Word Clustering using Long Distance Bigram Language Models

Nikoletta Bassiou and Constantine Kotropoulos, *Senior Member, IEEE*

***Abstract*—Two novel word clustering techniques employing language models of long distance bigrams are proposed. The first technique is built on a hierarchical clustering algorithm and minimizes the sum of Mahalanobis distances of all words after cluster merger from the centroid of the resulting class. The second technique resorts to the probabilistic latent semantic analysis (PLSA). Interpolated versions of the long distance bigrams are considered as well. Experiments conducted on the English Gigaword corpus (Second Edition) validate that 1) long distance bigrams create more meaningful clusters including less outliers than the baseline bigrams; 2) interpolated long distance bigrams outperform long distance bigrams in the same respect; 3) long distance bigrams perform better than bigrams incorporating trigger-pairs for various histories; 4) PLSA-based interpolated long distance bigrams yield the most efficient language model in the context of this study. To assess objectively the quality of the formed clusters, cluster validity indices as well as mutual information-based measures have been estimated and box plots for the intra-cluster dispersion are demonstrated.**

***Index Terms*—word clustering, language modeling, distance bigrams, probabilistic latent semantic analysis, cluster validity, trigger pairs, cluster dispersion.**

I. Introduction

Word clustering has been one of the most challenging tasks in the natural language processing. It exploits well founded unsupervised clustering techniques, such as hierarchical, partitional, fuzzy, or neural network-based ones, to reveal a simple, but yet valid organization of the data [1]. More precisely, word clustering assigns words to groups according to their contextual, syntactic, or semantic similarity. Such context information can be captured by language models that estimate the conditional probability of a word given its context. The definition of context results in various language model types, such as n-gram, long distance n-gram, skipping, caching, n-gram classes, and sentence mixture models [2]. These language models can also be extended by using word triggers [3] or can be combined with each other by interpolation.

In this paper, we investigate the ability of long distance bigram models to capture long distance word dependencies with a small number of free parameters in order to derive meaningful word clusters. More precisely, two word clustering techniques are proposed that take into consideration long distance bigram probabilities at varying distances within a context as well as interpolated long distance bigram probabilities. The first technique extends a hierarchical clustering algorithm that is based on the minimization of the sum of Mahalanobis distances of all words after cluster merger from the centroid of the resulting class [4]. The second technique is based on Probabilistic Latent Semantic Analysis (PLSA) that associates unobserved class variables with words in a way that increases the data likelihood [5]. Both techniques assume the same predetermined number of word clusters.

Although, word clustering techniques have already been reported in the literature, our contribution is in their enhancement with long distance bigram statistics. The first proposed technique, which was presented in our previous work [6] aiming at comparing two different interpolation methods for long distance bigrams, it is now elaborated further on a larger corpus where both long distance bigrams and their interpolated variants are considered. In addition, in this paper, PLSA, which has primarily been applied for document classification, is modified for word clustering by employing either long distance bigrams or their interpolated variants.

An assessment of word clustering techniques is undertaken with respect to the language model, that has been incorporated. More specifically, the word clusters derived in experiments conducted on a subset of the English Gigaword corpus (Second Edition) [7] are examined. Cluster validity indices, such as the Jaccard, the adjusted Rand, and the Fowlkes-Mallows, between the resulting clusters and random ones [8] confirm that the proposed techniques produce non-random clusters. The normalized variation of information between the clusterings derived by the clustering techniques under

The authors are with the Department of Informatics, Aristotle University of Thessaloniki, Box 451 Thessaloniki 541 24, GREECE.

Corresponding author: N. Bassiou, e-mail: nbassiou@aiia.csd.auth.gr.

Manuscript received July, 2008; revised November, 2008;

study, when the baseline bigram and the long distance bigrams are employed, also show that the long distance bigram models differ more from the bigram model than their interpolated variants do. Moreover, the intra-cluster dispersion demonstrates that the PLSA-based word clustering techniques, which exploit long distance bigram models, provide more compact clusters and the use of interpolated long distance bigram models within both clustering techniques eliminates the outliers, which are observed when long distance bigrams are used. The aforementioned facts are further validated by observing the words assigned to sample clusters, such as the words appearing in the context of week days and car races, for various distances. The just mentioned sample clusters reveal the ability of PLSA-based word clustering, when interpolated long distance bigrams are used, to generate meaningful clusters similar to those formed when a bigram model is interpolated with trigger word pairs for various histories. However, the clustering with trigger pairs assigns similar words into more than one clusters and it suffers from the additional complex task of appropriate trigger pair selection

The outline of the paper is as follows. In Section II, word clustering methods are reviewed. Statistical language modeling concepts, such as bigrams, long distance bigrams, and interpolation techniques, together with absolute discounting, that is used to alleviate the zero frequency problem, is addressed in Section III. The proposed word clustering methods that incorporate long distance bigrams with or without interpolation are described next, in Section IV. Experimental results are illustrated in Section V and conclusions are drawn in Section VI.

II. Related Work

Natural language processing applications, such as automatic thesauri creation, word sense disambiguation, language modeling, and text classification, have been greatly favored from word clustering methods. Although, word clustering methods may vary with respect to data description, the similarity measure considered, or the nature of the algorithms (i.e., hierarchical, partitional, fuzzy, and so on), they succeed to enhance the aforementioned applications by lowering the data dimensions and providing data representations, which reveal the similarity between words.

In automatic thesauri creation, a sigmoid Bayesian network is built from local term dependencies to model the similarity distribution between terms even with low frequencies [9]. In a similar approach, a hierarchical lexical clustering neural network algorithm automatically generates a thesaurus and succeeds to create clusters of synonyms, but at the expense of high computational complexity, since it resorts to best-matching finding [10]. In another approach, syntactic information is used to discover word similarities by means of a weighted Jaccard measure [11]. Both lexical and syntactic information are used to derive an information-theoretic similarity matrix in [12].

In automatic word sense disambiguation, several methods exploit co-occurrence statistics, which potentially reflect the semantic meaning of words. For example, an iterative bootstrapping procedure computes the word-sense probability distributions for word collocations [13]. By using heuristics, the algorithm finds the word senses starting from seed classifications that are subsequently refined to reveal all word senses. In another approach, the maximization of an information-theoretic measure of similarity between words with similar local contexts, which are defined in terms of syntactic dependencies, determines the assignment of words with similar senses to classes [14]. Lexical features representing co-occurring words in varying sized contexts are also exploited by an ensemble of Naive Bayesian classifiers to classify words according to their sense [15]. The clustering by committees (CBC) algorithm discovers a set of tight clusters, called committees, that are well scattered in the similarity space, and assigns words to the most similar clusters [16]. The words are represented by a feature vector whose elements are the mutual information between the word and its context. The overlapping features of a word assigned to a cluster are removed enabling thus less frequent senses of words to be discovered, while preventing sense duplication. Word clusters are also derived from an efficient word clustering algorithm that estimates the joint probabilities of word pairs by means of the Minimum Description Length principle [17]. The resulting word classes are then combined with a hand-made thesaurus for syntactic disambiguation. Another method augments the training data with sentences extracted from WordNet [18] and conducts semantic classification by means of a hierarchical multiclass perceptron [19].

In language modeling applications, word clustering based on word co-occurrences is exploited in order to alleviate the data sparseness problem and to improve model quality. For example, a hierarchical word clustering algorithm based on bigram and trigram statistics constructs word classes by iteratively merging classes so that the reduction in the average mutual information is minimal. The derived classes are then used to construct class-based n-grams [20]. A similar bottom-up algorithm builds multi-level class based interpolated language models by using the average mutual information criterion and representing the words as structural tags [21]. An exchange algorithm for bigram and trigram word clustering similar to k-means produces word classes by minimizing the perplexity of the class model [22]. The number of the resulting word classes, however, is constrained by the time complexity as well as the memory requirements of the algorithm. Recently, a more efficient exchange algorithm is proposed that generates word clusters from large vocabularies by greedily maximizing the log likelihood of a two-sided class bigram or trigram model on the training data [23]. Randomizing the aforementioned exchange algorithm by considering various initializations or data subsets yields random class-based models that can be later combined in an interpolated model to improve perplexity and reduce the word-error rate compared to the n-gram and the class-based language models [24]. A top-down asymmetric clustering algorithm that splits the clusters with respect to the maximal entropy decrease may generate different clusters for predicted words (predictive clusters) and conditional words (conditional clusters) [25]. Word clusters can be obtained by applying Latent Semantic Analysis (LSA) to bigram or trigram statistics as well. The word clusters could be then used to build a multi-span language model either in a maximum entropy framework or by straightforward interpolation, that captures both local and global constraints [26].

In text classification, the word clusters contribute to feature reduction. In contrast to the divisive entropy-based method, that produces soft noun clusters according to their conditional verb distributions [27], a hard agglomerative clustering algorithm, that is based on the minimization of the average Kullback-Leibler divergence between each class distribution over a given word and the mean distribution [28], produces word clusters that are later used to classify documents using a Naive-Bayes classifier. An extension of this work that guarantees a good text classification performance even for a small training corpus, generates word clusters by taking into consideration the global information over all word clusters [29]. A more generalized word clustering technique applies the Information Bottleneck method to find word clusters that preserve the information about document categories [30]. Similarly to aforementioned agglomerative methods, an information theoretic framework is exploited by a divisive algorithm to monotonically decrease an objective function for clustering, that is based on the Jensen-Shannon divergence [31].

III. Language Modeling

A. The n-Gram Language Model

The n-gram model assumes a Markov process of order $n-1$ and estimates the probability of a word given only the most recent $n-1$ preceding words [2]. The probability of a sequence of words $\mathbf{W} =< w_1\, w_2\, \ldots\, w_M >$, $P(\mathbf{W})$, is thus expressed as

$$\begin{aligned} P(\mathbf{W}) \;\; &= \;\; P(w_1) \prod_{i=2}^{n-1} P(w_i|w_1, \ldots, w_{i-1}) \\ & \qquad \prod_{i=n}^{M} P(w_i|w_{i-1}, \ldots, w_{i-n+1}). \end{aligned} \tag{1}$$

Usually bigram or trigram language models are employed to facilitate the computations in (1) for large n.

