ELSEVIER

Speech Communication 38 (2002) 29–45

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Thematic indexing of spoken documents by using self-organizing maps

Mikko Kurimo *,[1]

Neural Networks Research Centre, Helsinki University of Technology, P.O. Box 5400, Konemiehentie 2, 02150 Espoo, Finland

Received 18 February 2000; accepted 22 May 2001

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 in a vector space as a document vector which is the normalized sum of the word vectors in the story. A large collection of such document vectors is used to train a self-organizing map (SOM) to find latent semantic structures in the collection. As the stories in spoken news are short and will include speech recognition errors, smoothing of the document vectors using the semantic clusters determined by the SOM is introduced to enhance the indexing. The application in this paper is the indexing and retrieval of broadcast news on radio and television. Test results are given using the evaluation data from the text retrieval conference (TREC) spoken document retrieval (SDR) task.

Zusammenfassung

Eine Methode wird dargestellt, um einen nützlichen suchbaren Index für gesprochene Audiodokumente zur Verfügung zu stellen. Die Aufgabe unterscheidet sich von der traditionellen (Text-) Dokumentenindexierung, weil grosse Audiodatenbanken durch automatische Spracherkennung dekodiert werden und dabei häufig Dekodierungsfehler auftreten. Die Idee in diesem Artikel ist es, die Grösse der Datenbank auszunutzen und die besten Index-Terms für jedes Dokument mit Hilfe ähnlicher, im semantischen Vektorraum naheliegender Dokumente auszuwählen. Mit einem Spracherkenner wird zuerst der Audiostrom in einen Textstrom umgewandelt. Dann wird der Text jeder Nachricht durch einen Dokumentenvektor dargestellt, der die normalisierte Summe der Wortvektoren dieser Nachricht ist. Eine grosse Anzahl von Dokumentenvektoren wird für das Training einer selbstorganisierenden Karte verwendet, um die Cluster und die latenten semantischen Strukturen in dieser Collection zu finden. Weil die gesprochenen Nachrichten ziemlich kurz sind und Spracherkennungsfehler aufweisen, werden die Dokumentenvektoren durch die thematischen Cluster geglättent. Diese werden mit der selbstorganisierenden Karte aufgefunden und ermöglichen es,

* Tel.: +358-9-4515388; fax: +358-9-4513277.
E-mail address: mikko.kurimo@hut.fi (M. Kurimo).
URL: http://www.cis.hut.fi/mikkok
[1] Until 31 May 2000, the author worked at IDIAP in Martigny, Switzerland.

0167-6393/02/$ - see front matter
PII: S0167-6393(01)00042-5

einen besseren Index zu erhalten. Die in diesem Artikel vorgestellte Anwendung ist Indexing und Retrieval von Radio- und Fernsehnachrichtensendungen. Testergebnisse werden auf "TREC – spoken document retrieval" gegeben.

Résumé

Dans ce papier, une méthode est présentée pour déterminer un index utile à la recherche dans des documents audio. La tâche diffère de l'indexation traditionelle de documents textuels, parce que les grandes bases de données sonores sont décodées par la reconnaissance automatique de la parole, et des erreurs de décodage s'y produisent fréquemment. L'idée centrale dans cet article est de profiter de la taille de la base de données pour choisir les meilleures termes d'indexation pour chaque document et ce en considérant les autres documents qui lui sont proches dans un espace vectoriel sémantique. Pour ce faire, le signal acoustique est d'abord converti en texte par un système de reconnaissance de la parole. Ensuite, le texte de chaque document est représenté par un vecteur qui est la somme normalizée des vecteurs des mots du document. Une grande collection de vecteurs de document est employée pour former une carte de Kohonen qui permet une classification des documents et une découverte des structures sémantiques dans la collection. Comme les documents des nouvelles lues sont courts et incluent des erreurs de reconnaissance de la parole, l'idée de lisser les vecteurs de document en utilisant les classes thématiques déterminées par la carte d'auto-organisation de Kohonen est introduite pour obtenir une meilleure indexation. Dans cet article, l'approche précédente est appliquée à l'indexation et à la recherche dans les documents de nouvelles télévisées et de radio. Les résultats expérimentaux sont donnés en utilisant les données d'évaluation de TREC pour la tâche de recherche dans les documents sonores.

Keywords: Latent semantic indexing; Self-organizing maps; Spoken document retrieval; Broadcast news indexing

1. Introduction

As larger and larger audio databases become accessible, the problem of indexing for automatic information retrieval becomes extremely important. Often no written transcripts of the audio are available, and thus the indexing has to be based solely on the automatically decoded text.

One example of an important and large audio source is the broadcast news. There is a continuous flow of new multimedia data coming through many parallel channels in television and radio in the form of video, text and pure audio. The automatic and fast management of breaking news is crucial in many business areas, not to mention the broadcasting companies themselves. The management of large archives of past news items is a closely related and important task.

The focus application of this paper is the indexing and retrieval of broadcast news from radio and TV. However, the presented methods are also applicable for indexing other spoken audio sources. Many methods are similar to those 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 perform this extraction. One can recognize and index the audio at the phonetic level (Ng and Zue, 1998), spot 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 of a method should be made by considering the nature of the indexing task at hand (Abberley et al., 1999a). In the thematic indexing of spoken language (THISL) project (Abberley et al., 1999a), which is the framework of this paper, 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 within index terms reduces the effect of recognition errors. Naturally, not all words are equally effective for indexing and this can be taken into account by weighting the index terms (Robertson and Sparck-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), the latent semantic analysis (LSA) (Deerwester et al., 1990) of the document collection, and the smoothing and visualization by the self-organizing map (SOM) (Kohonen, 1997). 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 [2] to the document. In addition further index terms can be identified based on the semantic similarity. LSA is motivated by the dimensionality reduction and the noise suppression obtained by projecting the documents and index terms into a lower dimensional semantic vector space. Noise reduction is also the aim for smoothing the vectors by the probabilistic clustering in the semantic space. In the clustering of documents by SOM, the idea is to create a topology preserving mapping which identifies 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 section of this paper explains briefly the baseline THISL broadcast news indexing system and brings out the most relevant points for the thematic indexing method. The third section presents the new method and explains how the LSA, random mapping (RM) and SOM are applied. One section is devoted to the experiments and presentation of the evaluation metrics used to analyze the results. The final sections of the paper include brief discussions and final conclusions.

