Article:
Gotoh, Y. and Renals, S. (1999) Topic-based mixture language modelling. Natural Language Engineering, 5 (4). pp. 355-375. ISSN: 1351-3249

https://doi.org/10.1017/S1351324900002278

Natural Language Engineering **5** (4): 355–375. Printed in the United Kingdom

Topic-based mixture language modelling

YOSHIHIKO GOTOH and STEVE RENALS

Department of Computer Science, University of Sheffield,
Regent Court, 211 Portobello Street, Sheffield S1 4DP, UK
e-mail: {y.gotoh, s.renals}@dcs.shef.ac.uk

(*Received June 1998; revised July 1999*)

Abstract

This paper describes an approach for constructing a mixture of language models based on simple statistical notions of semantics using probabilistic models developed for information retrieval. The approach encapsulates corpus-derived semantic information and is able to model varying styles of text. Using such information, the corpus texts are clustered in an unsupervised manner and a mixture of topic-specific language models is automatically created. The principal contribution of this work is to characterise the *document space* resulting from information retrieval techniques and to demonstrate the approach for mixture language modelling. A comparison is made between manual and automatic clustering in order to elucidate how the global content information is expressed in the space. We also compare (in terms of association with manual clustering and language modelling accuracy) alternative term-weighting schemes and the effect of singular value decomposition dimension reduction (latent semantic analysis). Test set perplexity results using the British National Corpus indicate that the approach can improve the potential of statistical language modelling. Using an adaptive procedure, the conventional model may be tuned to track text data with a slight increase in computational cost.

1 Introduction

A typical Large Vocabulary Continuous Speech Recognition (LVCSR) system exploits an *n*-gram Language Model (LM) for scoring hypotheses generated by acoustic analysis of speech data. The *n*-gram model is syntactic and locally constrained, based on a Markov chain of a word sequence whose parameters are derived from word frequency counts given a training corpus. Because the *n*-gram is a statistical model, a fundamental assumption is that the task domain for an LVCSR system is similar to that for the training corpus. Consequently, a relatively large amount of training data may be required to accommodate the great number of variations that occur in spoken language. The *n*-gram approach works very well when these underlying assumptions of static task domain and sufficient training data hold. However, it is difficult for *n*-gram based systems to deal with tasks in which the domain may vary from the training conditions. To address this problem, several adaptive language modelling schemes have been proposed, in which some notion of 'topic' is inferred from the local text. An adaptive language model probability is computed that has

some dependence on this topic. Since the *n*-gram model has a constrained context (typically, the previous two or three words) most adaptive language modelling schemes attempt to exploit longer distance dependencies in some way.

Approaches to adaptive language modelling usually have two components: the automatic derivation of topic information from text, and the combination of global and topic-dependent text statistics. The topic of a document[1] is often obtained using a model that incorporates long distance or document-wide statistics. The 'bag-of-words' model used in Information Retrieval (IR), which is based on a histogram of weighted unigram frequencies, is often employed to estimate the topic of a document. Schemes to combine information from different language models include mixture modelling and maximum entropy.

A mixture formulation is widely used in speech and natural language processing because it provides inference techniques through a sound statistical foundation (Titterington, Smith and Makov 1985). A typical approach involves partitioning a corpus (either manually or automatically) according to text content to produce a set of component LMs which are then blended to produce a mixture model. Such a scheme has been employed by Kneser and Steinbiss (1993) and Clarkson and Robinson (1997). The *dynamic cache model* is a related approach, based on an observation that recently appearing words are more likely to re-appear than those predicted by a static *n*-gram model. Such a model usually combines cached unigram statistics for recent words with the baseline *n*-grams (Kuhn and De Mori 1990; Kneser, Peters and Klakow 1997).

Maximum entropy techniques were introduced into language modelling by Rosenfeld (1996), in which longer distance dependencies were incorporated into a model structure using *trigger pairs*. More recently efficient maximum entropy methods have been used to explicitly incorporate topic-conditional constraints within an *n*-gram model (Khudanpur and Wu 1999).

Automatic determination of topic may be posed as a problem of document clustering based on content. The standard methods are those based on the bag-of-words statistical model used in IR, discussed below. However, more sophisticated statistical models have been applied to this problem. Pereira, Tishby and Lee (1993) developed a method for the soft clustering of words by modelling the distribution of cluster membership for each word, then measuring the distributional similarity using the relative entropy. This approach has recently been applied to document classification by Baker and McCallum (1998). The class based *n*-gram (Brown, Della Pietra, deSouza, Lai and Mercer 1992) is an alternative model that uses mutual information of adjacent classes for word classification. That work was re-formulated as an *aggregate Markov model* that was able to discover soft word classes using the Expectation-Maximisation (EM) procedure (Saul and Pereira 1997). Further, Hofmann and Puzicha (1998) discussed statistical modelling for data co-occurrence; distributional clustering, the aggregate model, and other approaches may be viewed as different aspects of the same framework.

[1] We use the term 'document' loosely; in a speech recognition application it may refer to a window of, say, 500 words (see section 4).

Most state-of-the-art IR systems exploit a model of word (or term[2]) co-occurrence, to measure the similarity between two documents[3]. The basic notion is that the similarity between the two pieces of text is related to the frequency of co-occurring words (van Rijsbergen 1979). The same basic approach may be interpreted either probabilistically or as a distance measure in a high dimensional space (whose dimension is given by the vocabulary size). To avoid distortions occurring due to common non-content words, document length, etc., a weighting function is usually applied. A well-known weighting scheme is referred to as $tf \cdot idf$, where the term frequency within a document (tf) is weighted by the inverse document frequency (idf) which is based on the number of documents a particular term appears in Yu and Salton (1977). This scheme has been applied to language modelling by Sekine and Grishman (1996), who used such an IR system to collect articles using recent keywords, from which dynamic topic-specific LMs were constructed, and by Seymore, Chen and Rosenfeld (1998), where probabilities for on-topic and off-topic words were modified by a nonlinear interpolation technique.

A related IR approach, Latent Semantic Analysis (LSA) (Deerwester, Dumais, Furnas, Landauer and Harshman 1990), estimates document similarity in a reduced dimension space obtained by calculating the principal components (i.e. eigenvectors) of the higher dimensional space (Joliffe 1986; Hinton, Dayan and Revow 1997). The principal components capture the largest variation of words and documents without sacrificing much information. This reduction from a high dimensional discrete space to a lower dimensional continuous space has been a controversial technique in IR (Berry, Dumais and O'Brien 1995; Schütze and Silverstein 1997; Hofmann 1999), not least because it lacks a rigorous statistical foundation (e.g. see Hinton et al. 1997). However, Tipping and Bishop (1999) have pointed out that a probabilistic model may be forced on principal component analysis. Regardless of this, a proper probability model may not be required for modelling word co-occurrence or document classification since an exact probability estimate is not needed if the captured variance is sufficiently large for discrimination.

