DOCUMENT SPACE MODELS USING LATENT SEMANTIC ANALYSIS

Yoshihiko Gotoh *Steve Renals* *

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

ABSTRACT

In this paper, an approach for constructing mixture language models (LMs) based on some notion of semantics is discussed. To this end, a technique known as *latent semantic analysis* (LSA) is used. The approach encapsulates corpus-derived semantic information and is able to model the varying style of the text. Using such information, the corpus texts are clustered in an unsupervised manner and mixture LMs are automatically created. This work builds on previous work in the field of information retrieval which was recently applied by Bellegarda et. al. to the problem of clustering words by semantic categories. The principal contribution of this work is to characterize the *document space* resulting from the LSA modeling and to demonstrate the approach for mixture LM application. Comparison is made between manual and automatic clustering in order to elucidate how the semantic information is expressed in the space. It is shown that, using semantic information, mixture LMs performs better than a conventional single LM with slight increase of computational cost.

1. INTRODUCTION

Mixtures of language models (LMs), based on some notion of semantics, have recently been proposed as an approach to dealing domain adaptation [1, 2]. These approaches involve partitioning the corpus, according to the style of text, to produce a set of component LMs, which are then blended together to produce a mixture LM. Although relatively rare, some text corpora include manual tagging of articles by subject (e.g., the British National Corpus — introduced in Section 2). However, hand-labeled style may not necessarily produce the best partitions for use of conventional statistical speech recognition application. Furthermore, it may be quite difficult to track the varying style of texts. As a consequence, it is clearly of interest to develop an automatic method for clustering the corpus texts in an unsupervised manner. Unigram counts usually form the root of such automatic approaches, either via a straightforward clustering [2] or further elaborating with a normalization term that discounts high frequency words [1].

In this paper, the problem is approached through the construction of *document space* model that encapsulates corpus-derived semantic information. Once a consistent and powerful model is constructed, it can be applied for a number of language modeling tasks. In particular, it is straightforward to develop mixture LMs that are tuned to the varying style of the text.

To this end, an approach to information retrieval (IR) known as *latent semantic analysis* (LSA) is used in order to uncover semantic information from the corpus [3, 4]. The technique was recently used by Bellegarda et. al. for semantic word clustering in speech recognition [5]. The principal contribution of this work is to characterize the *document space* resulting from the LSA modeling and to demonstrate the approach for mixture LM application. The method for constructing this space is addressed. The hand-labeled corpus is then used in the experiments, hoping the comparison between manual and automatic approaches may elucidate how the semantic information is expressed in the space.

*This work was supported by "SPRACH" ESPRIT project 20077.

domain		#texts	#words
spoken	total	915	10 365 464
written	*imaginative*	625	19 664 309
	natural science	144	3 752 659
	applied science	364	7 369 290
	social science	510	13 290 441
	world affairs	453	16 507 399
	commerce	284	7 118 321
	arts	259	7 253 846
	belief/thought	146	3 053 672
	leisure	374	9 990 080
	total	3 209	89 740 544
all		4 124	100 106 008

Table 1. British National Corpus (BNC) is hand-labeled into domains with wide range of topic. Total in "written" part includes "unclassified" texts. These numbers are provided by Users Reference Guide [6]. Accusal counts may slightly vary according to how one processes the corpus texts.

2. BRITISH NATIONAL CORPUS (BNC)

Main focus of this work is on the British National Corpus (BNC) [6]. It contains examples of both spoken and written British English, manually tagged with the various level of linguistic information. It is a general corpus; it does not specifically restricted to any particular subject field, or genre. The corpus comprises of more than four thousand texts with about one hundred million words, which were hand-labeled into domains shown in Table 1.

4142 BNC texts were partitioned in random (and independent of hand-labeled domain information) as follows;

generation set: 3309 texts (80 %) for LM generation.

evaluation set: 400 texts (10 %) for LM evaluation.

The rest (419 texts) of the corpus were held out for future use. Also note that the whole corpus contains about 360000 independent words, out of which 19989 words were selected as a vocabulary in the unigram frequency order. Out of vocabulary (OOV) words were treated as an "unknown". This partition and vocabulary were maintained throughout the course of experiments described in this section and in Section 4.