2. Baseline system

This section provides a brief overview of the relevant features of the baseline speech retrieval system (Renals et al., 1998) on which the current latent semantic indexing (LSI) system is built. The main components of the system are the hidden Markov model/artificial neural network (HMM/ANN) hybrid LVCSR along with the indexing and IR. The new system is strongly related to the baseline system, as it utilizes the same speech decoder and the implementation of the IR is a combination of the baseline index and the latent semantic index.

The basis of the whole spoken document retrieval (SDR) system is the speech recognition. The documents created for indexing and retrieval are formed solely from the text output of 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 or sequences of constant duration in seconds (Robinson et al., 1999).

For the text-to-speech transformation different speech recognizers have been created to cover languages used in the framework of the THISL project: British English (Robinson et al., 1999), American English (Renals et al., 1998; Abberley et al., 1999b), and French (e.g., 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

[2] 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.

was expected. One of the reasons for the high WER was 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 also some improvements in the IR precision for lower WER ASR outputs and even improved precision 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 perceptual linear predictive analysis (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 64 K word pronunciation dictionary and large trigram LMs. The decoding (referred by S1 in Section 4) performed by the THISL project partners, took approximately 3 $\times$ realtime on standard hardware.

Even having the best and highly sophisticated speech recognition system does not guarantee success in a speech retrieval evaluation. In fact, the TREC SDR results from 1998 (Garofolo et al., 1999) and 1999 show that even systems with rather simple automatic speech recognition (ASR) can perform well with a successful information retrieval (IR) implementation. Some IR systems are even fairly robust for different ASR outputs ranging between 20% and 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 Okapi term weighting function is used to compute the relevance of each remaining index term as in (Abberley et al., 1999b). Specifically, the term weight $\mathrm{CW}(t,d)$ (''the collection weight'') (Robertson and Sparck-Jones, 1976) for term t in document d is computed as

$$\mathrm{CW}(t,d) = \frac{\mathrm{CFW}(t) * \mathrm{TF}(t,d) * (K+1)}{K(1-b) + b(\mathrm{NDL}(d)) + \mathrm{TF}(t,d)}, \tag{1}$$

where $\mathrm{TF}(t,d)$ is the frequency of the term in the current document and $\mathrm{CFW}(t)$ the frequency in the whole collection. $\mathrm{NDL}(d)$ is the normalized document length while K and b are user specified constants as in (Abberley et al., 1999b).

An additional 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 modifies the original queries in order to add associated terms retrieved from some external text databases. An LSI based system should profit similarly from the expanded queries, because it is unlikely that LSI would be able to derive the same associations using only the, usually much smaller, spoken database. However, QE was not included in the experiments presented in this paper, mainly because after QE the differences in indexing methods are not so easily seen by the IR comparisons.

3. The new method

3.1. Integrating LSA into the system

The motivation for attempting to take the semantics into account in the indexing of spoken documents is to optimally deal with the noise observed in the decoded texts. The noise comes both from the use of synonyms and homonyms and from the randomness of the choice of words. Also in decoded documents the decoding errors increase the word noise.

One way to automatically create a semantic [3] representation of a document is to consider the distribution of words in it (Salton, 1971). Of course,

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

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 to each other and the order in which the words appear. It also becomes meaningful only when the documents and the collections are large enough to give some statistical significance to the word count distributions. For some purposes, however, such as the indexing of documents based on their contents, this approximative semantic representation can still be very useful despite its coarse nature.

The word count vector representation of a document is difficult to handle, because the vectors are as long as the size of the whole vocabulary and the word counts themselves are noisy, especially for short documents. As word count vectors are typically very sparse, it is possible to project them into a lower dimensional subspace to reduce dimensionality and the inherent noise. The use of singular value decomposition (SVD) to find such a subspace is known as LSA (Deerwester et al., 1990). According to the L_2 norm any first k singular vectors of SVD represent the original vector set by the maximal accuracy in k-dimensional space (Eckart and Young, see (Golub and Reinsch, 1971)). Thus, the word and document coding (Eq. (2)) by SVD is based on the most important correlations in the whole document collection, and the other correlations are 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. Methods other than SVD that can extract latent semantics are, for example, the probabilistic LSA (Hofmann, 1999) and the method based on SOMs described in Section 3.3.

In this work, we employed SVD to code terms $w_i,\ i = 1,\ldots,n$ in the k-dimensional subspace of the first k eigenvectors as

$$x_i^{\mathrm{T}} = u_i S_k / \|u_i S_k\|, \tag{2}$$

by using the normalized row i of matrix $U_k S_k$ from the decomposition $A_k = U_k S_k V_k^{\mathrm{T}}$. To measure the closeness of the words (for smoothing or clustering, e.g.) we use the simple semantic similarity measure (Bellegarda, 1997)

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

In this work the document vectors $y_j,\ j = 1,\ldots,m$, were made by summing the semantic word vectors for each document. The sum of the semantic word vectors weighted by the word frequencies points to the average semantics of the chosen set of words. Thus, the effect of the decoding errors will be reduced, as 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. Another way to form the document vector is to set $y_j = S_k v_j^{\mathrm{T}} / \|S_k v_j^{\mathrm{T}}\|$ by employing directly the normalized column j of matrix $S_k V_k^{\mathrm{T}}$.