For language modelling or speech processing applications, LSA was first adopted by Bellagarda, Butzberger, Chow, Coccaro and Naik (1996), and further developed by Bellagarda (1998). In that work, the global constraints derived from the LSA calculation were integrated into a conventional local n-gram model using the conditional probability relation. Experiments using the Wall Street Journal corpus resulted in a substantial improvement both in perplexity (Bellagarda 1998) and in word error rate (Bellagarda 1999). The handling of probability mass (derived directly from the vector representation of words and documents in a reduced space) is somewhat unconventional, however the results obtained indicate the potential of the technique. Coccaro and Jurafsky (1998) compared the LSA-derived unigram probability with the n-gram prediction, and found that the LSA prediction was often better for words having relatively high global term weights.

[2] More generally stopping and stemming algorithms may be used to process the sequence of words in a document before modelling (Frakes and Baeza-Yates 1992).

[3] For IR applications the query may be regarded as a document.

This paper investigates topic-based mixture language processing using IR statistical models. Three term weighting schemes are examined (raw unigram frequencies, Okapi and entropy-based), and for each scheme dimension reduction using LSA may also be employed. Term weighted (and possibly reduced) vector representations are then used as references for partitioning a training corpus to semantically meaningful document classes. Each component LM is a conventional, locally constrained n-gram model calculated from a set of automatically partitioned documents. IR techniques also provide a framework for selecting a closely matched LM component to any piece of text data. This mechanism allows the approach to track varying document style by indirectly incorporating global content information. The approach may be conservative in some sense; however it avoids a fragile probability formulation based on LSA. It is also possible to compare different term weighting schemes with and without LSA-based reduction in a uniform framework. Furthermore, the required computation should stay at a level suitable for a practical LVCSR system.

The principal contribution of this paper is to characterise the *document space* resulting from the IR modelling framework and to investigate a mixture language modelling approach suitable for LVCSR. The major focus is on the *British National Corpus* (BNC) (Burnard 1995), a large, general corpus that covers a variety of topics. A particularly useful feature of the BNC is that it contains manually tagged subject information. In section 2, the mixture language model is presented. Term weighting schemes and the LSA related issues relevant to this paper are described in section 3. Using the BNC, section 4 demonstrates that the approach can partition words and documents into clusters that may later be used for building semantic class oriented LMs. Finally, in section 5, experimental results (test set perplexities) show that the LM mixture outperforms the conventional method, indicating an advantage of a flexible and adjustable structure to a robust but inflexible model. An adaptive language modelling scheme is also investigated; causal content history from the text data is tracked using IR techniques, then the single conventional LM is tuned to the subject using a mixture approach. Best results are obtained when the Okapi term weighting formula is used without applying the LSA.

2 Mixture language model

Conventional n-gram language modelling exploits local constraints from a document, i.e., a sequence of words $\{w_1, \cdots, w_t, \cdots\}$. Its parameters, $f(w_t|w_1^{t-1})$, are derived from word frequency counts given a large collection of documents. In spite of its simplicity, the n-gram LM is very robust, and it has proved difficult to develop potentially more sophisticated models that consistently outperform it for large vocabulary speech recognition tasks (Jelinek 1991).

A mixture LM, denoted by $\mathscr{M}$, is constructed as the weighted sum of J component LMs, $< \mathscr{M}_1, \cdots, \mathscr{M}_J >$, derived from a partitioned corpus (Kneser and Steinbiss 1993; Clarkson and Robinson 1997). Partitioning may be done either by hand or by machine. Let $f(w_t|w_1^{t-1}; \mathscr{M})$ and $f(w_t|w_1^{t-1}; \mathscr{M}_j)$ imply n-gram type parameters for a mixture and its j^{th} component, respectively. Formally, a mixture LM used in this

paper is defined as

$$f(w_t|w_1^{t-1};\mathscr{M}) = \sum_{j=1}^{J} c_j f(w_t|w_1^{t-1};\mathscr{M}_j) \tag{1}$$

where c_j's are mixing factors that satisfy $\sum_{j=1}^{J} c_j = 1$.

Mixing factors can be estimated using the EM algorithm (Dempster, Laird and Rubin 1977). Given a mixture LM of form (1), then considering the likelihood function for a document, the problem is to find c_j's that maximise the likelihood. Suppose that there exist T words in the document, then the p^{th} estimate of c_j is given by

$$c_j^{[p]} = \frac{1}{T}\sum_{\tau=1}^{T} \frac{c_j^{[p-1]} f(w_\tau|w_1^{\tau-1};\mathscr{M}_j)}{\sum_{k=1}^{J} c_k^{[p-1]} f(w_\tau|w_1^{\tau-1};\mathscr{M}_k)} \tag{2}$$

starting from an appropriate initial condition $c_j^{[0]}$. When $\tau = 1$, an n-gram parameter for a component j is simply $f(w_1;\mathscr{M}_j)$. Note that a posterior mode may be used by combining some prior function in equation (2). The procedure is similar to other mixture density parameter estimation problem; further discussion should be referred to elsewhere[4] (e.g. Redner and Walker 1984).

The estimation formula for mixing factors, given by equation (2), produces the new estimates only after the whole documents are processed. It is not very convenient because a major objective for using the mixture language modelling approach is to flexibly adjust to the varying style of documents. Instead, described below is the alternative version that is slightly modified for an incremental text data input. Suppose that $t-1$ words $\{w_1, \cdots, w_{t-1}\}$ have been processed so far, and now a new word w_t is given. Then the t^{th} estimation for c_j is obtained recursively by

$$c_j^{[t]} = \frac{t-1}{t} c_j^{[t-1]} + \frac{1}{t}\gamma_j^{[t]} \tag{3}$$

where $\gamma_j^{[t]}$ is computed by

$$\gamma_j^{[t]} = \frac{c_j^{[t-1]} f(w_t|w_1^{t-1};\mathscr{M}_j)}{\sum_{k=1}^{J} c_k^{[t-1]} f(w_t|w_1^{t-1};\mathscr{M}_k)} . \tag{4}$$

Using equations (3) and (4), information from the word sequence is incorporated incrementally into the mixing factors.

The framework for mixture language modelling has been established. However,

[4] The normal density is probably the most widely used distribution among mixture parameter estimation problems. For a normal mixture, means and covariances, in addition to mixing factors, are often determined using the EM procedure. In contrast, the mixture LM problem here solely estimates mixing factors without re-calculating the component LMs.

there remain two problems that need to be addressed: (1) how to classify the documents into semantically meaningful clusters; (2) how to select the LM component that best fits to a given piece of text data. For the first problem alone, manually tagged documents might suffice. But those manual tags cannot be relied on for the second problem when novel text data needs to be processed (especially for an LVCSR application). As a consequence, an automatic scheme that is able to handle any documents in an unsupervised manner is preferred.

3 Modelling the document space

A first step for modelling the document space is to calculate weights for words according to their importance in documents. It is a focal point of document classification because it affects the notion of semantics expressed in the document space. For example, unigram frequencies of vocabulary items may be used. As the total word counts often vary in orders of magnitude between documents, estimates of unigram probabilities can be used instead in order to avoid possible effects of document size. These measures are based on the intuition that if two documents share many vocabulary items, then there is a good chance that they concern a similar subject[5]. IR techniques do this comparison mathematically at the word or document level. One advantage of using a unigram related measure is that local constraints (i.e. short-span ordering of vocabulary items on the Markov chain) that might have an adverse global effect may be discarded.