2.1. Mixture LM

A mixture LM, $\mathcal{M}$, is constructed as the weighted sum of component LMs $< \mathcal{M}_1, \cdots, \mathcal{M}_j, \cdots >$ derived from

model		perplexity
single	"full LM"	186.9
mixture	10 domain LMs	178.8
	10 domain LMs & "full LM"	170.1

Table 2. This table shows perplexities for single and mixture LMs. Hand-labeled domain information was used for creating the mixture LMs, with and without "full LM".

the partitioned corpus (either hand-labeled or automatic) [2]. Given a document, i.e., a sequence of words $<w_1, \cdots, w_i, \cdots>$, it is computed using the conventional trigram LMs by

$$f(w_i|w_{i-2}, w_{i-1}; \mathcal{M}) = \sum_j c_j f(w_i|w_{i-2}, w_{i-1}; \mathcal{M}_j) \quad (1)$$

where c_j is a mixing factor such that $\sum_j c_j = 1$.

Mixing factors c_j are tuned on-the-fly to the previously processed part of the document using the expectation-maximization (EM) type algorithm [7]. Suppose n words, $< w_1, \cdots, w_n >$, have been processed from the beginning. Then, considering the likelihood function $f(w_1, \cdots, w_n|\mathcal{M})$ for the mixture LM, it is straightforward to derive incrementally adjusting formula for $c_j^{(n)}$;

$$c_j^{(n)} = \frac{1}{n} \sum_{i=1}^{n} \gamma_j(i) \quad (2)$$

where $\gamma_j(i)$ is estimated by

$$\gamma_j(i) = \frac{c_j^{(i-1)} f(w_i|w_{i-2}, w_{i-1}; \mathcal{M}_j)}{\sum_k c_k^{(i-1)} f(w_i|w_{i-2}, w_{i-1}; \mathcal{M}_k)} \quad (3)$$

with appropriate terminating condition. Note that a posterior mode may be used instead by combining some prior function at Equation (2).

2.2. Experiments for Domain Models

First, a single trigram based LM was derived from complete **generation set**. This LM is referred to as a "full LM". The perplexity was 186.9 for texts in **evaluation set** as shown in Table 2. This gives the baseline for the rest of experiments.

Next, following Clarkson et. al. [2], 3309 texts in **generation set** were partitioned into 10 domains using hand-labeled information embedded in each text; 1 domain for whole spoken texts and 9 domains (from *imaginative* to *leisure*) for written texts. A trigram based LM was created for each of 10 domains. This LM is referred to as a "domain LM". Table 2 also shows results for two types of mixture LMs; one by 10-domain LMs together with "full LM", another without. Initially, mixing factors $c_j^{(0)}$ were set proportional to the total number of trigrams for each component LM. When computing the perplexity, domain information from **evaluation set** was not used as it was assumed to be complete novel data from which no manually tagged information was available. In comparison to the baseline, improvement by mixture LMs is clearly observed. Specifically, perplexity reduction was significant when using the mixture with "full LM".

3. MODELING THE DOCUMENT SPACE

Latent semantic analysis (LSA) is a modern IR technique that is based on the singular value decomposition (SVD) of very large sparse term (word) by document matrix [3, 4]. Each column of such a matrix describes a document, with the entries being some measure corresponding to each vocabulary word in that document. The eigenvectors corresponding to the k largest eigenvalues are then used to define k-dimensional term and document spaces, where k is typically of the order of 100. Put simply, the approach effectively models the co-occurrence of vocabulary words 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 indicates that points in the derived subspace are more reliable indicators of meaning than individual words or terms [3, 8]. Furthermore, assuming that a document is a (linear) combination of words, it is possible to project any document (with tens of thousands vocabulary words) down to a few hundred dimensional vector, regardless of whether it is included in the original matrix. In this paper, this reduced dimensional space is referred to as a *document space*. One major advantage of this approach is that a lower dimensional document subspace is automatically inferred using the SVD.

3.1. Term by Document Matrix for LSA Models

A method to generate the term by document matrix is one focal point of the LSA approach because it affects the notion of semantics expressed in the space. For example, the unigram relative frequencies might be used for the column (i.e., document vector) entries of such matrix. As the total word counts often vary in orders of magnitude between documents, the unigram probabilities can be used instead if one wants to avoid the possible effect of the document sizes.