For a limited size training corpus with a vocabulary V of size $Q = |V|$, the probabilities in (1) are estimated by means of relative frequencies, i.e.

$$P(w_i|w_{i-1}, \ldots, w_{i-n+1}) \simeq \frac{N(w_{i-n+1}\, \ldots\, w_{i-1}\, w_i)}{N(w_{i-n+1}\, \ldots\, w_{i-1})} \tag{2}$$

where the notation $N(.)$ stands for the number of occurrences of the word sequence inside parentheses in the training corpus.

B. The Long Distance Bigram Model

To compensate for the loss of syntactic and semantic information from the more distant words, when bigrams or trigrams are used, and at the same time to reduce the number of free parameters, skipping models were proposed in [3], [32], [33]. The long distance bigrams predict word w_i based on word w_{i-d} [32]. Taking into consideration this notion, we introduce the notation $D(w_i, w_j) = d$, where $d \in \mathbb{Z}$ denotes the distance between words w_i and w_j. Accordingly, $i = j - d$. It is clear that for $D(w_i, w_j) = 1$, we get the baseline bigram. Assuming that the number of occurrences of the bigram $< w_i\, w_j >$ is $N_d(w_i\, w_j) \triangleq N\left(w_i\, w_j | D(w_i, w_j) = d\right)$, the probability of the distance bigram can be obtained as

$$P_d(w_j|w_i) = P\left(w_j|w_i, D(w_i, w_j) = d\right) \simeq \frac{N_d(w_i\, w_j)}{N(w_i)}. \tag{3}$$

Accordingly, the probability $P(\mathbf{W})$ of the resulting language model for a given word sequence is expressed by

$$P_d(\mathbf{W}) = \prod_{j=1}^{d} P(w_j) \prod_{j=d+1}^{M} P_d(w_j|w_i). \tag{4}$$

C. Interpolating Long Distance Bigram Language Models

To enhance the efficiency of the long distance bigram model, the probability of long distance bigrams in H different distances can be estimated [6]. Let

$$P^{(H)}(w_j|w_i) = P\left(w_j|w_i, D(w_i, w_j) \leq H\right). \quad (5)$$

It can be easily shown that the number of occurrences of the bigram $< w_i\, w_j >$ in H different distances is given by

$$\begin{aligned} N^{(H)}(w_i\, w_j) &= N\left(w_i\, w_j|D(w_i, w_j) \leq H\right) \\ &= \sum_{d=1}^{H} N_d(w_i\, w_j). \quad (6) \end{aligned}$$

Taking into consideration that the maximum value $N_d(w_i\, w_j)$ is $N(w_i)$ for $d \in [1, H]$, the probability (5) can be estimated by means of relative frequencies as follows:

$$P^{(H)}(w_j|w_i) \simeq \frac{N^{(H)}(w_i\, w_j)}{HN(w_i)} = \frac{\sum_{d=1}^{H} N_d(w_i\, w_j)}{H\, N(w_i)}. \quad (7)$$

The decomposition (7) can be interpreted as a weighted sum/average of the component probabilities $P_d(w_j|w_i)$ for $d \in [1, H]$. Introducing weights λ_d for each component probability, such that $\sum_{d=1}^{H} \lambda_d = 1$ and $0 \leq \lambda_d \leq 1$, (7) can be generalized to:

$$P^{(H)}(w_j|w_i) \simeq \frac{\sum_{d=1}^{H} \lambda_d N_d(w_i\, w_j)}{N(w_i)} = \sum_{d=1}^{H} \lambda_d P_d(w_j|w_i). \quad (8)$$

The substitution of (8) into (1) yields

$$\begin{aligned} &P^{(H)}(\mathbf{W}) = P^{(H)}(w_1\, w_2 \ldots w_M) = \\ &\prod_{j=1}^{d} \left(\sum_{d=1}^{H} \lambda_d\, P(w_j) \right) \prod_{j=d+1}^{M} \left(\sum_{d=1}^{H} \lambda_d\, P_d(w_j|w_i) \right) \quad (9) \end{aligned}$$

where λ_d are estimated on held out data by means of the Expectation Maximization (EM) algorithm [34].

D. Absolute Discounting

The probability estimates in (1), (4), and (9) become zero for unseen events (i.e., bigrams, long distance bigrams). Data sparseness necessitates addressing the problem of unseen events. This can be achieved by means of discounting, a smoothing technique which discounts the relative frequencies of seen events and redistributes the gained probability mass over the unseen events [35].

To adapt discounting for long distance bigrams, the *"count-counts"* $n_{r,d}(h)$ denoting the number of distinct words w that were seen following history h at distance d exactly r times and $n_{r,d}$ denoting the total number of distinct joint events that occurred exactly r times at distance d should be defined:

$$\begin{aligned} n_{r,d}(h) &= \sum_{w:N_d(h\, w)=r} 1 \quad (10) \\ n_{r,d} &= \sum_{hw:N_d(h\, w)=r} 1 = \sum_{h} n_{r,d}(h). \quad (11) \end{aligned}$$

The events with counts $r = 0, 1$ are characterized as unseen and singleton (hapax legomena) ones, respectively.

Absolute discounting estimates $P_d(w|h)$ as follows:

$$\begin{aligned} P_d(w|h) = \max &\left(0, \frac{N_d(h\, w) - b_d}{N(h)}\right) + \\ b_d\, &\frac{Q - n_{0,d}(h)}{N(h)}\, \frac{\beta_d(w|\bar{h})}{\sum_{w':N_d(\bar{h}\, w')=0} \beta_d(w'|\bar{h})} \end{aligned} \quad (12)$$

where b_d is a count-independent non-integer offset and $\beta_d(w|\bar{h})$ is a generalized distribution of a word given a generalized history $\bar{h}$ at distance d. The count-independent non-integer offset, b_d, which is estimated by means of the data log-likelihood for the leave-one-out model, is approximated by

$$b_d \simeq \frac{n_{1,d}}{n_{1,d} + 2n_{2,d}} \quad (13)$$

and the generalized distribution $\beta_d(w|\bar{h})$ is estimated by

$$\begin{aligned} \beta_d(w|\bar{h}) &= \frac{N_d(\bar{h}\, w)}{\sum_{w'} N_d(\bar{h}\, w')} = \frac{\sum_{z \in V} N_d(z\, w)}{\sum_{w'} \sum_{z \in V} N_d(z\, w')} \\ &= \frac{N(w)}{\sum_{w'} N(w')} = P(w). \end{aligned} \quad (14)$$

As a result, (3) using (12), after having substituted (14), is rewritten as

$$\begin{aligned} P_d(w_j|w_i) = \max &\left(0, \frac{N_d(w_i\, w_j) - b_d}{N(w_i)}\right) + \\ b_d\, &\frac{Q - n_{0,d}(w_i)}{N(w_i)}\, P(w_j). \end{aligned} \quad (15)$$

IV. Word Clustering

The efficiency of the just defined language models, which capture long-term word dependencies with a low number of parameters, is to be tested for word clustering. More precisely, two word clustering techniques, which exploit the language models presented in Section III, are proposed. The first technique is inspired by the hierarchical word clustering algorithm presented in [20], but it uses long distance bigrams instead of bigrams and trigrams, and it is based on the minimization of the sum of Mahalanobis distances of all words of the cluster merger from the centroid of the resulting class instead of the minimal reduction in the average mutual information. The second technique extends the idea of using bigram and trigram statistics in LSA [26] in using long distance bigram statistics in PLSA, which is the probabilistic version of LSA.

A. Word Clustering based on the Minimization of the sum of Mahalanobis Distances

Word clustering, as presented in [4], is based on the observation that the variance of the class-based conditional probabilities is smaller than the variance of the word-based conditional probabilities. In this paper, we extend this idea by exploiting the long distance bigram model and the interpolated long distance bigram model for the conditional probability estimates between words at different distances. The proposed algorithms are outlined next.

1) Algorithm based on Long Distance Bigrams (Method I-a): Given a vocabulary V of size $Q = |V|$, let us assume that R non-overlapping classes of functionally equivalent words C_k, $k = 1, \dots, R$, are found

$$V = \bigcup_{k=1}^{R} C_k, \qquad C_s \cap C_k = \emptyset \text{ for } s \neq k \tag{16}$$

such that

$$\forall w_i \in C_s, \forall w_j \in C_k, \quad P_d(w_j|w_i) = P_d(C_k|C_s) \tag{17}$$

where $P_d(w_j|w_i)$ and $P_d(C_k|C_s)$ are the transition probabilities between the words in the bigram $< w_i w_j >$ that lie at distance d as well as their corresponding classes C_s and C_k, respectively. Due to the constraints imposed by the limited size of any corpus, estimates of the transition probabilities $P_d(w_j|w_i)$ are employed, as defined in (3) and (12), with absolute discounting for the unseen events. Hence, every word w of the vocabulary is characterized by the transition probability estimates $P_d(w_j|w)$ from this word to all vocabulary words $w_j, j = 1, 2, \dots, Q$.