This paper also considers an application of another IR technique, Latent Semantic Analysis (LSA); it is based on the Singular Value Decomposition (SVD) of a very large, sparse, word by document matrix (Deerwester et al. 1990; Berry et al. 1995). Each column of the matrix describes a document, with the entries being some measure associated with vocabulary items in that document. The eigenvectors (i.e. principal components) corresponding to the s largest eigenvalues are then used to define s-dimensional word and document spaces, where s is typically of the order of 100. Put simply, the approach effectively models the co-occurrence of vocabulary items or documents provided by the very large matrix. The technique is referred to as 'latent semantic' because the projection to the lower dimensional subspace has the effect of clustering together semantically similar words and documents. IR performance data suggests that points in the derived subspace may be more reliable indicators of meaning than individual words (Deerwester 1990; Dumais 1991). Furthermore, assuming that a document is a linear combination of words (drawn from tens of thousands vocabulary items), it is possible to project any document down to a vector of a few hundred dimensions, regardless of whether it is included in the original matrix. A major advantage is that the lower dimensional document subspace is automatically inferred using the SVD.

[5] For example, suppose two documents share vocabulary items, say 'software' and/or 'internet', then the thriving contemporary industry is likely to be a topic for both.

3.1 Term weighting

When characterising a document by the unigram frequencies of the words within it, it would also be useful to weight the more important words. In a statistical approach to the problem such weighting must be done with respect to the training corpus, with the addition of any prior knowledge. In the field of information retrieval, *term weighting* schemes are used in which global and local factors are combined to produce weighting factors for the within-document unigram probabilities. This paper uses two such schemes, Okapi term weighting (Robertson and Spärck Jones 1997) and an entropy-based formula (Dumais 1991).

Suppose that g_i implies a global weight for a word w_i in a collection of documents and that l_{ij} is a local value within a certain document d_j. Then both schemes calculate a term weight a_{ij} as

$$a_{ij} = g_i \cdot l_{ij}. \tag{5}$$

The global weight is designed to enhance words which are not widely distributed across many documents. Using (5), a sparse document vector for d_j is defined as a collection of term weights $\boldsymbol{d}_j = \{a_{ij}\}$.

Okapi formula A simple but effective term weighting scheme was presented by Robertson, Walker, Jones, Hancock-Beaulieu and Gatford (1995), with further detail provided by Robertson and Spärck Jones (1997) and Spärck Jones, Walker and Robertson (1998). The scheme has been extensively tested through IR evaluation tasks such as *TREC* (Text Retrieval Conference). It defines a global weight as

$$g_i = \log N - \log n_i \tag{6}$$

where N is the total number of documents in the collection and n_i is the number of documents word w_i occurs in. This factor is known as the *collection frequency weight* or the *inverse document frequency*. Further, let c_{ij} be frequency counts for w_i in a document d_j. Then a local value is

$$l_{ij} = \frac{(k_1 + 1) \cdot c_{ij}}{k_1 \cdot \{(1 - k_2) + k_2 \cdot e_j\} + c_{ij}} \tag{7}$$

where a normalised document length is given by $e_i = \dfrac{m_j}{\bar{m}_j}$ with m_j and $\bar{m}_j$ being the number of words in d_j and its average for all documents, respectively. k_1 is a tuning constant; increasing k_1 would increase the influence of term frequency. The effect of document length may be modified by a constant k_2. When document length is fixed (as in this paper), $e_i = 1$ and equation (7) is reduced to

$$l_{ij} = \frac{(k_1 + 1) \cdot c_{ij}}{k_1 + c_{ij}}. \tag{8}$$

Entropy formula An entropy based approach has frequently been used in combination with LSA calculation (Dumais 1991; Bellagarda 1998). Let N denote the number of documents, and let t_i and c_{ij} also be frequency counts for w_i in entire collection of documents and in a document d_j. From the association with an entropy

factor, a global factor may be calculated by

$$g'_i = 1 - \left(-\frac{1}{\log N} \sum_{j=1}^{N} \frac{c_{ij}}{t_i} \cdot \log \frac{c_{ij}}{t_i} \right). \tag{9}$$

As for a local value, let m_j be the number of words in d_j and

$$l'_{ij} = \log \left(1 + \frac{c_{ij}}{m_j} \right) \tag{10}$$

will suffice.

3.2 Singular Value Decomposition

Term weighted document vectors $\boldsymbol{d}_j$ can be used directly when clustering documents. Alternatively, LSA may be applied to the word by document matrix $A = \{\boldsymbol{d}_j\}$ and clustering may be done in the reduced document space.

The principal computational burden of the LSA approach lies in the SVD of the word by document matrix. It is not unreasonable to expect the matrix to have dimensions of at least $20,000 \times 20,000$. However, this matrix is usually very sparse ($1 \sim 2\%$ or less of the elements are non-zero) and it is possible to perform such computations on a modern workstation (Berry, Do, O'Brien, Krishna and Varadhan 1993). Let A denote an $m \times n$ matrix (whose rank is r). Then it can be decomposed as

$$A = U \Sigma V^T \tag{11}$$

where V^T is the transpose of V. Σ is an $r \times r$ diagonal matrix whose non-zero elements correspond to the singular values, or the non-negative square roots of r non-zero eigenvalues for AA^T. U and V are $m \times r$ and $n \times r$ matrices whose rows may be referred to as word and document singular vectors. They define the orthonormal eigenvectors associated with the r eigenvalues of AA^T and $A^T A$, respectively.

The singular vectors corresponding to the s ($s \leq r$) largest singular values are then used to define an s-dimensional document space. Using these vectors, $m \times s$ and $n \times s$ matrices U_s and V_s can be redefined along with $s \times s$ singular value diagonal matrix Σ_s. It is then known that $\hat{A}_s = U_s \Sigma_s V_s^T$ is the closest matrix (in a least square sense) of rank s to the original matrix A (Berry et al. 1995). As a consequence, given an m-dimensional vector $\boldsymbol{d}$ that describes a document, it is warranted that an s-dimensional projection $\hat{\boldsymbol{d}}_s$ computed by

$$\hat{\boldsymbol{d}}_s = \boldsymbol{d}^T U_s \Sigma_s^{-1} \tag{12}$$

lies in the closest s-dimensional document subspace with respect to the original m-dimensional space. This is an important feature of the approach because a novel document (i.e. one that is not included in the original matrix A – possibly transcribed speech data in an LVCSR application) can be evaluated by calculating its document vector $\boldsymbol{d}$. The projection $\hat{\boldsymbol{d}}_s$ represents principal components that characterise 'semantic' information of the document.

Mixture LM Estimation

1. Form m-dimensional document vectors $\boldsymbol{d}$ from a training corpus (document collection): select vocabulary, stopping and stemming if required, and apply term weighting.
2. **Optional**: Apply SVD (11) to word-by-document matrix and obtain document projections $\hat{\boldsymbol{d}}_s$ in s-dimensional space.
3. Classify documents (whose vector representation may be either $\boldsymbol{d}$ or $\hat{\boldsymbol{d}}_s$) according to (13).
4. For each document class, build a class specific n-gram model from the corpus.

Adaptive Language Modelling

5. Form m-dimensional document vector for word sequence observed so far, applying term weighting, *etc*. If necessary, calculate the s-dimensional projection using equation (12).
6. Select the most suitable class specific LM component and blend together with a single LM estimated from the complete training corpus.
7. Calculate a score for the novel data.