When characterizing each document by the occurrence of each word, it would be useful if uniqueness of the word in the whole corpus could be considered. Such measure often used in IR area is the "inverse document factor". It calculates $\frac{p_j(w)}{p(w)}$ where $p_j(w)$ and $p(w)$ are the unigram probabilities of word w in document j and in whole corpus, respectively. This measure enhances the unigram probabilities of the document which are not very common in the whole document set. In IR work, this matrix is weighted by terms designed to improve the retrieval performance [8, 4]. This may be an area for further investigation for language modeling work.

3.2. Singular Value Decomposition (SVD)

The principal computational burden of this approach lies in the SVD of the term by document matrix. It is not unreasonable to expect this matrix to have dimensions of at least 20000×20000; however such matrices are sparse (1–2% of the elements are non-zero) and it is possible to perform such computations on a modern workstation [9]. First, a $m \times n$ matrix A (whose rank is r) can be decomposed as

$$A = U \Sigma V^T \quad (4)$$

where "^{T}" implies a transpose. Σ is an $r \times r$ matrix whose diagonal elements correspond to singular values, or the non-negative square roots of r non-zero eigenvalues for AA^T. Also U and V are $m \times r$ and $n \times r$ matrices whose columns are often referred to as term and document singular vectors. They define the orthonormal eigenvectors associated with the r eigenvalues of AA^T and $A^T A$, respectively [3, 4].

The singular vectors corresponding to the k ($k \leq r$) largest singular values are then used to define k-dimensional document space. Using these vectors, $m \times k$ and $n \times k$ matrices U_k and V_k may be redefined along with $k \times k$ singular value matrix Σ_k. It is then known that $A_k = U_k \Sigma_k V_k^T$ is the closest matrix (in a least square sense) of rank k to the original matrix A [4]. As a consequence, given an m-dimensional vector q for a document, it is warranted that k-dimensional projection $\hat{q}_k$

	spoken	*imag-inative*	*natural science*	*applied science*	*social science*
0	15	507	345	2 045	1 148
1	6	40	15	275	1 096
2	3 340	195	2	29	54
3	244	298	181	975	897
4	64	987	20	624	154
5	22	56	33	965	98
6	12	157	201	1 139	284
7	26	173	24	448	88
8	153	2 762	2	78	123
9	6	106	37	1 816	224
total	3 888	5 281	860	8 394	4 166

Table 3. This table shows how many documents from each domain were classified to one of 10 classes. Document space here was created first by inverse document factors, then clustered by the cosine angle criterion. Out of 10 domains, *spoken*, *imaginative*, *natural science*, *applied science*, and *social science* domains were extracted.

computed by

$$\hat{q}_k = q^T U_k \Sigma_k^{-1} \tag{5}$$

lies in the closest k-dimensional document space with respect to the original m-dimensional space. In the experiments described in Section 4, $m = 19989$ and $k = 200$ were used, effectively achieving an order of 100 reduction in the work space.

The k-dimensional projection $\hat{q}_k$ represents principal components that characterize "semantic" information of the document. Thus, corpus documents can be classified according to their projections using, say, *k-means* clustering algorithm together with some metric. Section 4 shows results with two different metrics; one with the Euclidean norm, $||\boldsymbol{a} - \boldsymbol{b}||$, and another with the cosine angle, $\cos\phi = \dfrac{\boldsymbol{a}\cdot\boldsymbol{b}}{||\boldsymbol{a}||\cdot||\boldsymbol{b}||}$, between two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

4. EXPERIMENTS FOR LSA MODELS

Because each text in the BNC contains tens to hundreds of thousands words, they were subdivided using the context cue information[1] so that varying style of text can be tracked. 3309 texts in **generation set** were divided into 67680 units. These units are referred to as "documents". An average document contains slightly more than a thousand words, however some documents may have orders of magnitude larger or smaller number of words.

In the experiments described below, 40000 documents were randomly chosen and 19989 × 40000 term by document matrices were generated using the unigram relative frequencies and inverse document factors. They were very sparse; approximately 1.6 % of the elements were not zero. The SVD was applied (using a publicly available package, "SVDPACKC" [9]), computing the top 200 singular values and their corresponding singular vectors. Using Equation (5), 67680 documents were projected on to 200-dimensional document space. Then, documents were clustered using *k-means* algorithm with the Euclidean norm or the cosine angle metric. The resulting document clusters are referred to as "classes" in order to differentiate from hand-labeled domains.