After normalizing the lengths of the semantic document vectors obtained by the SVD transformation, we can measure the semantic similarity of two documents by a simple dot-product $y_i^{\mathrm{T}} y_j$. In latent semantic indexing (LSI) the same measure is used to check how close an index term is to a document in the semantic space and, thus, to determine the relevance simply by $x_i^{\mathrm{T}} y_j$. In the original high-dimensional space this comparison of directions would be equal to the basic bag-of-words index without word noise, as 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 10 word occurrences against nine and one against zero. Furthermore, these comparisons do not apply similarly for rare and common words. However, despite the criticism, in some applications the SVD based LSI has been reported to perform well and the different word weighting functions can be combined with LSA to improve the performance. For document collections that fulfill certain assumptions of "good behavior", it can 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.

One problem we observed with LSI is how to judge the relevance between the generally frequent terms that were extracted from the decoded document and the rare words that were not observed in that decoding. The frequent term will gain from its observed frequency and the rare one from its general closeness to the topic. In this work it was decided to try a combined weight by taking into account both the LSI weight called $\mathrm{SW}(t,d)$ (a smoothed version of distance (3), see Section 3.3) and the Okapi weight $\mathrm{CW}(t,d)$ (Eq. (1)). This was implemented by re-scaling both weights between (0, 1) (to obtain CW' and SW') and taking a linear combination

$$W_{td} = (1-\lambda)\mathrm{CW}'(t,d) + \lambda \mathrm{SW}'(t,d), \tag{4}$$

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, $\lambda = 0$ would be equal to the baseline *thislIR* scoring (Renals et al., 1998) trusting completely in the decoded terms and $\lambda = 1$ would give complete control to the semantics extracted from the given document collection. It is noteworthy, however, that despite re-scaling the different dynamic ranges of these combined measures may cause problems. This has been noted, for example, in attempts to integrate LSA and conventional frequency based measures for statistical language modeling (Coccaro and Jurafsky, 1998).

3.2. Using RM in the system

The RM has recently been successfully used both to speed up the SVD in LSI (Papadimitriou et al., 1998) and to transform large vocabularies and document collections into a vector space suitable for SOM analysis (Kaski, 1998). The foundation of RM is that if points are mapped into a random subspace of a suitably high dimension, then the distances between the points are approximately preserved (Johnson and Lindenstrauss, 1984). In this paper, RM is used to provide a fast SVD and an SOM of semantic document indexes starting from the word count representation of the documents.

A successful dimensionality reduction in high-dimensional sparse vectors does not need very complicated methods. For example, for the document classification application (Kaski, 1998) it was experimentally shown that using a RM of dimensionality $l \geqslant 90$ provides comparable document classification results as those obtained using an SVD of rank $k = 50$. In applications where the computational complexity severely restricts the use of SVD, a simpler method of dimensionality reduction may be the most convenient way to avoid introducing other unwanted approximations. It is possible to optimize the sparse SVD directly by using special iterative SVD algorithms like the Single Vector Lancsos (Berry, 1992). However, it is still sensitive to high dimensions and an acceptable solution in a feasible time is not always guaranteed. The traditional way to handle very large collections is sampling. Either documents or terms or both documents and terms can be sampled. The disadvantages of sampling are that the sampled documents or terms should be carefully selected to maintain good accuracy and it is therefore hard to predict the IR results concerning the lost items.

RM is made by assigning an l-dimensional random vector, $r_i \in \mathscr{R}^l$, of unit length to each term $w_i,\ i = 1,\ldots,n$ (n is the vocabulary size, $l \ll n$). In this paper, we approximated the SVD based LSA by computing an SVD for the random mapped term-document matrix. In this matrix, a document is not represented by an n-dimensional word count vector as usually, but by an l-dimensional weighted average vector of the random vectors assigned to the terms used in the document. Instead of weighting the word vectors just by the word counts, we applied two slightly more sophisticated measures for the term relevance. The first measure was the inverse document frequency weight (Eq. (8)) and the second one the entropy (mutual information) weight (Eq. (9)) (Bellegarda, 1999). The SVD of this approximated document matrix was employed to code the term w_i by the first k eigenvectors similarly as in Eq. (2) except that u_i was replaced by $r_i\tilde{U}_k$ from the obtained decomposition $\tilde{A}_k = \tilde{U}_k\tilde{S}_k\tilde{V}_k^{\mathrm{T}}$, so that

$$x_i = r_i\tilde{U}_k\tilde{S}_k / \|r_i\tilde{U}_k\tilde{S}_k\|. \tag{5}$$

The rows $\tilde{u}_i$ of matrix $\tilde{U}_k$ just give the semantic subspace coding for each random dimension, so the new code vectors are obtained by projecting the original random vectors to the semantic subspace. The document vectors are weighted sums of the new word vectors and the smoothing can then be applied as in normal SVD (see Section 3.3.1 either for the word vectors, for the document vectors, or both for the word and the document vectors).