Fig. 1. Procedures for generating a mixture LM (offline; steps 1–4) and language model adaptation (online; steps 5–7).

3.3 Clustering and mixture modelling

Words and documents can be classified according to their vector representations using the *k-means* clustering algorithm. A consistent distance measure is the cosine of the angle between two vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, i.e.,

$$\cos\phi = \frac{\boldsymbol{x}_1 \cdot \boldsymbol{x}_2}{||\boldsymbol{x}_1|| \cdot ||\boldsymbol{x}_2||} \,. \tag{13}$$

In this case, either the angle ϕ or simply $1 - \cos\phi$ may be sufficient for *k-means* clustering. $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ may be word or document vectors, either with or without SVD reduction. In the experiment, document classes and topic-based mixtures derived from the m-dimensional document vectors $\boldsymbol{d}$ are compared with those from the reduced s-dimensional document projections $\hat{\boldsymbol{d}}_s$.

Figure 1 outlines the procedure for generating a mixture LM and applying the resultant model for online adaptive language modelling. The most costly stage of the procedure is the optional step 2, which involves the large SVD computation. However, this is an offline procedure that needs only be applied once as the model estimation stage. The adaptive language modelling procedure does not require heavy computation. At step 6, the topic-dependent models are indirectly augmented with global content information.

4 The British National Corpus

To characterise the language modelling approach through IR techniques, this paper focuses on the *British National Corpus* (BNC) (Burnard 1995). It contains examples from both spoken and written British English, manually tagged with various levels of linguistic information. It is designed as a general corpus with a great variety of topics; it is not specifically restricted to any particular subject field or genre. The corpus comprises more than four thousand text files with a total of about 100

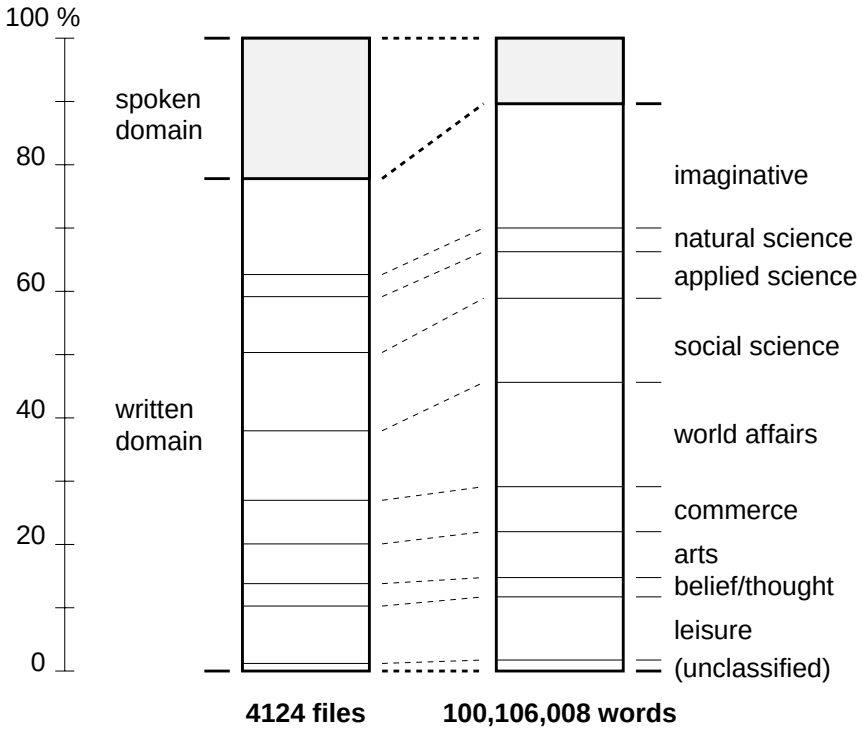

Fig. 2. The BNC contains 4124 text files with over 100 million words. It is manually tagged with various levels of linguistic information. In terms of the word count, approximately one tenth of the corpus are the manually transcribed 'spoken' texts. The rest are the 'written' part, consisting of nine subject fields. Further breakdown of linguistic information and detailed statistics may be found in the BNC Users Reference Guide (Burnard 1995).

million words. Figure 2 illustrates the distribution of subject fields according to their hand-labelled information. They are referred to as *domains* to distinguish them from the document *classes* automatically inferred using IR techniques.

4.1 Corpus partition

The corpus was separated randomly (and independent of manually tagged domains and of automatically inferred classes) as follows; 3308 text files (80%) for LM generation and 400 text files (10%) for LM evaluation. The remainder (420 files) was held out for future use. The whole corpus contained approximately 360,000 independent words, out of which 19,945 words were selected as a vocabulary in unigram frequency order. Out of vocabulary (OOV) words were treated as an 'unknown' category. These conditions were maintained throughout the course of experiments described in this paper.

The main objective for using the BNC was to compare the automatically derived document classes with manually tagged domain information. They were constructed as follows;

Manually tagged domains Following Clarkson and Robinson (1997), text files in the LM generation set were classified into ten domains – one 'spoken' domain and nine domains from 'written' part of the BNC (figure 2). Note that the BNC is a relatively balanced general corpus; the word count for each domain differs by less than an order of magnitude. The smallest domain (i.e. belief/thought, containing slightly more than three million words) is probably sufficient for generating a component level LM for the use of mixture LM experiments in section 5.

Automatically inferred classes Because each text file in the BNC contained tens to hundreds of thousands words, they were subdivided mechanically into shorter units so that varying styles of text could be tracked. To this end, it could have been possible to use some linguistic information, such as context cue '<*div*>' embedded in the corpus texts (Burnard 1995). However, as such information is usually not available when processing novel data, a fixed size window (of either 200, 500, 1000, or 2000 words) approach was adopted. Windows were shifted along text files without any overlap. For example, using the 1000-word window, 3308 text files in the LM generation set were divided into 87,149 units. Those units were referred to as 'documents'.

From each piece of text segmented by a fixed size window, document vectors were derived using the three term weighting schemes. The document vectors were 19,945-dimensional (i.e. the same as the vocabulary size). When LSA processing was required, 40,000 documents were randomly chosen and $19,945 \times 40,000$ word by document matrix was generated. It was very sparse; for a 1000-word window case, approximately 1.6% of the matrix elements were non-zero. The SVD was applied, computing the 200 largest singular values and their corresponding singular vectors[6]. Using equation (12), all document units were projected on to the 200-dimensional space. Finally, 19 945-dimensional document vectors (or their corresponding 200-dimensional projections) were clustered into 10 to 1000 classes using the *k-means* algorithm with the cosine distance measure.

4.2 Word classes

First, semantic classification of vocabulary items was demonstrated. After segmentation by the 1000-word window, vocabulary words were weighted using the Okapi term weighting formula (7) with $k_1 = 10$. SVD processing was applied and 19,945 words were clustered into 1000 classes (approximately 20 vocabulary items per each class in average) using the word singular vectors. The following cluster was found among them:

{april, august, december, february, january, july, june, march, november, october, september}.

[6] This computation was achieved by a publicly available package, *SVDPACKC* (Berry et al. 1993).