The probability of occurrence of the Q generalized long distance bigrams in $N_d(w) = N(w)$ repeated Bernoulli trials is given by [36]:

$$P\left(N_d(w\,w_1), \dots, N_d(w\,w_Q)\right) = [N(w)]! \prod_{j=1}^{Q} \frac{[P_d(w_j|w)]^{N_d(w\,w_j)}}{[N_d(w\,w_j)]!} \tag{18}$$

where $P\left(N_d(w\,w_1), \dots, N_d(w\,w_Q)\right)$ denotes the probability of having $N_d(w\,w_j)$, $j = 1, 2, \dots, Q$ occurrences of the corresponding distance bigrams in the training set. In (18), since $N(w)$ is sufficiently large and $N_d(w\,w_j)$ is in the $\sqrt{N(w)}$ neighborhood of $N(w)\ P_d(w_j|w)$, according to De Moivre-Laplace theorem, the right-hand side is approximated by a Q-dimensional Gaussian probability density function (pdf) [36]:

$$P\left(P_d(w_1|w), \dots, P_d(w_Q|w)\right) = N(\boldsymbol{\mu}_w, \mathbf{U}_w) \tag{19}$$

with mean vector and covariance matrix given by

$$\boldsymbol{\mu}_w = \left(P_d(w_1|w), \dots, P_d(w_Q|w)\right)^T \tag{20}$$

$$\mathbf{U}_w = \frac{1}{N(w)} \operatorname{diag}\left[P_d(w_1|w), \dots, P_d(w_Q|w)\right] \tag{21}$$

where ^{T} denotes vector/matrix transposition and $\operatorname{diag}[\ ]$ denotes a diagonal matrix having the arguments inside brackets as its elements on the main diagonal. In Appendix A, a detailed derivation of (19)-(21) is given.

The algorithm starts with every word w_i in the vocabulary forming an individual class on its own and proceeds with merging the two classes for which the sum of Mahalanobis distances of all the words of the cluster merger from the centroid of the class is minimum. Let $\mathbf{v}_i$ be the vector of transition probabilities whose kth component is the transition probability from word w_i to word w_k $P_d(w_k|w_i)$, i.e.

$$\mathbf{v}_i = \left(P_d(w_1|w_i), \dots, P_d(w_Q|w_i)\right)^T \tag{22}$$

the probability of the hypothesis that classes C_p and C_q are merged to form a single class is given by

$$P(C_p \cup C_q) = \prod_{\forall i \to w_i \in C_p \cup C_q} \frac{1}{(2\pi)^{Q/2} (\det(\mathbf{U}_{pq}))^{1/2}} \exp\Big\{-\frac{1}{2}(\mathbf{v}_i - \boldsymbol{\mu}_{pq})^T [\mathbf{U}_{pq}]^{-1} (\mathbf{v}_i - \boldsymbol{\mu}_{pq})\Big\} \tag{23}$$

where $\boldsymbol{\mu}_{pq}$ and $\mathbf{U}_{pq}$ are the mean vector and covariance matrix of the class formed by merging classes C_p and C_q, respectively [4]. In (23), $\det(\cdot)$ denotes the determinant of a matrix. Classes to be merged should maximize (23). By taking the logarithm of the right-hand side in (23) and dropping the normalization term, we should equivalently choose C_{p^*} and C_{q^*} such that

$$(p^*, q^*) = \arg\min_{(p,q)} \sum_{\forall i \to w_i \in C_p \cup C_q} (\mathbf{v}_i - \boldsymbol{\mu}_{pq})^T [\mathbf{U}_{pq}]^{-1} (\mathbf{v}_i - \boldsymbol{\mu}_{pq}). \tag{24}$$

$\boldsymbol{\mu}_{pq}$ can be estimated as

$$\boldsymbol{\mu}_{pq} = \frac{1}{|C_p| + |C_q|} \left(\sum_{\forall i \to w_i \in C_p \cup C_q} \mathbf{v}_i \right) \tag{25}$$

where $|C_p|$ and $|C_q|$ are the number of words that belong to the corresponding classes. In par to (21), the covariance matrix is diagonal with kkth element given by

$$[\mathbf{U}_{pq}]_{kk} = \frac{1}{[\sum_{\forall i \rightarrow w_i \in C_p \cup C_q} N(w_i)]^2} \sum_{\forall i \rightarrow w_i \in C_p \cup C_q} N^2(w_i) [U_{w_i}]_{kk}. \tag{26}$$

The derivation of (25) and (26) is analyzed in Appendix B.

Summarizing, the clustering algorithm works as follows:

- Step 1: Each word of the vocabulary comprises a class on its own. The algorithm starts with Q classes.
- Step 2: The two classes that minimize (24) are merged.
- Step 3: If the number of remaining classes equals a predetermined number of classes R, the algorithm stops. Otherwise, a new iteration starts at Step 2.

There are approximately $(Q-k)^2/2$ class pairs that have to be examined for merging in each iteration k of the just described algorithm. In order to avoid the exhaustive search, the words of the vocabulary are sorted in decreasing order of frequency and the first $R+1$ words are assigned to $R+1$ distinct classes. At each iteration, the class pair for which the sum of Mahalanobis distances of all the words of the cluster merger from the centroid of the class is minimum is found and the merger is performed yielding R classes. The insertion of the next word of the vocabulary in a distinct class results again in $R+1$ classes. So at iteration k, the $(R+k)$-th most probable word of the vocabulary is assigned in a distinct class and the algorithm proceeds until no vocabulary words are left. After $Q-R$ steps, the words of the vocabulary have been assigned to R classes. Therefore, at iteration k, the number of class candidates to be tested for merger is $((R+1)-k)^2/2 < (Q-k)^2/2$ [20].

2) Algorithm based on Interpolated Long Distance Bigrams (Method I-b): Let us assume again that R non-overlapping classes can be created, so that:

$$\forall w_i \in C_s, \forall w_j \in C_k, \quad P^{(H)}(w_j|w_i) = P^{(H)}(C_k|C_s) \tag{27}$$

where $P^{(H)}(w_j|w_i)$ and $P^{(H)}(C_k|C_s)$ are the interpolated transition probabilities between the words $< w_i\, w_j >$ that lie at distances $d = 1, \ldots, H$ as well as between their corresponding classes C_s and C_k. $P^{(H)}(w_j|w_i)$ can be estimated by (8).

Following similar lines to IV-A1, the probability of the occurrence of the Q generalized long distance bigrams, in $\sum_{d=1}^{H} \lambda_d N(w) = N(w)$ repeated Bernoulli trials can be expressed as [36]:

$$P\left(\sum_{d=1}^{H} \lambda_d N_d(w\, w_1), \ldots, \sum_{d=1}^{H} \lambda_d N_d(w\, w_Q)\right) = [N(w)]! \prod_{j=1}^{Q} \frac{\left[\sum_{d=1}^{H} \lambda_d P_d(w_j|w)\right]^{\sum_{d=1}^{H} \lambda_d N_d(w\, w_j)}}{\left[\sum_{d=1}^{H} \lambda_d N_d(w\, w_j)\right]!} \tag{28}$$

where $P\left(\sum_{d=1}^{H} \lambda_d N_d(w\, w_1), \ldots, \sum_{d=1}^{H} \lambda N_d(w\, w_Q)\right)$ denotes the probability of having $\sum_{d=1}^{H} \lambda_d N_d(w w_j)$, $j = 1, 2, \ldots, Q$ occurrences of the corresponding distance bigrams in the training set. Applying the De Moivre-Laplace theorem [36], we get

$$P\left(\sum_{d=1}^{H} \lambda_d P_d(w_1|w), \ldots, \sum_{d=1}^{H} \lambda_d P_d(w_Q|w)\right) = N(\boldsymbol{\mu}_w^{(H)}, \mathbf{U}_w^{(H)}) \tag{29}$$

with mean vector given by

$$\boldsymbol{\mu}_w^{(H)} = \left(\sum_{d=1}^{H} \lambda_d P_d(w_1|w), \ldots, \sum_{d=1}^{H} \lambda_d P_d(w_Q|w)\right)^T \tag{30}$$

and diagonal covariance matrix $\mathbf{U}_w^{(H)}$, i.e.

$$\mathbf{U}_w^{(H)} = \frac{1}{N(w)} \cdot \operatorname{diag}\left[\sum_{d=1}^{H} \lambda_d P_d(w_1|w), \ldots, \sum_{d=1}^{H} \lambda_d P_d(w_Q|w)\right]. \tag{31}$$

The derivation of (29) - (31) is similar to that of (19) - (21).

The clustering algorithm starts with every word w_i in the vocabulary forming an individual class and proceeds by merging the two classes that satisfy

$$(p^*, q^*) = \arg\min_{(p,q)} \sum_{\forall i \rightarrow w_i \in C_p \cup C_q} (\mathbf{v}_i - \boldsymbol{\mu}_{pq}^{(H)})^T \, [\mathbf{U}_{pq}^{(H)}]^{-1} \, (\mathbf{v}_i - \boldsymbol{\mu}_{pq}^{(H)}) \tag{32}$$

where $\mathbf{v}_i$ is the vector of estimated transition probabilities from this word to any other vocabulary word given by

$$\mathbf{v}_i = \left(\sum_{d=1}^{H} \lambda_d P_d(w_1|w_i), \ldots, \sum_{d=1}^{H} \lambda_d P_d(w_Q|w_i)\right)^T \tag{33}$$

and an estimate of $\boldsymbol{\mu}_{pq}^{H}$ can be obtained by (25). Again we have employed a diagonal covariance matrix with kkth element given by

$$[\mathbf{U}_{pq}]_{kk}^{(H)} = \frac{1}{[\sum_{\forall i \rightarrow w_i \in C_p \cup C_q} N(w_i)]^2} \sum_{\forall i \rightarrow w_i \in C_p \cup C_q} N^2(w_i)\, [U_{w_i}]_{kk}^{(H)}. \tag{34}$$

The derivation of (34) follows similar lines to that of (26).