4.1. Association between Domains and Classes

Although similarity does not necessarily imply the superiority in language modeling task, it is still of interest to compare automatically generated classes against hand-labeled domains. Table 3 shows how many documents from each domain were classified to one of 10 classes.

[1]The context cue "<*div*>" was used for dividing the text. Further information is found in [6].

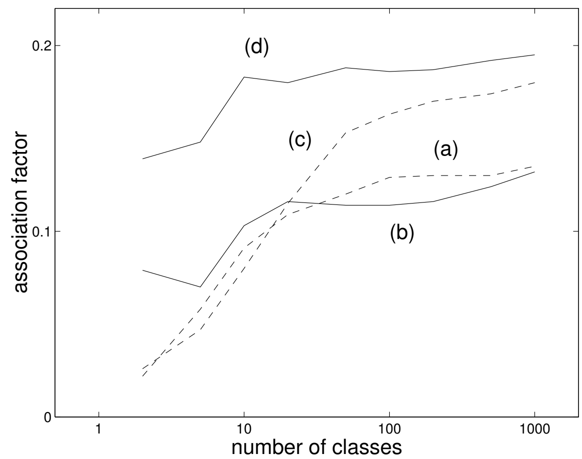

Figure 1. This figure shows strength of association between domains and classes for the following document space models; (a) unigram frequency/Euclidean norm, (b) unigram frequency/cosine angle, (c) inverse doc factor/Euclidean norm, and (d) inverse doc factor/cosine angle.

Document space here was created first by inverse document factors, then clustered by the cosine angle criterion. For example, it is observed from the table that most of documents in *spoken* domain were identified as **class 2**, while more than half of those in *imaginative* domain have fallen to **class 8**. Also, distribution of documents in *natural science* looks quite similar to that in *applied science* but rather different from that in *social science*.

Although interesting, it is difficult to compare one document space to the other just by observing the frequency table. In order to quantify the strength of association between domains and classes, an entropy based factor was computed for each approach [10]. Suppose p_{ij} is the probability that some document in class i is hand-labeled as domain j. Then, joint entropy is $H(i,j) = -\sum_i \sum_j p_{ij} \log p_{ij}$. Class entropy $H(i)$ and domain entropy $H(j)$ are computed similarly. Using entropies, a quantified measure of association is obtained by

$$U(i,j) = 2 \times \frac{H(i) + H(j) - H(i,j)}{H(i) + H(j)} \tag{6}$$

If there is no association between domain and class, then $U(i,j) = 0$ because $H(i) + H(j) = H(i,j)$. On the other hand, if domains and classes are completely dependent, then $U(i,j) = 1$ because $H(i) = H(j) = H(i,j)$.

Figure 1 shows the strength of association between domains and classes for each document space model. In comparison, it seems the association was stronger for the document space model constructed from inverse document factor matrix and cosine angle clustering.

4.2. Perplexity for LSA Class Models

Table 4 shows perplexities for mixture LMs derived from each document space model. Each mixture LM consists of 10 component LMs. The class information for **evaluation set** was not used when computing the perplexity. This is the same condition as for domain LMs in Section 2 (except that mixture LMs were constructed from automatically derived classes instead of hand-labeled domains). It is observed that a mixture LM from the document space with the inverse document factors and cosine angle criterion seems to work better than the other LMs. It is interesting to note that this document space have shown stronger association to the hand-labeled domain in comparison to the others (see Figure 1).

term/doc matrix	clustering criterion	#classes	perplexity
unigram frequency	Euclidean	10	182.1
	cosine angle	10	183.5
inverse doc factor	Euclidean	10	184.2
	cosine angle	10	176.8

Table 4. This table shows perplexities for mixture LMs derived from each document space model, corresponding to plot (a) to (d) of Figure 1. Each mixture LM consists of 10 component LMs. In comparison, the perplexity of the "full LM" alone was 186.9 (see Table 2).

matrix/ clustering	#classes	without "full LM"	with "full LM"
inverse doc/ cosine angle	10	176.8	171.9
	20	179.4	169.8

Table 5. Document space here was created first by inverse document factors, then divided to 10/20 clusters using the cosine angle criterion. Perplexities are computed for mixture LMs of 10/20 component LMs and with/without "full LM".

