognition - Advanced Topics*, chapter 10, pages 233–258. Kluwer Academic Publishers.

[Salton, 1971] Salton, G. (1971). *The SMART Retrieval System-Experiments in Automatic Document Processing*. Prentice-Hall, NJ.

[Siegler and Witbrock, 1999] Siegler, M. and Witbrock, M. (1999). Improving the suitability of imperfect transcriptions for information retrieval from spoken documents. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 505–508.

[Simula et al., 1999] Simula, O., Ahola, J., Alhoniemi, E., Himberg, J., and Vesanto, J. (1999). Self-organizing map in analysis of large-scale industrial systems. In Oja, E. and Kaski, S., editors, *Kohonen Maps*, pages 375–387. Elsevier.

[Ultsch, 1999] Ultsch, A. (1999). Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In Oja, E. and Kaski, S., editors, *Kohonen Maps*, pages 33–45. Elsevier.

[Xu and Croft, 1996] Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proc. ACM SIGIR*.