B. Word Clustering based on PLSA

PLSA performs a probabilistic mixture decomposition by defining a generative latent data model, the so called aspect model, which associates an unobserved class variable $z_k \in Z = \{z_1, z_2, \ldots, z_R\}$ with each observation. Here, the observation is simply the occurrence of a word/term $w_j \in V = \{w_1, w_2, \ldots, w_Q\}$ in a text/document $t_i \in T = \{t_1, t_2, \ldots, t_M\}$, while the unobserved class variable z_k models the topic a text was generated from. Following the basic assumption of the aspect model, all the observation pairs (t_i, w_j) are assumed to be independent and identically distributed, and conditionally independent given the respective latent class z_k. Accordingly, the joint distribution of a word w_j in a text t_i generated by latent topic z_k is given by

$$P(t_i, w_j, z_k) = P(t_i) P(z_k|t_i) P(w_j|z_k). \tag{35}$$

Summing over all possible realizations of z_k, the joint distribution of the observed data is obtained, i.e.

$$P(t_i, w_j) = \sum_{k=1}^{R} P(t_i, w_j, z_k) = P(t_i) \underbrace{\sum_{k=1}^{R} P(z_k|t_i) P(w_j|z_k)}_{P(w_j|t_i)}. \tag{36}$$

As can be seen from (35), the text-specific word distributions $P(w_j|t_i)$ are obtained by a convex combination of the R aspects/factors $P(w_j|z_k)$. This implies that the texts are not exclusively assigned to clusters, but they are characterized by a specific mixture of factors with weights $P(z_k|t_i)$.

Representing each text t_i as a sequence of words $< v_1\, v_2\, \ldots\, v_{Q_i} >$, where Q_i is the number of words in text t_i, $P(t_i, w_j)$ can be decomposed as follows

$$P(t_i, w_j) = P(v_1\, v_2\, \ldots\, v_{Q_i}, w_j) = P(v_1|v_2\, \ldots\, v_{Q_i}, w_j)\, P(v_2|v_3\, \ldots\, v_{Q_i}, w_j)\, \ldots\, P(v_{Q_i}|w_j)\, P(w_j). \tag{37}$$

1) Algorithm based on Long Distance Bigrams (Method II-a): Taking into consideration the long distance bigram model described in Section III-B, (37) can be expressed in terms of long distance bigram probabilities at a certain distance d. That is,

$$\begin{aligned} P(t_i, w_j) &\simeq P(w_j) \prod_{l=1}^{Q_i} P_d(v_l|w_j) \\ &= P(w_j) \prod_{w_l \in t_i} P_d(w_l|w_j). \end{aligned} \tag{38}$$

Following similar lines to (36), $P_d(w_l|w_j)$ can be obtained by summing over all possible realizations of z_k, i.e.

$$P_d(w_l|w_j) = \sum_{k=1}^{R} P_d(z_k|w_j)\, P_d(w_l|z_k). \tag{39}$$

The learning problem is formulated as maximization of the log-likelihood function with respect to the entailed probabilities. The log-likelihood function $\mathcal{L}$ with the help of (39) can be expressed as

$$\mathcal{L} = \sum_{j=1}^{Q} N_d(w_j)\, \log P(w_j) + \sum_{j=1}^{Q} \sum_{l=1}^{Q} N_d(w_j\, w_l) \log \left[\sum_{k=1}^{R} P_d(z_k|w_j) P_d(w_l|z_k) \right] \tag{40}$$

where $N_d(w_j\, w_l)$ is the number of word co-occurrences $< w_j\, w_l >$ that lie at distance d and $\sum_{j=1}^{Q} N_d(w_j) = \sum_{j=1}^{Q} \sum_{l=1}^{Q} N_d(w_j\, w_l)$. The maximization of the log-likelihood $\mathcal{L}$ can be achieved by applying the EM algorithm, which alternates between two steps [37]:

1) Expectation (E)-step, where posterior probabilities are computed for the latent variables based on the current estimates of the parameters

$$\hat{P}_d(z_k|w_j, w_l) = \frac{P_d(w_l|z_k) P_d(z_k|w_j)}{\sum_{k'=1}^{R} P_d(w_l|z_{k'}) P_d(z_{k'}|w_j)}. \tag{41}$$

2) Maximization (M)-step, which involves maximization of the expected log-likelihood depending on the posterior probabilities computed in the previous E-step [5], which in our case takes the following form:

$$P(w_l|z_k) = \frac{\sum_{j=1}^{Q} N_d(w_j\, w_l) \hat{P}_d(z_k|w_j, w_l)}{\sum_{l'=1}^{Q} \sum_{j=1}^{Q} N_d(w_j\, w_{l'}) \hat{P}_d(z_k|w_j, w_{l'})} \tag{42}$$

$$P(z_k|w_j) = \frac{\sum_{l=1}^{Q} N_d(w_j\, w_l) \hat{P}_d(z_k|w_j, w_l)}{\sum_{k'=1}^{R} \sum_{l=1}^{Q} N_d(w_j\, w_l) \hat{P}_d(z_{k'}|w_j, w_l)}. \tag{43}$$

By alternating (41) with (42)-(43), a procedure that converges to a local maximum of the log-likelihood results. Each word w_j is assigned to one only cluster C_{s_j} such that

$$s_j = \arg\max_k P(z_k|w_j)\ ,\ \ j = 1, 2, \ldots, Q. \tag{44}$$

2) Algorithm based on Interpolated Long Distance Bigrams (Method II-b): Taking into consideration the interpolated long distance bigram model described in Section III-C, (38) can be expressed in terms of interpolated long distance bigram probabilities as follows:

$$\begin{aligned} P(t_i, w_j) &\simeq P(w_j) \prod_{l=1}^{Q_i} P^{(H)}(v_l|w_j) \\ &= P(w_j) \prod_{w_l \in t_i} P^{(H)}(w_l|w_j). \end{aligned} \tag{45}$$

Following similar lines to (37), $P^{(H)}(w_l|w_j)$ can be obtained by summing over all possible realizations of z_k, i.e.

$$P^{(H)}(w_l|w_j) = \sum_{k=1}^{R} \left[\sum_{d=1}^{H} \lambda_d P(w_l|z_k) \right] P(z_k|w_j). \tag{46}$$

The log-likelihood function to be maximized with respect to the probabilities $P_d(w_l|z_k)$ and $P^{(H)}(z_k|w_j)$ is given by:

$$\begin{aligned} \mathcal{L} = &\sum_{j=1}^{Q} N(w_j) \log P(w_j) + \\ &\sum_{j=1}^{Q} \sum_{l=1}^{Q} N(w_j\, w_l) \log \left[\sum_{k=1}^{R} \sum_{d=1}^{H} \lambda_d P_d(w_l|z_k) P(z_k|w_j) \right] \end{aligned} \tag{47}$$

where $\sum_{j=1}^{Q} N(w_j) = \sum_{j=1}^{Q} \sum_{l=1}^{Q} N(w_j\, w_l)$.

It can be shown that the PLSA algorithm for (47) alternates between two steps:

1) E-step:

$$\hat{P}_d(z_k|w_j, w_l) = \frac{P_d(w_l|z_k) P(z_k|w_j)}{\sum_{k'=1}^{R} P_d(w_l|z_{k'}) P(z_{k'}|w_j)}. \tag{48}$$

2) M-step:

$$P_d(w_l|z_k) = \frac{\sum_{j=1}^{Q} N(w_j\, w_l) \hat{P}_d(z_k|w_j, w_l)}{\sum_{l'=1}^{Q} \sum_{j=1}^{Q} N_d(w_{l'}\, w_j) \hat{P}_d(z_k|w_j, w_{l'})} \tag{49}$$

$$P(z_k|w_j) = \frac{\sum_{l=1}^{Q} \sum_{d=1}^{H} \lambda_d N(w_j\, w_l) \hat{P}_d(z_k|w_j, w_l)}{\sum_{k'=1}^{R} \sum_{l=1}^{Q} \sum_{d=1}^{H} \lambda_d N(w_j\, w_l) \hat{P}_d(z_{k'}|w_j, w_l)}. \tag{50}$$

By alternating (48) with (49)-(50), a procedure that converges to a local maximum of the log-likelihood results.

V. Experimental Results

The word clustering algorithms developed in Section IV were implemented and tested for various long distance bigram models starting from the baseline bigram corresponding to distance $d = 1$ and proceeding to long distance bigram models at distance d=2–6 and d=9 (Methods I-a and II-a). Interpolated long distance bigram models at distance H=2–6 (Methods I-b and II-b) have also been investigated. In addition, language models combining the baseline bigram with trigger pairs [3] at various histories (d=2–6 and d=9) were also incorporated in the best performing PLSA-based clustering algorithm for comparison purposes.

The clustering algorithms are compared with respect to the quality of the word clusters they produce. For this purpose, various cluster validity indices, such as the Jaccard, the Adjusted Rand, and the Fowlkes-Mallows [8] between the resulting clusters and random clusters are estimated first to demonstrate that the resulting clusters are non-random. Next, the intra-cluster dispersion for all clusterings has been studied to objectively assess the clusters produced. Sample clusters, that are derived by the clustering methods under study, are presented to validate the aforementioned objective measurements.

A. Dataset and Parameter Setting

The experiments were conducted on a subset of the English Gigaword Second Edition Corpus[1] [7] that includes 17288 newswire texts of story type produced by the English Service of Agence France-Presse (afp_eng) during 1994 – 1995. The texts have been pre-processed in order to remove tags, non-English words, numbers or symbols with no meaning. The word documents were also stemmed using the Porter stemmer [38]. The resulting vocabulary contains 41744 words with frequencies ranging from 162289 to 1. Due to memory limitations, however, a vocabulary cut-off was applied by discarding words with frequency of appearance less than 50. The cut-off vocabulary size is 4720 words. In Table I, the vocabulary size and the number of long distance bigrams at various distances are summarized.

To derive the conditional probabilities needed in Methods I and II, first the frequencies of the distance bigrams at distances d=1–6 and $d = 9$ were estimated and

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2005T12

TABLE I
VOCABULARY SIZE AND NUMBER OF LONG DISTANCE BIGRAMS.