This 11-word cluster consisted of months of a year with one exception of 'may', which was found in the cluster containing many verbs such as 'are', 'tend' and 'vary'. Another example:

{comet, cooled, core, cosmic, earth, equator, furthest, improbable, jupiter, lunar, mars, mercury, moon, nasa, planet, planetary, planets, satellites, solar, terrestrial, tidal, venus}.

As noted earlier, the IR techniques model the co-occurrence of words between documents, implying that words belonging to the same class tend to appear in the same document. With that in mind, there weren't many surprises either in this second example, where names of our planetary system and related words are classified. Some planet names (such as 'saturn', 'uranus', 'neptune', 'pluto') were missed from the list – as a matter of fact they were not included in the vocabulary; even if they had, they could have been clustered together with other mythological words rather than those for the planetary system.

Each term weighting scheme was tested for classification of vocabulary items. It is difficult to show any statistical picture, however, by inspection, most of the word clusters generated were sensible and not very difficult to guess the common concept among the members regardless of weighting scheme. Many clusters also contained a few isolated 'spurious' words. For example, a word 'improbable' might not be very intuitive among the second collection shown above. But, perhaps, it might not be totally 'improbable' either that the word was frequently used in the documents discussing about our planetary system.

4.3 Association between classes and domains

Figure 3 shows the association between automatically inferred document classes and manually tagged domains. A large circle implies a strong association between the corresponding domain and class. For example, many documents in class 4 came from either applied science, social science or commerce, and those in class 5 were from world affairs. On the other hand, most documents in the spoken domain were identified as class 0, while most of those in the imaginative domain have fallen in either class 1 or class 9. Note that this figure corresponds to document clustering with a 1000-word window, Okapi term weighting ($k_1 = 10$), and SVD reduction.

Association factor It is not straightforward to compare how closely or loosely these domains and classes are associated just by observing the sizes of circles in pictures such as figure 3. To quantify the strength of association, an entropy based factor has been described in Press, Flannery, Teukolsky and Vetterling (1988), which they refer to as the *uncertainly coefficient*: let $\mathscr{I}$ and $\mathscr{J}$ denote document classification for manual domains and automatic classes. Define probabilities that a document is classified to domain i (regardless of class) and class j (regardless of domain) by $p_{i\bullet}$ and $p_{\bullet j}$. Then, entropies for partitions $\mathscr{I}$ and $\mathscr{J}$ are given by $H(\mathscr{I}) = \sum_i p_{i\bullet} \log p_{i\bullet}$ and $H(\mathscr{J}) = \sum_j p_{\bullet j} \log p_{\bullet j}$, respectively. Further, denoting the probability of a document being classified as domain i with class j as p_{ij}, a joint entropy for $\mathscr{I}$

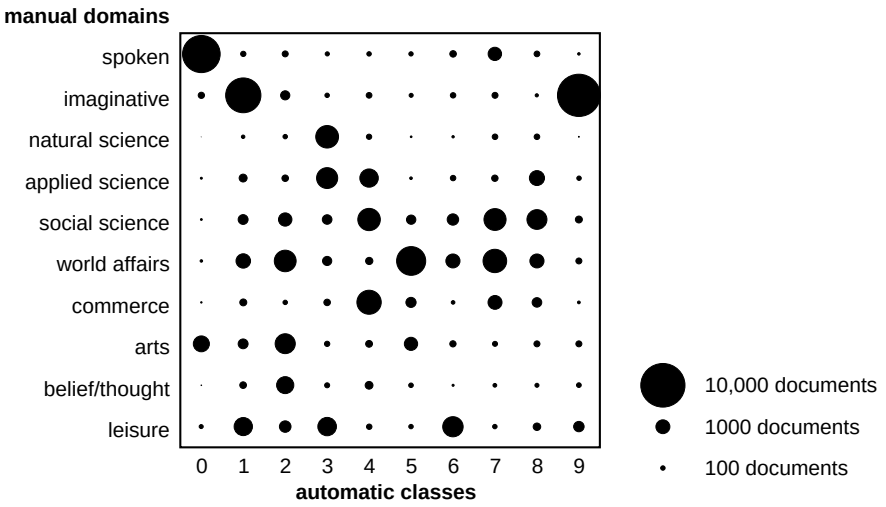

Fig. 3. Association between 10 automatically inferred classes and 10 manually tagged domains. Texts in the generation set were segmented using the 1000-word window, resulted in 87,149 'document units' in total. The Okapi term weighting formula (7), $k_1 = 10$ was used, then an SVD was applied. The area size of each circle corresponds to the number of document units belonging to that class/domain.

with $\mathscr{J}$ may be obtained by $H(\mathscr{I}, \mathscr{J}) = \sum_{ij} p_{ij} \log p_{ij}$. Using these entropies, the association factor between $\mathscr{I}$ and $\mathscr{J}$ is calculated by

$$A(\mathscr{I}, \mathscr{J}) = 2 \times \frac{H(\mathscr{I}) + H(\mathscr{J}) - H(\mathscr{I}, \mathscr{J})}{H(\mathscr{I}) + H(\mathscr{J})} \tag{14}$$

This measure varies from zero (no association) to one (completely dependent) according to the strength of association between $\mathscr{I}$ and $\mathscr{J}$.

Table 1 summarises the association factors between 10 automatic classes and 10 manual domains. When the SVD was not applied, the Okapi term weighting formula seemed to generate document clusters that are the closest to the manual classification. The simple unigram frequency scheme resulted in the lowest association factor, but nevertheless was able to pick up some amount of contents from documents. The SVD contributed significantly for the unigram frequency case, but adversely affected both the Okapi and entropy term weighting schemes. This result alone indicates the effectiveness of the Okapi formula. In section 5, the association factor will further be discussed in relation with mixture LM perplexity.

5 Mixture language modelling experiments

This section continues experiments using the BNC. It was demonstrated in section 4 that IR techniques could be used for clustering documents to automatically derive topic-based classes. Here, trigram based component LMs were constructed from document clusters, appropriate discounting and smoothing techniques were applied, then various types of mixture models were evaluated.

Table 1. *Association factors between 10 automatic classes and 10 manual domains for each term weighting scheme with and without SVD reduction. Note that figure 3 correspond to the Okapi formula* ($k_1 = 10$) *with SVD, which has an association factor of 0.351*

Term weighting scheme	Without SVD	With SVD
Unigram frequency	0.247	0.385
Okapi formula ($k_1 = 10$)	0.428	0.351
Entropy formula	0.363	0.325

5.1 Blind mixture updating

To set a baseline for the rest of experiments, a single trigram based model was derived from the complete LM generation set. This LM is referred to as a 'single conventional LM'. Its perplexity was 180.0 for texts in the LM evaluation set (which consisted of 400 text files, and just over 10 million words).

Table 2 shows the perplexities for mixtures of 10 manually tagged domain LMs and 10 automatically inferred class LMs, constructed as in section 4. For class LMs, the corpus was first segmented using the 1000-word window, and the Okapi term weighting formula ($k_1 = 10$) was applied, forming 87,149 document vectors. An SVD was *not* calculated at this moment. Document vectors were clustered into 10 'semantic' classes, then for each class a component LM was constructed from the corresponding segments of corpus texts. Mixture calculation was done 'blindly', i.e. domain or class information was not considered for pieces of texts in the evaluation set. Initially, the mixing factors were set proportional to the entire *n*-gram size for each component[7]. During the evaluation process they were updated blindly, but incrementally, using equation (3). This implies that the mixing factors were adjusted without identifying their domains or classes. It was found that the perplexity was 172.6 for the mixture of 10 manual domain LMs, better than the single conventional model. Furthermore, the manually tagged domain approach was improved upon by the mixture of 10 automatic class LMs, which achieved a perplexity of 164.2.