The LSI approximation obtained by combining RM and SVD is good, because all the terms and documents are included and the SVD computation can be conventional as the dimensions are low. The point is that we are using an *approximated term-document matrix*, and 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. An analysis result (Papadimitriou et al., 1998) states that $\tilde{A}_{2k}$ recovers from the original term-document matrix A almost as much [4] as A_k, so that with high probability

$$\|A - \tilde{A}_{2k}\|_F^2 \leqslant \|A - A_k\|_F^2 + 2\epsilon\|A\|_F^2 \tag{6}$$

for l large enough, i.e., $l = O(\log n/\epsilon)$. The speed-up will then be from $O(mnc)$, which is for the direct sparse SVD of m documents and c non-zero elements per row, to $O(mc \log n + m \log^2 n)$. The complexity of RM is $(O(mcl))$ and the non-sparse SVD $(O(ml^2))$. In practice, the dimension of the random vectors $l \ll n$ can well be just a couple of hundreds or even lower, as successfully applied in the "semantic" SOMs (Ritter and Kohonen, 1989) and SOMs of massive document collections (Kohonen et al., 1999).

3.3. Using SOM in the system

The motivation behind 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. It is also capable to extract some semantic information from very large document collections (Kohonen, 1997; Kohonen et al., 1999) by a topology-preserving mapping into a low-dimensional space. The second motivation comes from the need to *visualize* the relations between the semantic topics in the document collection. For example, in IR it is useful to see which topics are present in the database in general, what are the topics corresponding to the best retrieved documents, and which additional topics are semantically close to them.

3.3.1. 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 word error rate (WER) document decodings. LSA also suffers from word noise, if it is performed using a database of noisy data and if the database is insufficiently large to provide 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 between the K nearest neighbors (KNN) for each document. However, this is too slow for large document collections, if no major optimizations are made to reduce its complexity ($O(m^2k)$ 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 obtain a more continuous mapping, 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 learn to represent some often occurring document types of the collection, a single document can 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 also considerably faster than having to perform a KNN search of the whole input data. For an SOM of s

[4] The Frobenius norm $\|A\|_F^2$ is the sum of the squares of all the matrix elements.

units the complexity of the smoothing is only $O(mks)$ and the training of the SOM $O(ks^2)$. This can be further reduced by some efficient approximations (Kohonen et al., 1999).

SOM is not the only possible clustering method for smoothing. Methods such as K-means can provide a good clustering with less algorithmic complexity. However, SOM is rather convenient for smoothing large data sets. If the extraction of the main structures from the data is more important than quantization error or mapping of some individual data points, SOM is quite robust to its configuration, initialization and overlearning. This means that an exhaustive series of optimization experiments becomes unnecessary, as a suboptimal SOM may already be good enough.

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 the closest and most relevant index terms by a direct matching, 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. In this paper, the index is stochastic which means that, theoretically, all the terms are used to index every document, but each term-document association is weighted by the relevance. Thus, the smoothed LSI weight $\mathrm{SW}(t,d)$ is the weighted average of the projections to the K (best-matching) clusters $C_1,\ldots,C_K$,

$$\mathrm{SW}(t,d)=\sum_{i=1}^{K} g(t,C_i)g(C_i,d)\Bigg/\sum_{i=1}^{K} g(C_i,d), \quad (7)$$

where the weights are proportional to the projections from the document to those clusters. The projections use the similarity measure (Eq. (3)) for word and document vectors in the semantic space.

In practice, however, it was necessary to restrict the size of the created index file, so it was decided to store only the best index terms for each document. One suggestion for such a threshold was to compute the mean and variance of the index weights for each document and use the 99% significance level of a normal distribution. However, it is not guaranteed that the distribution of the distances (3) would be even close to a normal distribution and that all the stored index terms were semantically significant. This 99% threshold was set in preliminary experiments (Kurimo and Mokbel, 1999) to be high enough to avoid huge index files and low enough to still get many index terms for each document. This trade-off can be avoided by computing the index weights only for the presented query terms. However, in many IR applications a fast query processing time is important, so the generation of an index file with pre-computed weights is preferred.

For huge document collections (millions of documents), SOM has recently been successfully applied to organize the documents and reveal some semantic structures (Kohonen et al., 1999) without the time consuming semantic preprocessing that was previously used. This suggests that in indexing we could also try the SOM directly for document vectors formed from the RM word vectors, although the current evaluation data used for training is quite small. Experiments with this direct approximation of LSI indicated (Table 5) that leaving SVD out does not decrease significantly the final precision of LSI in this data.

As well as the semantic document vectors, the semantic word vectors can be smoothed by an SOM. This can be motivated by a more reliable representation for rare words which, generally, are more affected by word noise. Because 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.

3.3.2. 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 the user's mind", it is more efficient to try to provide some additional structural information about the documents, and to let the user determine *the relevant question* for the problem at hand. The visualization can give an overview of the existing topics, their hierarchies, and show the best index terms to describe them. Additional valuable information is to visualize where and how the obtained IR results are mapped in the collection.

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). As the 2D map plane tends to fold 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 the "mountain" between them (see Fig. 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.

From the LSI perspective, it is interesting to select some characteristic descriptors as labels to show the contents of the map clusters (see Fig. 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. Perform the stochastic indexing (Eq. (4));
2. Find the Voronoi regions (i.e., list the mapped documents) for all clusters;
3. In each cluster, sum the indexing weights for terms in the Voronoi region;
4. Show the top ranked index terms for each cluster as topic labels.

To view 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 Voronoi 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 Voronoi regions (i.e., lists where the documents would be mapped if the best match were ignored) with appropriate weights. However, it is unlikely that this would change significantly the order of top ranked descriptors.

In Fig. 1 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 stems already give some hints about the organization of the topics. For example, on the lower left corner, the labels are related to stock markets and a bit further 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 Palestinians, can be seen. Fig. 2 is a detailed version of Fig. 1. There we see

[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.cis.hut.fi/mikkok/scfig1.ps.gz and http://www.cis.hut.fi/mikkok/scfig2.ps.gz for better pictures with colors.