Size of		*Number of Long Distance Bigrams*						
Original Vocabulary	*Vocabulary after cut-off*	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$	$d = 6$	$d = 9$
41744	4582	616854	783196	839605	868590	884075	893322	903506

next absolute discounting was applied. Furthermore, the interpolation weights needed for the interpolation of the long distance bigrams at distances H=2–6 were estimated by a two-way cross validation on held-out data by means of the Expectation-Maximization algorithm.

For all clustering algorithms, the predefined number R of the resulting classes was set to 300. In addition, for the PLSA-based clustering algorithms (Methods II-a and II-b), the convergence criterion for the EM algorithm requests the relative log-likelihood change between two successive EM-steps to be less than 10^{-4}, a condition that has reached after approximately 100 iterations. It is also worth mentioning that the PLSA-based algorithms have executed 10 times for each language model in order to guarantee that the results are not affected by the EM convergence to local minima.

To select among the possible Q^2 long distance word-pairs (trigger pairs) at the same distances as the ones employed in the long distance bigrams (d=2-6 and $d = 9$), a probability threshold $p_0 = 1.5/Q$ was set. A trigger interaction between two words was thus allowed only if the corresponding word pairs probability in the bigram model was below p_0. The conditional probabilities of the extended model (i.e. bigram with trigger pairs) was then estimated by a back-off technique as described in [39].

B. Word Clustering Results and Their Assessment

Among the validity indices, the ones that are based on external criteria, i.e. the Jaccard, the Adjusted Rand and Fowlkes-Mallows were selected [8]. These indices are based on counting the word pairs on which two clusterings C and C' agree or disagree. More precisely, let us defined the following parameters:

a: number of word pairs that are found in the same cluster in both partitions;

b: number of word pairs that are found in the same cluster in C, but in different clusters in C';

c: number of word pairs that are found in different clusters in C, but in the same cluster in C';

d: number of word pairs that are found in different clusters for both partitions.

The maximum number of all pairs is $M = a+b+c+d = Q(Q-1)/2$. The indices used are defined in Table II [8]. These indices admit values between 0 and 1. The higher value is admitted by an index, the stronger the similarity between the two clusterings is. The adjusted rand index exhibits greater sensitivity than the rand index due to the wider range of values it can take. This is why it is usually preferred from rand index [40].

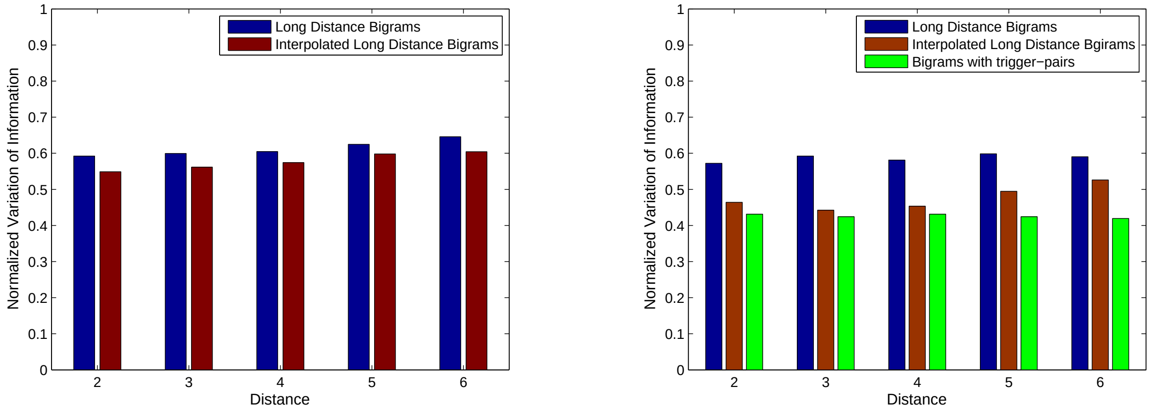
TABLE II
EXTERNAL INDICES.

Jaccard	*Adjusted Rand*	*Fowlkes-Mallows*
$J = \frac{a}{a+b+c}$	$\frac{rand\ index - expected\ index}{maximum\ index - expected\ index}$	$FM = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$

TABLE III
COMPARISON OF GENERATED CLUSTERINGS BY THE PROPOSED METHODS AGAINST RANDOMLY GENERATED ONES USING EXTERNAL INDICES.

Clustering Method	*Long Distance Bigram Model*	*Jaccard*	*Adjusted Rand*	*Fowlkes-Mallows*
Method I-a	$d = 1$ (Classic Bigram)	0.0026	0.0717	0.0053
	$d = 2$	0.0023	0.0712	0.0046
	$d = 3$	0.0025	0.0717	0.0051
	$d = 4$	0.0025	0.0715	0.0050
	$d = 5$	0.0023	0.0712	0.0046
	$d = 6$	0.0021	0.0708	0.0042
	$d = 9$	0.0022	0.0709	0.0044
Method I-b	Interpolated $H = 2$	0.0023	0.0710	0.0048
	Interpolated $H = 3$	0.0022	0.0710	0.0044
	Interpolated $H = 4$	0.0024	0.0714	0.0048
	Interpolated $H = 5$	0.0023	0.0713	0.0047
	Interpolated $H = 6$	0.0024	0.0715	0.0049
Method II-a	$d = 1$ (Classic Bigram)	0.0022	0.0530	0.0048
	$d = 2$	0.0024	0.0617	0.0050
	$d = 3$	0.0017	0.0668	0.0036
	$d = 4$	0.0025	0.0700	0.0051
	$d = 5$	0.0022	0.0695	0.0045
	$d = 6$	0.0024	0.0723	0.0048
	$d = 9$	0.0024	0.0708	0.0047
Method II-b	Interpolated $H = 2$	0.0024	0.0574	0.0050
	Interpolated $H = 3$	0.0024	0.0570	0.0050
	Interpolated $H = 4$	0.0020	0.0570	0.0050
	Interpolated $H = 5$	0.0021	0.0650	0.0044
	Interpolated $H = 6$	0.0023	0.0684	0.0048

In our experiments, the values of these indices were estimated for the resulting clusterings and randomly generated clusterings (Table III). The Monte Carlo technique has been used to estimate the probability density function of each index under the null hypothesis that the data are randomly distributed [41]. A significance level of $\rho = 0.05$ has been set. More precisely, the presented clustering algorithms were applied for 100 randomly generated data sets, and the aforementioned cluster validity indices were estimated. The values of the indices for the randomly generated data sets were compared to the values of indices corresponding to the real data. Considering the significance lever $\rho = 0.05$, the null



(a) (b)

Fig. 1. Normalized variation of information between the clusterings derived by: (a) Methods I-a with the baseline bigram and the long distance bigrams and Method I-b, when the baseline bigram and interpolated long distance bigrams are employed; (b) Method II-a with the baseline bigram and the long distance bigrams, Method II-b with the baseline bigram and interpolated long-distance bigrams, as well as Method II-a employing either the baseline bigram or trigger-pair bigrams.

hypothesis was rejected, since $(1-\rho)100 = 95$ values of these cluster validity indices are smaller than the values of the respective cluster validity indices corresponding to the real data. Some indicative values of Jaccard, adjusted Rand and Fowlkes-Mallows for the clusterings corresponding to the randomly generated data sets are 0.0012, 0.0364 and 0.0030 respectively. As can be seen from Table III, the values of all indices are very close to 0 verifying that the clusters generated by the proposed clustering methods are not random.

In addition to the cluster validity indices, the variation of information $VI(C, C')$ is also estimated as a figure of merit for clustering assessment [42]. Given a clustering C containing R clusters, $C = \{C_1, \ldots, C_R\}$, the entropy of the clustering is given by

$$H(C) = -\sum_{k=1}^{R} P(k) \log P(k). \tag{51}$$

$P(k)$ is the probability of a word belonging to cluster C_k in clustering C:

$$P(k) = \frac{n_k}{\sum_{k=1}^{R} n_k} \tag{52}$$

where n_k is the number of words assigned to cluster C_k. Defining the average mutual information between two clusterings C and C', given the probabilities $P(k)$, $k = 1, \ldots, R$ for clustering C, $P'(k')$, $k' = 1, \ldots, R'$ for clustering C', and $P(k, k')$ for their intersection $C \cap C'$ is defined as [43]:

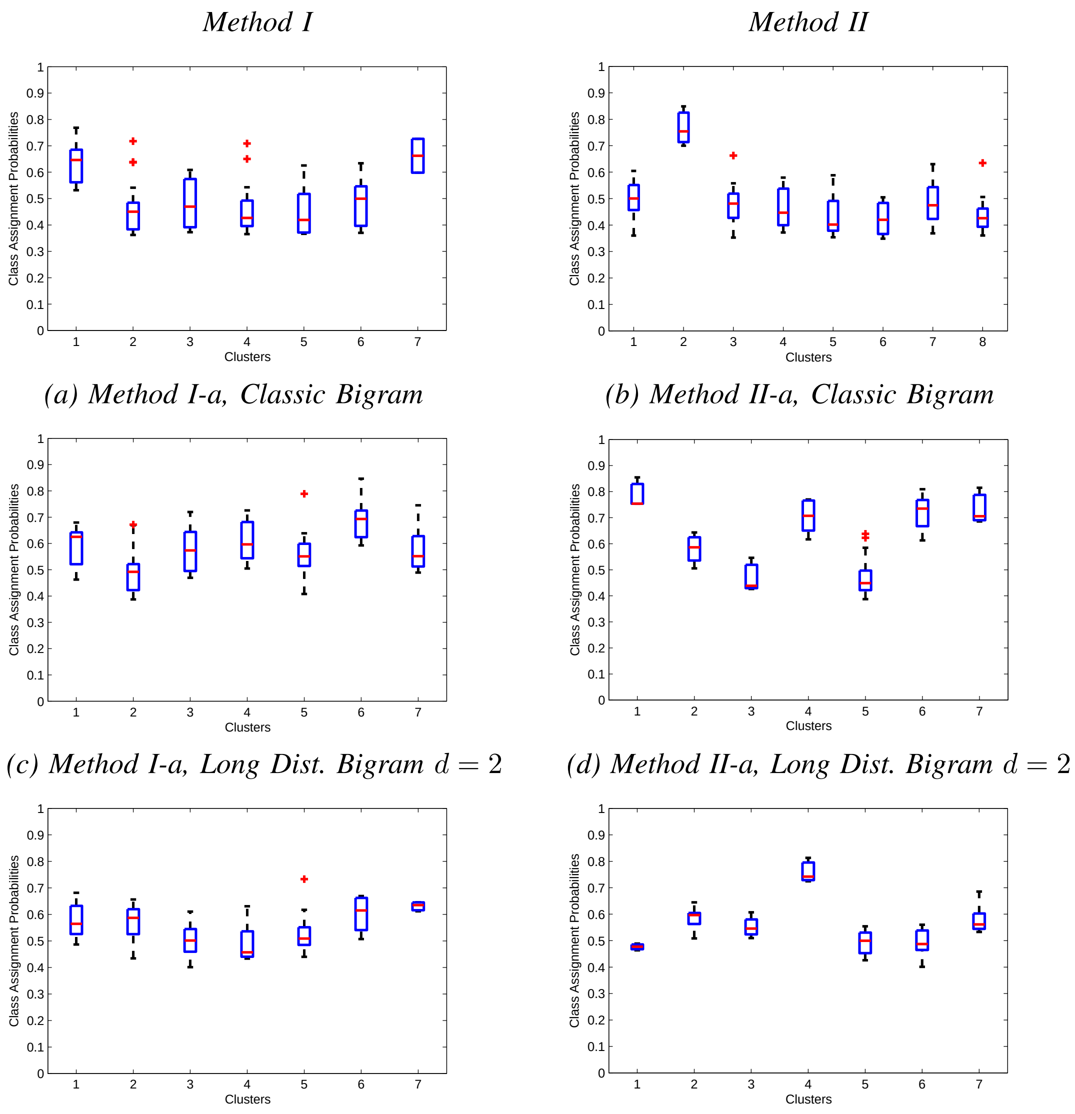
$$\overline{M}(C, C') = \sum_{k=1}^{R} \sum_{k'=1}^{R'} P(k, k') \log \frac{P(k, k')}{P(k)P'(k')} \tag{53}$$

the variation of information $VI(C, C')$, as proposed in [42], is estimated as another figure of merit for the assessment of clusterings:

$$VI(C, C') = [H(C) - \overline{M}(C, C')] + [H(C') - \overline{M}(C, C')]. \tag{54}$$

The first term in (54) measures how well clustering C can be predicted from C', while the second one how well clustering C' can be predicted from C. $VI(C, C')$ takes only positive values. If $C = C'$, then $VI(C, C') = 0$. The upper bound of $VI(C, C')$ is $\log Q$. Usually, the variation of information is normalized by dividing it with its upper bound.

Figure 1 plots the normalized variation of information between clusterings derived by the proposed methods that are based on either the minimization of the sum of Mahalanobis distances of all words after cluster merger from the centroid of the resulting class, without (Method I-a) or with interpolation (Method I-b) as well as PLSA without (Method II-a) or with interpolation (Method II-b), when long distance bigrams for several distances are used against the baseline bigram as well as when trigger-pair bigrams are employed against the baseline bigram model in Method II-a. The lower the value of variation of information is, the more similar the clusterings under comparison are. As can be seen in Figure 1(a), the clusterings generated by minimizing the sum of Mahalanobis distances of all words after cluster merger from the centroid of the resulting class applied to interpolated long distance bigrams are more similar to those obtained by the same method, when it is applied to bigrams than when long distance bigrams (without interpolation) are employed in the same context.



(a) Method I-a, Classic Bigram *(b) Method II-a, Classic Bigram*

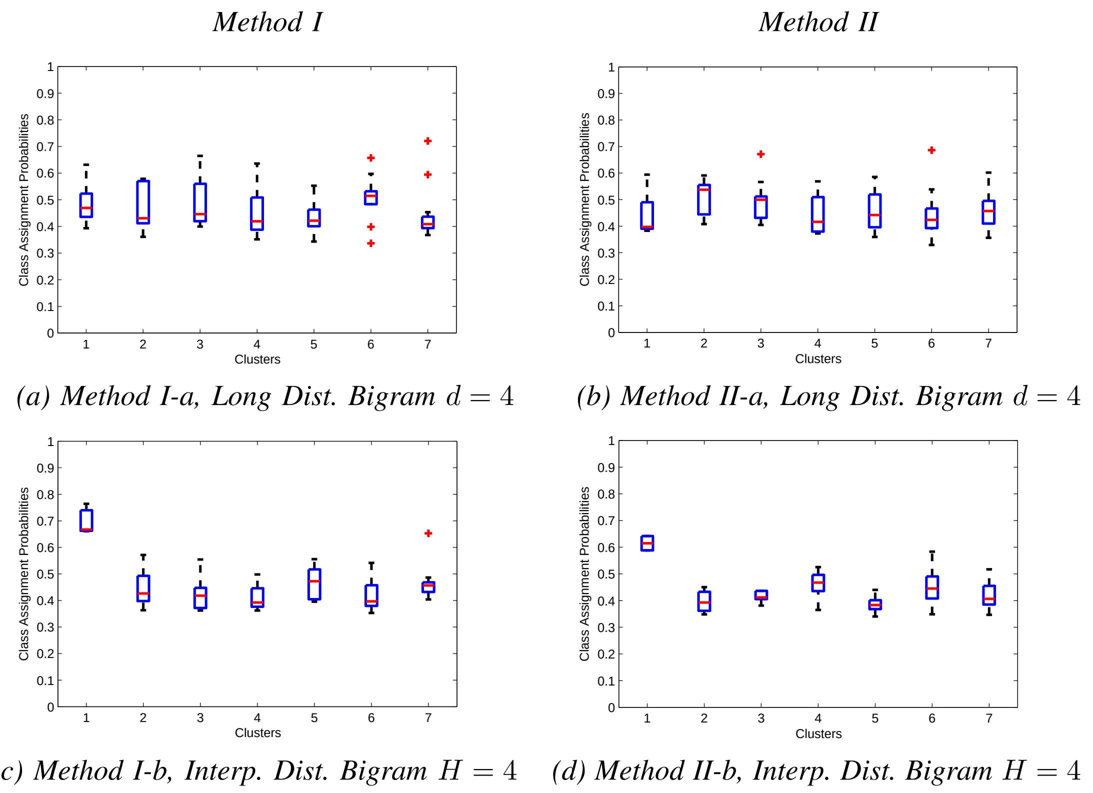
(c) Method I-a, Long Dist. Bigram $d = 2$ *(d) Method II-a, Long Dist. Bigram* $d = 2$

(e) Method I-b, Interp. Dist. Bigram $H = 2$ *(f) Method II-b, Interp. Dist. Bigram* $H = 2$

Fig. 2. Intra-cluster statistics (dispersion, outliers) of cluster assignment probabilities for sample clusters derived by (a) Method I-a and (b) Method II-a applied to the baseline bigrams; (c) Method I-a and (d) Method II-a applied to long distance bigrams at distance $d = 2$; (e) Method I-b and (f) Method II-b applied to interpolated long distance bigrams at distance $H = 2$.

This can be attributed to the fact that interpolated long distance bigrams include the baseline bigram. Similarly, in Figure 1(b) it is seen that trigger-pair bigrams, when they are employed within the PLSA-based clustering, yield a clustering that is more similar to that obtained by PLSA-based clustering applied to the baseline bigrams than when the aforementioned algorithm is applied to long-distance bigrams with our without interpolation. In this case, the trigger-pairs bigram in some respect inherit information from the baseline bigrams model. It is also seen that as distance increases, the clusterings obtained by both methods with the long distance bigrams differ more than those when the baseline bigram is employed in the same context.

A comparison of the clustering methods in terms of their intra-cluster dispersion is illustrated using box plots in Figures 2 and 3. Due to space limitations, results are presented for histories $d = 1, 2$ and $d = 4$. Figure 2 depicts the cluster assignment probability of each word for seven sample clusters derived by the clustering methods under study, when the baseline bigram model, the long distance bigrams at distance $d = 2$ and the interpolated long distance bigrams at distance $H = 2$ are used. Similarly, Figure 3 shows the cluster assignment probability of each word for the same seven sample clusters derived by the clustering methods under study, when the long distance bigrams at distance $d = 4$ and the interpolated long distance bigrams at distance $H = 4$ are used. By comparing Figures 2 (a) and (b) with Figures 2 (c) and



(a) Method I-a, Long Dist. Bigram $d = 4$ *(b) Method II-a, Long Dist. Bigram* $d = 4$

(c) Method I-b, Interp. Dist. Bigram $H = 4$ *(d) Method II-b, Interp. Dist. Bigram* $H = 4$

Fig. 3. Intra-cluster statistics (dispersion, outliers) of cluster assignment probabilities for sample clusters derived by (a) Method I-a and (b) Method II-a applied to long distance bigrams at distance $d = 4$; (c) Method I-b and (d) Method II-b applied to interpolated long distance bigrams at distance $H = 4$.

(d) as well as Figures 3 (a) and (b), it can be seen that intra-cluster dispersion, which represents the cluster compactness, is smaller when long distance bigrams are used instead of the baseline bigrams. Furthermore, it can be verified from Figures 2 (c) and (e) and Figures 2 (d) and (f), that interpolated long distance bigrams when they are employed in the clustering algorithms reduce the outliers observed when long distance bigrams are used. The same observation can be made by the inspection of Figures 3 (a) and (c) as well as Figures 3 (b) and (d). Moreover, the examination of Figures 2 (c)– (f) and Figures 2 (a)– (d) reveals the superiority of the PLSA-based Methods II-a and II-b over the Methods I-a and I-b, since the dispersion of the PLSA-based clusters is smaller than the dispersion of the clusters produced by Methods I-a and I-b.