Table 2 also shows trigram hit rates for all approaches. Those for mixture models are averages from components (weighted by mixing factors) – they have reached just over 42%, approximately 20% lower than the single conventional model (62.4%). For many cases, the hit rate for higher order *n*-grams is a good indicator for the performance (e.g. the perplexity, as well as the Word Error Rate (WER) for speech recognition systems); it can even be said this is the major reason why a larger corpus is preferred for LM generation. When the corpus is partitioned to smaller subsets, lower hit rates are implicitly unavoidable because a certain cutoff level needs to be applied by a discounting scheme. Despite this implicit handicap (and it was as much

[7] This initialisation scheme was based on a crude assumption that the entire *n*-gram size might be a reasonable indicator of performance for each component LM.

Table 2. *Perplexities and trigram hit rates for single and mixture LM approaches. For 'automatically inferred class LMs', the Okapi term weighting formula* ($k_1 = 10$) *was used and an SVD was* not *applied. Mixture calculation was done 'blindly'. Trigram hit rates for mixture models are averages from components weighted by corresponding mixing factors*

Model	Perplexity	Trigram hit (%)
Single conventional LM	180.0	62.4
Mixture of		
10 manually tagged domain LMs	172.6	42.7
10 automatically inferred class LMs	164.2	42.8

as 20% absolute against the conventional approach), both mixture models have shown improved perplexities. This suggests the advantage of the mixture approach built from domain/class specific models which are better matched to a task.

Variations for automatic class generation Several approaches were discussed in section 3 for automatic clustering. Figure 4 compares mixtures of 10 class LMs generated using unigram relative frequencies, the Okapi term weighting formula ($k_1 = 10$), and the entropy formula. Window sizes of 200, 500, 1000 and 2000 words were applied for text segmentation, and they were tested with and without SVD reduction. As described earlier, mixture calculation was done blindly and the initial mixing factors were set proportional to the entire model size for each component. Figure 4 indicates that these mixtures resulted in a lower perplexity than the conventional single model approach. As for the text segmentation size, a 1000-word window seems a good choice; 2000 words may be too coarse to track the varying style of texts, while 200-word window is probably not large enough to capture the characteristics from each document unit.

Among these three term weighting schemes, the Okapi formula seems a better choice than the other two (when the SVD reduction was not applied). Unsurprisingly, the simple unigram frequency scheme performed worst although it was still an improvement over the conventional single model approach. When the SVD was used the picture changed, resulting in an improvement for the unigram weighting scheme, but an increased perplexity for the Okapi and entropy formula cases. As a consequence, there existed hardly any difference in performance among these three. The SVD calculation seemed to neutralise the effect of term weighting. By applying the SVD, the mixture LM approach outperformed the single model regardless of term weighting schemes or window sizes. On the other hand, none of them performed as well as the case using the Okapi formula and no SVD reduction.

In spite there being consistent differences in perplexity scores among term weighting schemes and SVD calculation, the trigram hit rate remains approximately the same level (42%) for all cases. Further, the Okapi formula with k_1 ranging from 2 to 200 was also tested for a 1000-word window. It was found that deviations from

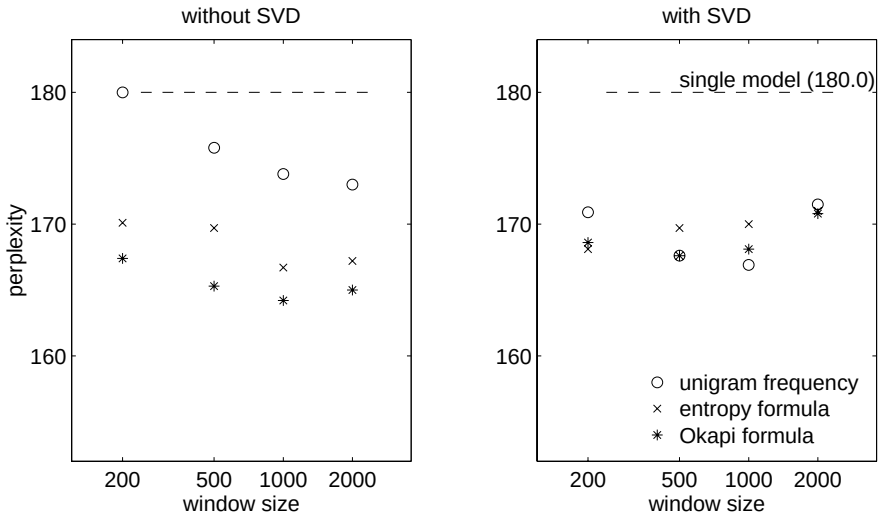

Fig. 4. Mixture LM perplexities for several variations of automatic class generation using three term weighting schemes (unigram relative frequencies, Okapi formula with $k_1 = 10$, and entropy formula) with and without SVD calculation. Window sizes were 200, 500, 1000 or 2000 for text segmentation, and the number of mixture components was 10 for all cases.

the $k_1 = 10$ case were well below 1% for all cases both with and without SVD processing.

LM perplexity and association factor The association factor between domains and classes, $A(\mathscr{I}, \mathscr{J})$, was described in section 4. Figure 5 shows the relation between mixture LM perplexities and association factors, with and without SVD calculation. Each figure contains cases using unigram relative frequencies, the entropy formula, and the Okapi formula with several different values of k_1. When the SVD was not applied, there was a (near-)linear relation between the mixture LM perplexities and association factors, regardless of term weighting scheme. Deviations from linearity were not very significant when the SVD was applied.

These results suggest that the association factor is a reasonably good predictor for the mixture LM performance. On the other hand, the perfect match between domains and classes will not produce good results as the mixture of 10 domain models in Table 2 indicates.

5.2 LM adaptation

This section describes the mixture based adaptive language modelling approach for tracking varying styles of text. It can be used when an automatic procedure is available for classifying documents. Experiments here made use of class information derived from segmented evaluation texts.