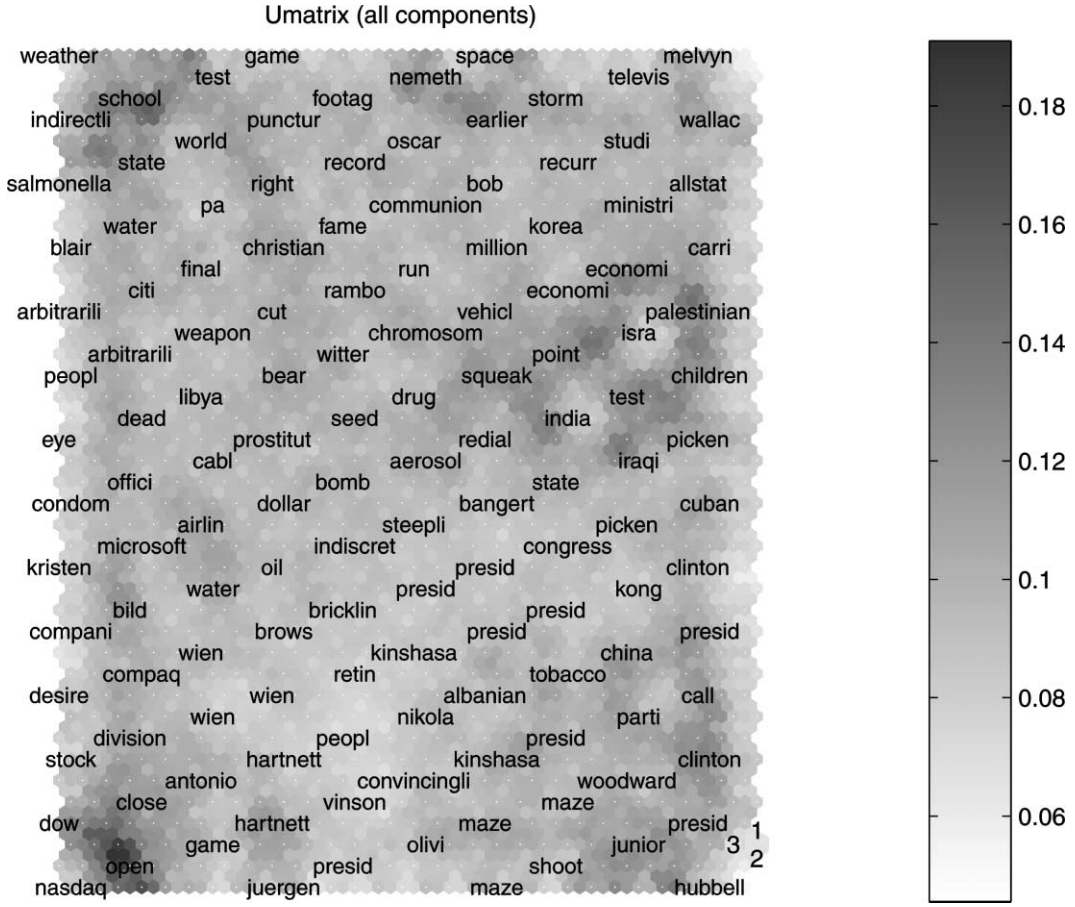

Fig. 1. An example of visualizing an indexed document collection by a labeled U-matrix. There are 1200 cells corresponding to the 1200 clusters (nodes) of the SOM grid. The semantic vectors of neighboring cells in this 2D map are, in general, near each other also in the original high-dimensional vector space, but due to the map folding, the distances are better shown in color. The lighter 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 Fig. 2 to zoom).

more clearly the clusters in the close neighborhood of the three best matches for the query that has been made.

The hit histogram, i.e., how many documents there are in each cluster, can be added to the same display, e.g., by using dots of variable sizes (Simula et al., 1999). Instead of the U-matrix the colors in the map can optionally show the distribution of a chosen SOM component plane, i.e. how relevant the topics are for just a certain semantic dimension. Other useful distributions which can be shown 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 large (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. A demonstration of the WEBSOM [7] shows an example of how to use several display hierarchies (Honkela et al., 1996; Kohonen, 1997). There a