This document space (i.e., created by inverse document factor matrix/cosine angle clustering) was further tested. Table 5 shows perplexities for mixture LMs of 10/20 component LMs and with/without "full LM". Not surprisingly, mixture LM with "full LM" worked better as it could provide more smoothed space. Because this approach computes perplexities and the mixing factors from all models, increasing the number of mixture components results in a less manageable system.

4.3. Using Class Information for Evaluation Set

So far, document space information was used only when constructing the class LMs from **generation set**. This experiment makes use of semantic notion for **evaluation set**. First, documents (10459 in total) in **evaluation set** were projected down to the document space. For each projection, the closest class LM was selected. Instead of unknowingly computing a mixture of all models, this approach evaluated a mixture of "full LM" and the selected class LM.

Table 6 shows perplexities for such mixtures when document space was divided to 10 to 1000 clusters using the cosine angle criterion. The table suggests that a mixture LM performed better when the document space was divided to around 100, however it might be strongly affected by the condition of the experiment. This approach took advantage of automatic nature of the LSA modeling. A single "full LM" was tuned to the document space with slight increase of computational cost. It achieved perplexity level very close to a mixture of all models although computation was an order of magnitude faster (because it was a mixture of just two LMs).

5. SUMMARY

In this paper, LSA based approach for modeling the document space has been described. Using the LSA derived semantic information, mixture LMs were constructed in an unsupervised manner. Manually tagged corpus (BNC) was used in the experiments and the comparison was made between manual and automatic approaches. The results does suggest that the approach was able to detect (at least a part of) semantic information from the document. In general, mixture LMs performed as much as 10 % lower perplexity than a conventional single LM. In particular, using semantic information, a single LM was tuned to the document space with slight increase of computational cost.

term/doc matrix	clustering criterion	#classes	perplexity
inverse doc factor	cosine angle	10	175.2
		20	172.9
		50	172.2
		100	171.9
		200	172.4
		500	174.7
		1000	175.4

Table 6. Document space here was created first by inverse document factors, then divided to 10 to 1000 clusters using the cosine angle criterion. When computing the perplexities, a class LM closest to the document space projection of **evaluation set** was blended with the "full LM".

The LSA approach is corpus based, statistical, and automatic; thus particularly well suited for a conventional state-of-art large vocabulary speech recognition application. Ongoing work involves the evaluation of speech recognition performance using lattice rescoring on the ABBOT continuous speech recognition system [11].

REFERENCES

[1] R. Iyer and M. Ostendorf. Modeling long distance dependence in language: Topic mixture vs. dynamic cache models. In *Proceedings of ICSLP-96*, Philadelphia, PA, October 1996.

[2] P. R. Clarkson and A. J. Robinson. Language model adaptation using mixtures and an exponentially decaying cache. In *Proceedings of ICASSP-97*, volume 2, pages 799–802, Munich, April 1997. IEEE.

[3] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

[4] Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995.

[5] Jerome R. Bellegarda, John W. Butzberger, Yen-Lu Chow, Noah B. Coccaro, and Devang Naik. A novel word clustering algorithm based on latent semantic analysis. In *Proceedings of ICASSP-96*, volume 1, pages 172–175, Atlanta, May 1996. IEEE.

[6] Lou Burnard (Editor). *Users Reference Guide, British National Corpus Version 1.0.* Oxford University Computing Service, May 1995.

[7] F. Jelinek and R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Pattern Recognition in Practice: Proceedings of an International Workshop held in Amsterdam*, pages 381–397, May 1980.

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

[9] Michael Berry, Theresa Do, Gavin O'Brien, Vijay Krishna, and Sowmini Varadhan. SVDPACKC (version 1.0) user's guide. Technical Report CS-93-194, University of Tennessee, Department of Computer Science, 1993.

[10] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, Cambridge, UK, 1988.

[11] Tony Robinson, Mike Hochberg, and Steve Renals. The use of recurrent networks in continuous speech recognition. In *Automatic Speech and Speaker Recognition – Advanced Topics*, pages 233–258. Kluwer, 1996.