In Tables IV-VII, sample clusters produced by the methods under study are demonstrated, supporting the aforementioned observations. The clusters contain words representing various aspects of the corpus, such as days of the week and car race. Inspecting the representative clusters, the superiority of Methods I-b and II-b, which employ the interpolated long distance bigram models, against Methods I-a and II-a, which employ only a single long distance bigram model, in producing more meaningful and compact clusters (with less outliers) is validated easily. For the cluster containing the week days, method I-a gathers $4-5$ days out of 7, while method II-b, which employs the interpolated long distance bigram models, collects $5-6$ days. In the same way, the PLSA-based methods II-a and II-b contain respectively $4-7$ and 7 week days within one cluster.

Finally, the PLSA-based method II-a, which has been proved to provide more meaningful clusters than Method I-a, is applied to a bigram model extended with trigger-pairs, that was selected from histories d=2–6 and d=9. The same sample clusters are presented in Table VIII. As can be seen, clustering with trigger pairs produces meaningful clusters as Methods I-b and II-b do. However, the compact clusters, which are obtained by Methods I-b and II-b when applied to long distance bigrams, are now split into more than one clusters with trigger-pairs.

VI. Conclusions

Two techniques for word clustering are developed which employ long distance bigram models with and without interpolation in order to capture long-term word dependencies with a few parameters. The first technique is based on the minimization of the sum of Mahalanobis distances of all words after cluster merger from the

TABLE IV
CLUSTERS PRODUCED BY METHOD I-A.

Long Distance Bigram Model	week days related clusters	race related clusters
$d = 1$	year, years, said, thursday, wednesday, tuesday, monday	win, winning, wins, race, races, racing, premier, draw, norway, romania, bulgaria, austria, barcelona, finland, lap, laps, circuit, circuits, pole, poles, sena, reynolds
$d = 2$	thursday, monday, tuesday, wednesday, flow, flows, flowed, evan, evans	win, winning, wins, race, races, racing, ireland, argentina, brazil, indies, indy, lap, laps, diego, honda, toyota, mph, mile, miles
$d = 3$	thursday, monday, tuesday, wednesday, colleagues, colleague, salary, salaries	lead, leading, race, races, racing, australia, brazil, win, winning, wins, beaten, tennis, prix, shock, shocking, shocked, engine, engineer, engineers, engineering, vehicle, vehicles, yellow, pole, poles
$d = 4$	year , percent, thursday, week , tuesday, wednesday	race, races, racing, television, televised, televisions, series, serie, william, williams, series, opponent, opponents, boat, boats, formula, mansell, mph, fittipaldi, tycoon, tycoons, senna
$d = 5$	friday, wednesday, tuesday, monday, creation, authorisation, authorised	race, races, racing, television, televised, televisions, away, title, titled, titles, william, williams, midfield, midfielder, midfielders, tennis, formula, degree, degrees, compete, competed, competence, competing, competiveness, bruguera, fittipaldi, pete, andretti
$d = 6$	week, weeks, percent, thursday, friday, tuesday, wednesday	race, races, racing, television, televisions, televised, associate, associated, association, associations, associating, short, engine, engineer, engineers, engineering, object, objected, objective, objectives, objections, tell, telling, delors, mike, bonn, fittipaldi, simon, simones
$d = 9$	office, report, reports, reported, reporter, reporters, week, weeks, thursday, friday, tuesday, wednesday	race, races, racing, television, televisions, televised, water, waters, watered, grand, carlo, carlos, mansell, formula, pack, packing, packed, pierre, mph, senna

TABLE V
CLUSTERS PRODUCED BY METHOD I-B.

Interpolated Long Distance Bigram Model	week days related clusters	race related clusters
$H = 2$	thursday, monday, tuesday, wednesday, year, years	ford, capriati, toyota, benneton, car, cars, race, races, racing, mile miles, lap, laps, fittipaldi, circuit, circuits, driver, drivers
$H = 3$	friday, week, weeks, thursday, monday, tuesday, wednesday	car, cars, race, races, racing, position, positions, positive, positively, trial, trials, william, williams, mph, mile, miles, lap, laps, circuit, circuits, fittipaldi, ford, capriati, toyota, benneton, driver, drivers
$H = 4$	year, years, friday, thursday, monday, tuesday, wednesday, saturday	race, races, racing, win, winning, wins, trial, trials, circuit, circuits, sport, sports, sporting, retire, retired, retirement, driver, drivers, engine, engineer, engineers, engineering, prix, mph, brazil, barcelona, senna, mansell
$H = 5$	thursday, monday, tuesday, wednesday, friday, saturday, week, weeks, month, months, elaborating, elaborate	half, race, races, racing, premier, penalty, penalties, century, centuries, mansell, formula, barcelona, runner, runners, ford, fittipaldi, grand, mph, senna, prix
$H = 6$	year, years, week, weeks, month, months, thursday, monday, tuesday, wednesday, elaborate, elaborating	poll, polls, polling, race, races, racing, mph, lose, losing, fifth, william, williams, mansell, formula, seventh, toyota, honda, ninth, pole, poles, driver, drivers, engine, engineer, engineers, engineering, victory, victories, position, positions, positive, positively

TABLE VI
CLUSTERS PRODUCED BY METHOD II-A.

Long Distance Bigram Model	week days related clusters	race related clusters
$d = 1$	thursday, friday, sunday, saturday, week, weeks, tuesday	win, winning, wins, race, races, racing, victory, victories, success, successful, defeat, defeats, defeated, court, interference, sport, sports, sporting, car, cars ,engine, engineer, engineers, engineering, fittipaldi, william, williams
$d = 2$	friday, tuesday, wednesday, monday, week, weeks, sunday, saturday, late, evening	race, races, racing, victory, victories, tour, tours, row, lap, laps, brazilian, austrian, tie, shuttle, shuttled, fittipaldi, mph, season, seasons
$d = 3$	monday, thursday, friday, wednesday, week, weeks, early, holiday, holidays	time, times, win, winning, wins, race, races, racing, series, leadership, sponsor, sponsors, sponsored, brazilian, brazilians, technical, technically, pena, fittipaldi, gift, gifts, gifted, inspired, inspiring, inspiration, nigel
$d = 4$	thursday, wednesday, friday, pope, rumour, rumours, rumoured week, weeks, monday, saturday	series, win, winning, wins, leadership, season, seasonally sponsor, sponsors, sponsored, brazilian, fifth, race, races, racing, protest, protests, protested, protesting, mansell, nigel
$d = 5$	thursday, tuesday, wednesday, monday, source, sources, saturday	car, cars, race, races, racing, drive, drives, driving, william, williams, row, rows, retire, retired, retirement, mansel, indy, indies, formula, prix, lap, circuit, circuits, contender, contenders, fastest, mph, pole, pit, pits, pitting, pitted, andretti, reign, reigned, reigning, tires, tired
$d = 6$	thursday, tuesday, wednesday, saturday, week, weeks, sign, signed, signing, discuss, discussion, discussing, arrive, arrived, arrival, arriving, friday, holiday, holidays, formal, formally	car, cars, race, races, racing, season, seasons, finish, finished, pit, pits, pitting, pitted, mile, miles, persist, persisted, persistence, persistance, persistent, hero, heroes, win, winning, wins, mansell, prix, driver, drivers
$d = 9$	wednesday, saturday, score, scores, scored, scoring, friday, beat, beating, wednesday, draw, drawing	race, races, racing, season, seasons, series, paul, grand, row, rows, speed, speeds, indy, indies, austrian, lap, laps, circuit, circuits, fittipaldi, fastest, mph, pole, poles, polling, polled, andretti, senna, mansell, benetton, nigel

centroid of the resulting class. The second technique resorts to the probabilistic latent semantic analysis (PLSA). The validity assessment of clustering results has demonstrated that both techniques produce more compact word clusters with less outliers when either long distance bigrams or their interpolated versions are employed rather than when the classic bigram model is used. These clusters have also been proven to be better than those produced when trigger pairs from various histories are employed in conjunction with the baseline bigram, since they have less outliers. Moreover, trigger-pairs clusters of similar words are often split into more than one

TABLE VII
CLUSTERS PRODUCED BY METHOD II-B.

Interpolated Long Distance Bigram Model	week days related clusters	race related clusters
$H = 2$	thursday, friday, tuesday, wednesday, monday, sunday, saturday, week, weeks	race, races, racing, season,seasons, position, positions, positive, positively, career, careers, grand, olympic, olympics, indy, indies, formula, lap, laps, fastest, mph, runner, runners, circuit, circuits, fittipaldi, teammate, teammates, pit, pits, pitting, pitted, pole, poles, driver, drivers
$H = 3$	thursday, friday, tuesday, wednesday, monday, sunday, saturday, weeks	race, races, racing, season, seasons, position, positions, positive, positively, grand, memory, memories, memorial, olympic, olympics, indy, indies, formula, lap, laps, tire, tires, tired, william, williams, runner, runners, circuit, circuits, fittipaldi, teammate, teammates, pit, pits, pitting, pitted, pole, poles, driver, drivers
$H = 4$	thursday, friday, tuesday, wednesday, monday, sunday, saturday, week, weeks	race, races, racing, season, seasons, position, positions, positive, positively, grand, memory, memories, memorial, olympic, olympics, indy, indies, formula, lap, laps, wear, wearing, william, williams, runner, runners, circuit, circuits, fittipaldi, teammate, teammates, pit, pits, pitting, pitted, mph, pole, poles, driver, drivers
$H = 5$	thursday, friday, tuesday, wednesday, monday, sunday, saturday	race, races, racing, season, seasons, position, positions, positive, positively, grand, memory, memories, memorial, olympic, olympics, indy, indies, formula, lap, laps, tire, tires, tired, driver, drivers, runner, runners, circuit, circuits, fittipaldi, teammate, teammates, pit, pits, pitting, pitted, mph, engine, engineer, engineers, engineering, senna
$H = 6$	thursday, friday, tuesday, wednesday, monday, sunday, saturday, month, months	win, winning, wins, race, races, racing, game, games, victory, victories, success, successful, succession, defeat, defeats, defeated, formula, triumph, triumphs, triumphed, season, seasons, fastest, indy, indies, mile, miles, lap, laps, pit, pits, pitting, pitted, fittipaldi, senna, car, cars, circuit, circuits, grand, position, positions, positive, positively, prix

TABLE VIII
CLUSTERS PRODUCED BY THE PLSA-BASED METHOD FOR A BIGRAM MODEL EXTENDED WITH TRIGGER-PAIRS SELECTED FROM VARIOUS HISTORIES (THE SYMBOL / INDICATES CLUSTER SPLITTING).