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

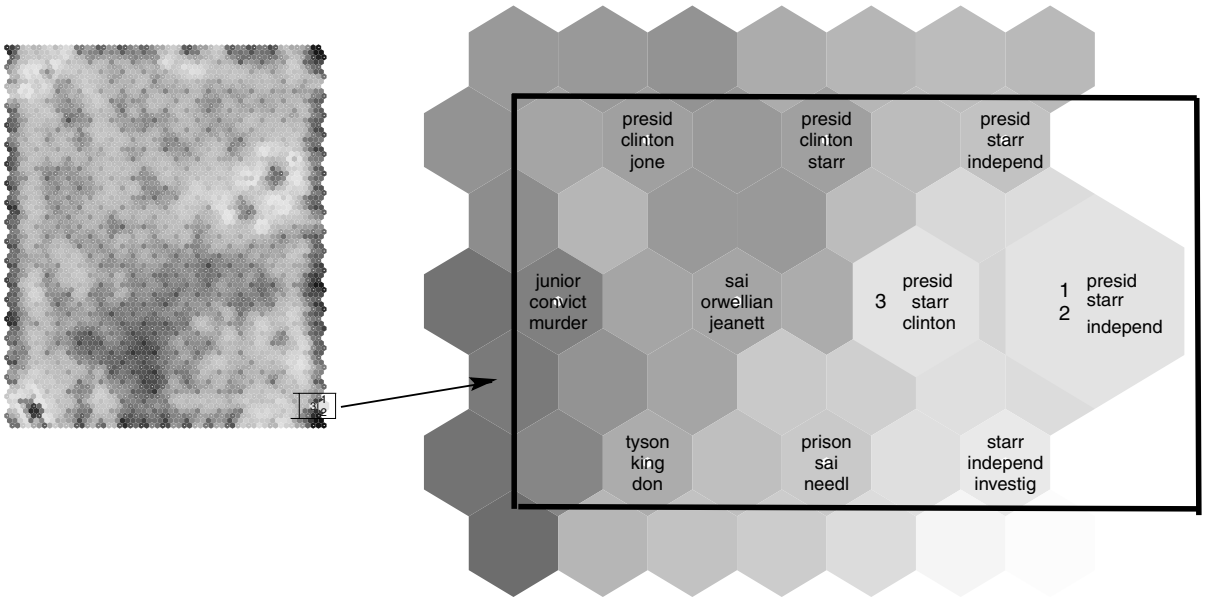

Fig. 2. Displaying the latent document topics in a 2D map of hexagons. The original query was "Lewensky" (sic.). The closest map cells for the three best documents are shown with magnified hexagons. The topic labeling used in Fig. 1 is here extended to the three best index terms.

user can select an interesting area from any level and zoom in or out to see the nearby topics and finally examine the selected cluster by viewing the associated documents. A similar interface could be useful to visualize and use an index of a large spoken document database.

4. Experiments

4.1. Evaluation measures for SDR

All the results reported here are based on the test queries of the 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.

The comparison of the spoken document indexes is not a straight-forward task. The WER of speech recognition varies considerably and it is unclear how much this affects 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 on different documents varies. 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). However, this involves a transformation of the LSI scores into probabilities, which is not straightforward 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 Fig. 3). An index is considered to be superior to another, if the precision of the retrieval results in each recall level is higher than that of the other method. The *recall* is the proportion of relevant documents which are retrieved and the *precision* is 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 the total number of

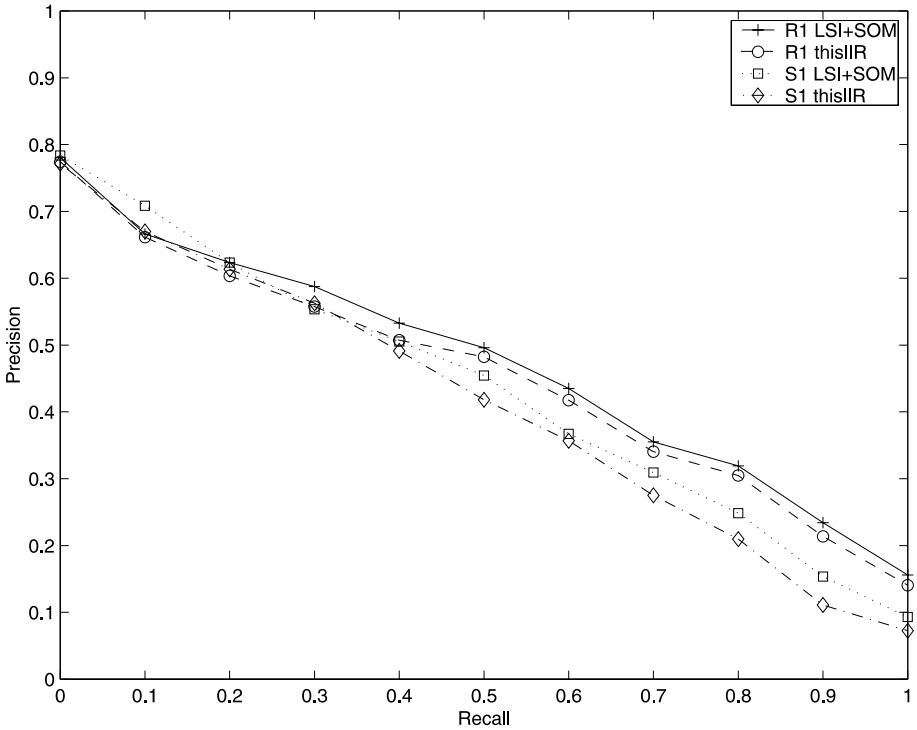

Fig. 3. The recall-precision curves for the proposed LSI + SOM and the baseline *thislIR* using reference transcriptions (R1) and THISL decoding (S1).

Table 1
Results for the indexing systems in different broadcast news sets and decodings (see Section 5)

		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

Precisions at the lowest level (0.10), at level R, and in average are given (P10%, RP% and AP%, respectively).

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 often the most relevant for practical IR applications where the users usually rather revise their queries than scan through all the given answers.

4.2. Results

Two broadcast news databases with fixed 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 h of news segmented into 3000 stories and the TREC-8 550 h in 22,000 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 performed 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* version 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 Eq. (4)). QE was disabled as explained in Section 2.

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 5.

Table 2
Testing different ways to weight and combine the LSA score and the traditional (Okapi) frequency weight (see Section 5)

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

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.

Table 3
Average precisions (AP%) of results for the variations of the smoothing of the document vectors (see Section 5)

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
*K*NN (instead of SOM)	37.2	41.0

The default parameters (used in the baseline LSI + SOM system) are given on the first row.

Table 4
Average precisions (AP%) of results for the variations of word SOM used for smoothing the semantic word vectors

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

The default (used for LSI + SOM in Table 1) did not use any word vector smoothing.

Table 5
Average precisions (AP%) of results for different word and document vector dimensions and weighting

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

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.