The approach is illustrated in figure 6. The offline procedure resulted in the single trigram based conventional LM and a set of class specific component LMs. The 1000-word window was applied for text segmentation. The LM adaptation

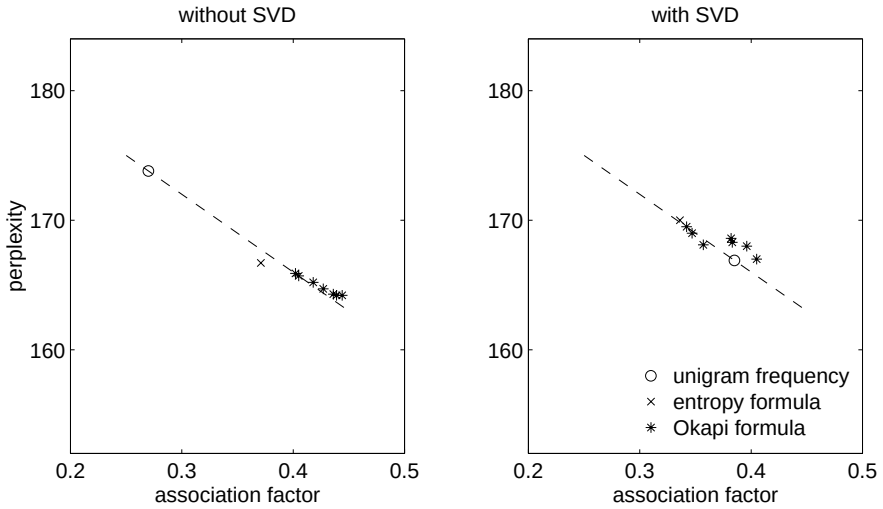

Fig. 5. Mixture LM perplexities and class/domain association factors for automatic class LMs using three term weighting schemes (unigram relative frequencies, Okapi formula with $k_1 = 2, 5, 10, 20, 50, 100, 200$ and entropy formula) with and without SVD calculation. A 1000-word window was used for text segmentation, and the number of mixture components was 10 for all cases.

experiment proceeded as follows: a recently observed piece of text was selected first, an appropriate term weighting scheme was applied, and then the closest class LM (to the segmented document) was blended with the single conventional model, resulting a mixture of two LMs. If necessary, the SVD projection was calculated after term weighting. Otherwise, the mixture modelling framework was the same including the mixing factor initialisation and updating procedures.

Figure 7 shows perplexities using three term weighting schemes, with and without SVD projection. The number of class specific LMs was set between 10 and 1000. The figure implies that mixture based LM adaptation worked better than the single model by a fair margin. By blending with the appropriate choice of class specific LM, the perplexity of the single conventional model has decreased from 180 down to around 160–170. In comparison to the blind mixture approach, the performance for adaptive modelling was best when the number of class LMs was relatively small, then gradually declined as the number grew. Because training corpus size is an important factor for statistical LM processing (the larger the better, in general), too many small class LMs might not contribute very much for adaptation. On the other hand, a very large class LM derived from a large collection of documents might not be very useful either as it could overlap in great part with the conventional model, losing the class oriented characteristics.

When SVD reduction was not applied, there existed a clear difference between the three term weighting schemes. For any number of class LMs, the Okapi and entropy formulae resulted in lower perplexities than the unigram frequency case. SVD projection resulted in lower perplexity for the unigram frequency scheme (particularly when the number of classes was not too large), but increased the

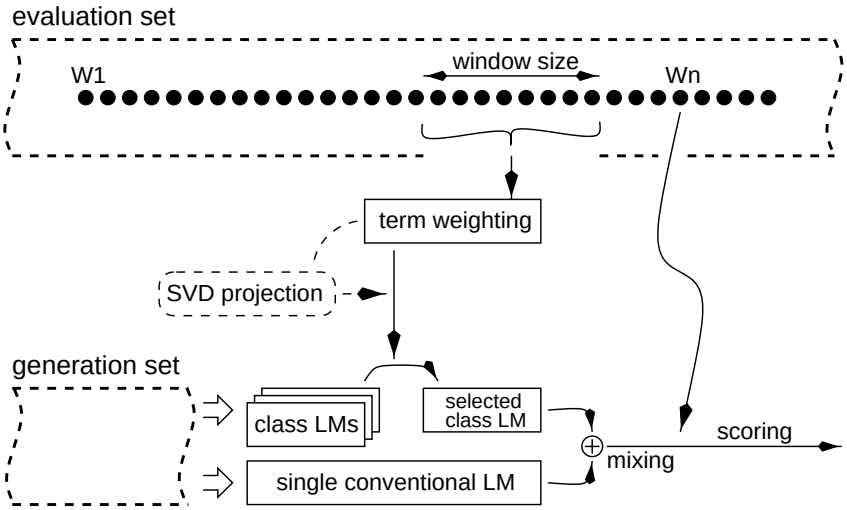

Fig. 6. Mixture based adaptive language modelling using document class information. The single trigram based LM and a set of class specific component LMs were produced offline. The evaluation was online; a recently processed piece of text was selected first, an appropriate term weighting scheme was applied with the option of SVD calculation, and finally the closest class LM (to the segmented document) was blended with the single conventional model.

perplexity for the Okapi and entropy weightings. The Okapi term weighting scheme resulted in the lowest perplexities both with and without SVD dimension reduction.

Finally, we note that this adaptive language modelling scheme has a computational advantage over the full mixture model, since it involves only two LMs (the single conventional model and the selected class model) for each word.

6 Conclusion

In this paper an approach to topic-based language models has been explored using mixture modelling, together with simple statistical models of semantics that have been developed in the field of information retrieval. These bag-of-words models involved term weighting of unigram statistics from corpus documents, and optionally projecting the high dimensional discrete space into a much lower dimensional continuous document space using the SVD of a very large, sparse word by document matrix. A corpus could thus be represented as a set of document vectors. Demonstrations using the BNC indicated that some part of the meaning of a text can be extracted using this simple statistical model.

IR models were incorporated into a conventional *n*-gram model of language, as used in speech recognition, as a basis for discrimination between documents, and a mixture LM was constructed in an unsupervised manner. The mixture was able to rely on the overall, broad structure of the conventional model estimated from the entire training corpus together with the better-fitted parameters of the relevant class model. Results from blind mixture experiments indicated that the approach could improve the potential of language modelling over the conventional method. Using

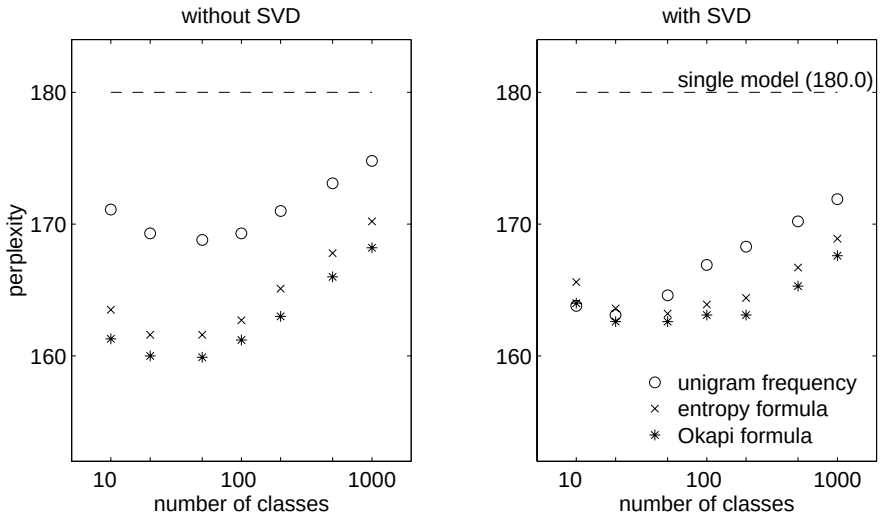

Fig. 7. Perplexities for mixture based LM adaptation using three term weighting schemes (unigram relative frequencies, Okapi formula with $k_1 = 10$, and entropy formula) with and without SVD calculation. The text segmentation window size was fixed to 1000, and the number of class specific LMs was set between 10 and 1000. For comparison, the perplexities for the single conventional LM and for the mixtures of 10 class LMs (Okapi formula with $k_1 = 10$, no SVD) were 180.0 and 164.2, respectively. (See Table 2.)

an adaptive procedure, the conventional model was tuned to track text data with a slight increase in computational cost.