Table 2 concentrates on the combinations of the indexing weights given by the following two sources (see Eq. (4)): the frequency weights from the Okapi criterion CW (Eq. (1)) and the LSA scores SW (Eq. (7)). First we tested different λ values. The smaller test (TREC-7) gives superior APs for the smaller λ's, but the larger test shows no changes. The next test was to lower the significance threshold S (see, Section 3.3.1) to increase the amount of semantic index terms. 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 in Eq. (1)) 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 Eq. (4) 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 provide insignificant change to the measured AP. If we discard the SOM and just use the K_d closest other document vectors of the collection (KNN), the quality of the results deteriorates, however. Furthermore, even with small K (here we used $K = 3$), the KNN search for the whole document collection is very laborious.

Table 4 shows test results of the smoothing of 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 3.3.1). SOM_w is the size of the word SOM. This smoothing seems to have little effect on the APs.

Table 5 shows probably the most interesting tests of the parameter robustness. Here, we varied the construction and dimensionality of the original word and document vectors before the SOMs were 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,

$$W_i^{\text{idf}} = 1 - \frac{\log f_i^{\text{d}}}{\log m}, \tag{8}$$

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 in the document j,

$$W_i^{\text{ent}} = 1 + \frac{\sum (f_{ij}^{\text{w}} / \sum f_{ij}^{\text{w}}) \log (f_{ij}^{\text{w}} / \sum f_{ij}^{\text{w}})}{\log m}. \tag{9}$$

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 the AP much. Also of note is that 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.

5. 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 22,000 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. Thus, the computational load produced by the actual indexing is not so important, as the indexing is orders of magnitude faster than ASR (for these tests just one hour without SVD). If the amount of documents is increased the SVD becomes unfeasible quite easily, as can be seen from its computational complexity. Adaptive LSA methods that can be updated without computing a new SVD are especially important for applications where it is necessary to frequently add new document vectors or new words to the vocabulary.

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: $S1 = 55\%$, $B1 = 55\%$, $R1 = 56\%$. All the best systems in these evaluations exploited 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 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.

We tried the Matched Pairs test (Gillick and Cox, 1989) to analyze the statistical significance of the results. This test assumes that the result of each individual test query, for example AP, is independent and then compares whether there is any difference between the performance of the two algorithms in these tests. For APs in Table 1, for example, the Matched Pairs finds *thislIR* and LSI + SOM to be significantly different at 95% significance level for TREC-8/B1, but not for TREC-7/S1.

The main reason why the LSI did not give as significant improvements for the evaluated speech databases as expected, is probably the database size. SVD and SOM were performed using only rather small evaluation sets. It is assumed that significantly more training data is required to able to automatically learn statistically meaningful latent semantic representations from the decoded speech, where the stories are short and the obtained text contains many recognition errors. Thus, in the broadcast news indexing, the use of contemporary newswire texts to train the semantic models would be more convenient.

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 *K*NN 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 clustering is here an essential part of the LSI.

Table 2 shows the somewhat surprising result that the final accuracy changes little for different λ values. This can mean that the linear combination (Eq. (4)) is a suboptimal way to balance between the two different relevance weights SW and CW. Further experiments could be performed for adjusting the dynamic ranges of the weights or analyzing the confidence of LSA as in (Coccaro and Jurafsky, 1998).

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 for this application. 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 unconstrained speech and more difficult acoustic conditions have shown severe difficulties.

In addition to the decoded text output, the ASR can also provide further assistance for 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 as well could be used 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 may, thus, achieve low scores.

6. Conclusions

A novel method for LSI is described and tested for spoken audio. The motivation for developing this method was to gain robustness against recognition errors and word noise as well as to improve the speed and visualization of the LSI. This method includes RM for rapid and controlled dimensionality reduction, entropy based word weighting, stochastic index weights by combined Okapi term weighting and semantic matching, and using 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 slightly better than without the LSI.

Acknowledgements

This work was performed at IDIAP within the European Union ESPRIT Long Term Research

Project THISL with financial support from the Swiss Federal Office of Education and Science. I wish to thank the THISL partners, especially Dave Abberley from Sheffield University, for assisting me in obtaining the evaluation data and the speech decodings.

References

Abberley, D., Kirby, D., Renals, S., Robinson, T., 1999a. The THISL broadcast news retrieval system. In: ESCA ETRW Workshop on Accessing Inform. Spoken Audio, Cambridge, UK, pp. 14–19.

Abberley, D., Renals, S., Robinson, T., Ellis, D., 1999b. The THISL SDR system at TREC-8. In: Proc. 8th Text Retrieval Conf. (TREC-8).

Allan, J., Callan, J., Croft, W., Ballesteros, L., Byrd, D., Swan, R., Xu, J., 1998. INQUERY does battle with TREC-6. In: Proc. Sixth Text Retrieval Conf. (TREC-6), pp. 169–206.

Andersen, J., 1998. Baseline system for hybrid speech recognition on French, COM 98-7, IDIAP, Martigny, Switzerland.

Bellegarda, J.R., 1997. A statistical language modeling approach integrating local and global constraints. In: IEEE Workshop on Automatic Speech Recognition Understanding, pp. 262–269.

Bellegarda, J.R., 1999. Speech recognition experiments using multi-span statistical language models. In: Proc. IEEE Internat. Conf. on Acoust. Speech Signal Process. (ICASSP), Phoenix, Arizona, pp. 717–720.

Berry, M.W., 1992. Large-scale sparse singular value computations. Internat. J. Supercomput. Appl. 6 (1), 13–49.

Bourlard, H., Morgan, N., 1994. Connectionist Speech Recognition – A Hybrid Approach. Kluwer Academic Publishers, Dordrecht.

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

Coccaro, N., Jurafsky, D., 1998. Towards better integration of semantic predictors in statistical language modeling. In: Proc. Internat. Conf. on Spoken Lang. Process. (ICSLP'98), Sydney, Australia.