The main results of the paper have involved the comparison of different term weighting schemes and the effect of SVD dimension reduction. Our principal conclusions are:

1. Automatic clustering based on simple unigram statistical models results in language models of lower perplexity compared with manual clustering into topics.
2. The Okapi term-weighting approach (7) consistently resulted in topic language models with the lowest perplexity (and had the closest association with manual clustering into topics).
3. SVD dimension reduction was only helpful in combination with unsophisticated term weighting schemes. It had a negative effect when used with the Okapi or entropy formulae.

Acknowledgements

This work was supported by ESPRIT long term research project SPRACH (EP20077) and UK EPSRC grant GR/M36717.

References

Baker, L. D. and McCallum, A. K. (1998) Distributional clustering of words for text classification. *Proceedings of SIGIR'98*, Melbourne, pp. 96–103.

Bellegarda, J. R. (1998) A multi-span language modeling framework for large vocabulary speech recognition. *IEEE Trans. Speech and Audio Processing* **6**(5): 456–467.

Bellegarda, J. R. (1999) Speech recognition experiments using multi-span statistical language models. *Proceedings of ICASSP-99*, **II**, Phoenix, pp. 717–720.

Bellegarda, J. R., Butzberger, J. W., Chow, Y.-L., Coccaro, N. B. and Naik, D. (1996) A novel word clustering algorithm based on latent semantic analysis. *Proceedings of ICASSP-96*, **1**, Atlanta, pp. 172–175.

Berry, M., Do, T., O'Brien, G., Krishna, V. and Varadhan, S. (1993) SVDPACKC (version 1.0) user's guide. Technical Report CS-93-194, University of Tennessee, Department of Computer Science. (Available from http://www.cs.utk.edu/ library/1993.html.)

Berry, M. W., Dumais, S. T. and O'Brien, G. W. (1995) Using linear algebra for intelligent information retrieval. *SIAM Rev.* **37**(4): 573–595.

Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C. and Mercer, R. L. (1992) Class-based *n*-gram models of natural language. *Computational Linguistics* **18**(4): 467–479.

Burnard, L. (1995) *Users Reference Guide, British National Corpus Version 1.0.* Oxford University Computing Service.

Clarkson, P. R. and Robinson, A. J. (1997) Language model adaptation using mixtures and an exponentially decaying cache. *Proceedings of ICASSP-97*, **2**, Munich, pp. 799–802.

Coccaro, N. and Jurafsky, D. (1998) Towards better integration of semantic predictors in statistical language modeling. *Proceedings of ICSLP-98*, **6**, Sydney, pp. 2403–2406.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R. (1990) Indexing by latent semantic analysis. *J. Soc. Infor. Sci.* **41**(6): 391–407.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statistical Soc., series B* **39**(1): 1–38.

Dumais, S. T. (1991) Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, and Computers* **23**(2): 229–236.

Frakes, W. B. and Baeza-Yates, R. (1992) *Information Retrieval: Data Structures and Algorithms.* Englewood Cliffs, NJ: Prentice Hall.

Hinton, G. E., Dayan, P. and Revow, M. (1997) Modeling the manifolds of images of handwritten digits. *IEEE Trans. Neural Networks* **8**(1): 65–74.

Hofmann, T. (1999) Probabilistic latent semantic indexing. *Proceedings of SIGIR'99*, Berkeley, CA, pp. 50–57.

Hofmann, T. and Puzicha, J. (1998) Statistical models for co-occurrence data. *Technical Report A.I. Memo No. 1625*, Massachusetts Institute of Technology, Artificial Intelligence Laboratory. (Available from http://www.ai.mit.edu/pubs.html.)

Jelinek, F. (1991) Up from trigrams! The struggle for improved language models. *Proceedings of Eurospeech-91*, **3**, Genova, pp. 1037–1040.

Joliffe, I. T. (1986) *Principal Component Analysis.* Springer Series in Statistics. Berlin: Springer Verlag.

Khudanpur, S. and Wu, J. (1999) A maximum entropy language model integrating n-grams and topic dependencies for conversational speech recognition. *Proceedings of ICASSP-99*, **1**, Phoenix, pp. 553–556.

Kneser, R., Peters, J. and Klakow, D. (1997) Language model adaptation using dynamic marginals. *Proceedings of Eurospeech-97*, **4**, Rhodes, pp. 1971–1974.

Kneser, R. and Steinbiss, V. (1993) On the dynamic adaptation of stochastic language models. *Proceedings of ICASSP-93*, **II**, Minneapolis, pp. 586–589.

Kuhn, R. and De Mori, R. (1990) A cache-based natural language model for speech recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* **12**(6): 570–583.

Pereira, F., Tishby, N. and Lee, L. (1993) Distributional clustering of English words. *Proceedings of ACL-93*, Columbus, OH, pp. 183–190.

Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T. (1988) *Numerical Recipes in C*. Cambridge, UK: Cambridge University Press.

Redner, R. A. and Walker, H. F. (1984) Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.* **26**(2): 195–239.

Robertson, S. E. and Spärck Jones, K. (1997) Simple, proven approaches to text retrieval. *Technical Report TR356*, University of Cambridge, Computer Laboratory. (Available from http://www.ftp.cl.cam.ac.uk/ftp/papers/reports/.)

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M. and Gatford, M. (1995) Okapi at TREC-3. *Overview of the 3rd Text Retrieval Conference (TREC-3)*, pp. 109–126.

Rosenfeld, R. (1996) A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language* **10**: 187–228.

Saul, L. and Pereira, F. (1997) Aggregate and mixed-order markov models for statistical language processing. *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, Providence, RI, pp. 81–89.

Schütze, H. and Silverstein, C. (1997) Projections for efficient document clustering. *Proceedings of SIGIR'97*, Philadelphia, pp. 74–81.

Sekine, S. and R. Grishman (1996, February). NYU language modeling experiments for the 1995 CSR evaluation. *Proceedings of DARPA Speech Recognition Workshop*, Harriman, NY, pp. 123–128.

Seymore, K., Chen, S. and Rosenfeld, R. (1998) Nonlinear interpolation of topic models for language model adaptation. *Proceedings of ICSLP-98*, **6**, Sydney, pp. 2503–2506.

Spärck Jones, K., Walker, S. and Robertson, S. E. (1998) A probabilistic model of information retrieval: Development and status. *Technical Report TR446*, University of Cambridge, Computer Laboratory. (Available from http://www.ftp.cl.cam.ac.uk/ftp/papers/reports/.)

Tipping, M. E. and Bishop, C. M. (1999) Mixtures of probabilistic principal component analyzers. *Neural Computation* **11**: 443–482.

Titterington, D. M., Smith, A. F. M. and Makov, U. E. (1985) *Statistical Analysis of Finite Mixture Distributions*. Chichester: John Wiley & Sons.

van Rijsbergen, C. J. (1979) *Information Retrieval* (2nd ed.). London: Butterworths.

Yu, C. T. and Salton, G. (1977) Effective information retrieval using term accuracy. *Comm. ACM* **20**: 135–142.