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

Garofolo, J.S., Voorhees, E.M., Auzanne, C.G.P., Stanford, V.M., 1999. Spoken document retrieval: 1998 evaluation and investigation of new metrics. In: ESCA ETRW Workshop on Accessing Informat. Spoken Audio, pp. 1–7.

Gillick, L., Cox, S., 1989. Some statistical issues in the comparison of speech recognition algorithms. In: Proc. IEEE Internat. Conf. on Acoust. Speech Signal Process. (ICASSP'89), Glasgow, Scotland, pp. 532–535.

Golub, G., Reinsch, C., 1971. Handbook for Matrix Computation II, Linear Algebra. Springer, New York.

Hofmann, T., 1998. Probabilistic latent semantic analysis. TR 98-042, International Computer Science Institute, Berkeley, California.

Hofmann, T., 1999. Probabilistic topic maps: Navigating through large text collections. In: Proc. Third Symp. on Intelligent Data Anal. (IDA'99), Amsterdam, Netherlands.

Honkela, T., Kaski, S., Lagus, K., Kohonen, T., 1996. Newsgroup exploration with WEBSOM method and browsing interface. Tech. Rep. A32, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland.

Johnson, S., Jourlin, P., Moore, G., Jones, K.S., Woodland, P., 1999. The Cambridge university spoken document retrieval system. In: Proc. IEEE Internat. Conf. on Acoust. Speech Signal Process. (ICASSP'99), Phoenix, Arizona, pp. 49–52.

Johnson, W., Lindenstrauss, J., 1984. Extensions of Lipshitz mapping into Hilbert space. Contemp. Math. 26, 189–206.

Kaski, S., 1998. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In: Proc. Internat. Joint Conf. on Neural Networks (IJCNN'98), Anchorage, Alaska, Vol. 1, pp. 413–418.

Kohonen, T., 1997. Self-organizing Maps, second extended ed. Springer, Berlin.

Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., Saarela, A., 1999. Self-organization of a massive text document collection. In: Oja, E., Kaski, S. (Eds.), Kohonen Maps. Elsevier, Amsterdam, pp. 171–182.

Kurimo, M., 1999. Indexing audio documents by using latent semantic analysis and SOM. In: Oja, E., Kaski, S. (Eds.), Kohonen Maps. Elsevier, Amsterdam, pp. 363–374.

Kurimo, M., Mokbel, C., 1999. Latent semantic indexing by self-organizing map. In: ESCA ETRW Workshop on Accessing Informat. Spoken Audio, Cambridge, UK, pp. 25–30.

Lagus, K., Kaski, S., 1999. Keyword selection method for characterizing text document maps. In: Proc. Internat. Conf. on Artificial Neural Networks (ICANN'99). Vol. 1. IEE, London, pp. 371–376.

Ng, K., Zue, V.W., 1998. Phonetic recognition for spoken document retrieval. In: Proc IEEE Internat. Conf. on Acoust. Speech Signal Process. (ICASSP'98), Seattle, Washington, pp. 325–328.

Papadimitriou, C., Raghavan, P., Tamaki, H., Vempala, S., 1998. Latent semantic indexing: A probabilistic analysis. In: Proc. 17th ACM Symp. on Principles of Database Systems, Seattle, USA (invited for publication in J. Comp. System Sci.).

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

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

Renals, S., Abberley, D., Cook, G., Robinson, T., 1998. THISL spoken document retrieval. In: Proc. Seventh Text Retrieval Conf. (TREC-7).

Ritter, H., Kohonen, T., 1989. Self-organizing semantic maps. Biol. Cyb. 61 (4), 241–254.

Robertson, S., Sparck-Jones, K., 1976. Relevance weighting of search terms. J. Amer. Soc. Inform. Sci. 27 (3), 129–146.

Robinson, T., Christie, J., 1998. Time-first search for large vocabulary speech recognition. In: Proc. IEEE Internat. Conf. on Acoust. Speech Signal Process. (ICASSP'98), Seattle, Washington, pp. 829–832.

Robinson, T., Hochberg, M., Renals, S., 1996. The use of recurrent networks in continuous speech recognition. In: Lee, C.H., Paliwal, K.K., Soong, F.K. (Eds.), Automatic Speech and Speaker Recognition – Advanced Topics. Kluwer Academic Publishers, Dordrecht, pp. 233–258, Chapter 10.

Robinson, T., Abberley, D., Kirby, D., Renals, S., 1999. Recognition, indexing and retrieval of british broadcast news with the THISL system. In: Proc. 6th Europ. Conf. on Speech Commun. Technol., Budabest, Hungary, pp. 1067–1070.

Salton, G., 1971. The SMART Retrieval System-Experiments in Automatic Document Processing. Prentice-Hall, Englewood Chiffs, NJ.

Siegler, M., Witbrock, M., 1999. Improving the suitability of imperfect transcriptions for information retrieval from spoken documents. In: Proc. IEEE Internat. Conf. on Acoust. Speech Signal Process. (ICASSP'99), Phoenix, Arizona, pp. 505–508.

Simula, O., Ahola, J., Alhoniemi, E., Himberg, J., Vesanto, J., 1999. Self-organizing map in analysis of large-scale industrial systems. In: Oja, E., Kaski, S. (Eds.), Kohonen Maps. Elsevier, Amsterdam, pp. 375–387.

Ultsch, A., 1999. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In: Oja, E., Kaski, S. (Eds.), Kohonen Maps. Elsevier, Amsterdam, pp. 33–45.

Xu, J., Croft, W.B., 1996. Query expansion using local and global document analysis. In: Proc. ACM SIGIR, pp. 4–11.