Bigram with trigger-pairs from history	week days related clusters	race related clusters
$d = 2$	friday, tuesday, wednesday, pineau, publicly, monday, overnight, shortly	cars, race, races, racing, link, links, linked, linking, corp, corps, engine, engineer, engineers, vehicle, vehicles, indy, indies, formula, lap, laps, mile, miles, circuit, circuits, driver, drivers, mph, pole, poles, toyota, benneton
$d = 3$	thursday, monday, sunday, saturday, april, week, weeks, wednesday shortly	car, cars, oppose, opposed, opposing, driver, drivers, engine, engineer, engineers, indy, indies, formula, lap, laps, store, stores, stored, storing, pole, poles, andretti, race, races, racing, mansell
$d = 4$	friday, tuesday, monday, early, shortly / thursday, wednesday, sunday, saturday	team, teams, race, races, racing, season, seasons, driver, drivers, practice, practices, practiced, practical, practically, sport, sports, favourite, favourites, favouritism, indy, indies, formula, lap, laps, circuit, circuits, fittipaldi, teammate, teammates, unser, pole, poles, senna, mansell, prix
$d = 5$	thursday, friday, monday, late / tuesday, wednesday, sunday, saturday	position, positions, positive, positively, car, cars, senna, black, blacks, william, williams, mph, race, races, racing, route, routes, routed, win, winning, wins, formula, mind, minds, mindfull, lap, laps, drink, drinks, drinking, pole, poles, fastest, tire, tires, tired
$d = 6$	thursday, friday, tuesday, wednesday, monday	race, races, racing, vehicle, vehicles, contribute, contributed, contribution, teammate, teammates, win, winning, wins, personnel, engine, engineer, engineers, engineering, vehicle, vehicles, adjust, adjusted, adjustment, adjustments, season, seasons, formula, triumph, triumphs, triumphed, mph, function, functions, functioning, pit, pits, pitting, pitted, william, williams
$d = 9$	thursday, friday, tuesday, wednesday, monday, sunday	team, teams, match, matched, matching, race, races, racing, season, seasons, event, events, tournament, tournaments, series, driver, drivers, course, career, retire, retired, retirement, indy, indies, formula, circuit, circuits, mile, miles, fittipaldi, win, winning, wins, position, positions, positive, positively

clusters. Furthermore, it has been demonstrated that both clustering methods form more meaningful clusters when the interpolated long distance bigrams are used, while the PLSA-based technique has been proved to be better than the one based on the minimization of the sum of the Mahalanobis distances of all words after cluster merger from the centroid of the resulting class.

APPENDIX A
DETAILED DERIVATION OF (19) - (21)

Equation (18) is written:

$$P_{N(w)}\left(N_d(w\,w_1),\ldots,N_d(w\,w_Q)\right) = \frac{N(w)!}{[N_d(w\,w_1)]!\cdots![N_d(w\,w_Q)]!}\cdot [P_d(w_1|w)]^{N_d(w\,w_1)}\cdots[P_d(w_Q|w)]^{N_d(w\,w_Q)} \qquad (55)$$

According to [36], the probabilities of Bernoulli trials can be approximated by a Gaussian function:

$$\begin{aligned} P_n(k_1,k_2,\ldots,k_Q) &= \frac{n!}{k_1!k_2!\ldots k_Q!}\cdot p^{k_1}\cdots p^{k_Q} \\ &\simeq \frac{\exp\left[-\frac{1}{2}\left[\frac{(k_1-np_1)^2}{np_1}+\cdots+\frac{(k_Q-np_Q)^2}{np_Q}\right]\right]}{\sqrt{(2\pi n)^{Q-1}\,p_1\,p_2\cdots p_Q}} \end{aligned} \qquad (56)$$

Taking into consideration that

$$\begin{aligned} N(k_1,k_2,\ldots,k_Q) &= N(\boldsymbol{\mu},\mathbf{U}) \\ &= \frac{\exp\left[-\frac{1}{2}(\boldsymbol{k}-\boldsymbol{\mu})^T\mathbf{U}^{-1}(\boldsymbol{k}-\boldsymbol{\mu})\right]}{\sqrt{(2\pi n)^{Q-1}p_1\cdots p_Q}} \end{aligned} \qquad (57)$$

the mean vector and covariance matrix are given by:

$$\boldsymbol{\mu} = n\,(p_1,p_2,\ldots,p_Q)^T \qquad (58)$$

$$\mathbf{U} = n\,\mathrm{diag}\,[p_1,p_2,\ldots,p_Q] \qquad (59)$$

where $\mathrm{diag}[\cdot]$ denotes a diagonal matrix having the indicated arguments as elements on its main diagonal.

The exponent in (56) can be rewritten as

$$\left(\frac{k_1-np_1}{\sqrt{np_1}},\dots,\frac{k_Q-np_Q}{\sqrt{np_Q}}\right)^T \cdot \left(\frac{k_1-np_1}{\sqrt{np_1}},\dots,\frac{k_Q-np_Q}{\sqrt{np_Q}}\right) = ((k_1-np_1),\dots,(k_Q-np_Q))^T \operatorname{diag}\left[\frac{1}{np_1},\dots,\frac{1}{np_Q}\right] (k_1-np_1,\dots,k_Q-np_Q) = \left((\frac{k_1}{n}-p_1),\dots,(\frac{k_Q}{n}-p_Q)\right)^T \left\{\frac{1}{n}\cdot\operatorname{diag}\left[p_1,\dots,p_Q\right]\right\}^{-1}\left(\frac{k_1}{n}-p_1,\dots,\frac{k_Q}{n}-p_Q\right) \tag{60}$$

That is,

$$P_n\left(\frac{k_1}{n},\dots,\frac{k_Q}{n}\right) = N(\frac{1}{n}\boldsymbol{\mu}, \frac{1}{n^2}\mathbf{U}). \tag{61}$$

If relative word frequencies are replaced by probabilities in (61), we arrive at (19) - (21). □

Appendix B Fast Computation of the Mean and Covariance for the Merged Classes

The mean vectors for class C_p and C_q are given respectively by

$$\boldsymbol{\mu}_p = \frac{1}{|C_p|}\sum_{\forall i: w_i\in C_p}\mathbf{v}_i \tag{62}$$

$$\boldsymbol{\mu}_q = \frac{1}{|C_q|}\sum_{\forall j: w_j\in C_q}\mathbf{v}_j \tag{63}$$

where $\mathbf{v}_i$ is described in (22) and $\mathbf{v}_j$ is similarly defined. When C_p and C_q are merged to form a single class C_l, the resulting (gross) mean vector is given by:

$$\boldsymbol{\mu}_l = \boldsymbol{\mu}_{pq} = \frac{1}{|C_l|}\sum_{\forall i: w_i\in C_l}\mathbf{v}_l = \frac{|C_p|\boldsymbol{\mu}_q + |C_q|\boldsymbol{\mu}_p}{|C_p|+|C_q|} \tag{64}$$

According to (21), the kkth element of the covariance matrix is expressed as

$$[U_{w_i}]_{kk} = \frac{1}{N(w_i)}\, P_d(w_k|w_i) \tag{65}$$

for $w_i \in C_p$. Similarly, for $w_j \in C_q$

$$[U_{w_j}]_{kk} = \frac{1}{N(w_j)}\, P_d(w_k|w_j). \tag{66}$$

For $w_l \in C_p \cup C_q$ we have

$$[U_{w_l}]_{kk} = \frac{1}{N(w_l)}\, P_d(w_k|w_l). \tag{67}$$

Moreover,

$$P_d(w_k|w_l) = \frac{N_d(w_i\, w_k) + N_d(w_j\, w_k)}{N(w_i)+N(w_j)} = \frac{1}{N(w_i)+N(w_j)} \left[N(w_i)P_d(w_k|w_i) + N(w_j)P_d(w_k|w_j)\right]. \tag{68}$$

Taking into consideration (65) and (66), (68) is rewritten

$$P_d(w_k|w_l) = \frac{1}{N(w_i)+N(w_j)} \left\{[N(w_i)]^2[U_{w_i}]_{kk} + [N(w_j)]^2[U_{w_j}]_{kk}\right\}. \tag{69}$$

The substitution of (69) into (67) yields

$$[U_{w_l}]_{kk} = \frac{1}{[N(w_i)+N(w_j)]^2} \left\{[N(w_i)]^2\,[U_{w_i}]_{kk} + [N(w_j)]^2\,[U_{w_j}]_{kk}\right\} \tag{70}$$

which can be generalized to

$$[\mathbf{U}_{pq}]_{kk} = \frac{1}{[\sum_{\forall l\to w_l\in C_p\cup C_q} N(w_l)]^2} \sum_{\forall l\to w_l\in C_p\cup C_q} N^2(w_l)\,[U_{w_l}]_{kk}. \;\square \tag{71}$$

References

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, September 1999.

[2] J. T. Goodman, "A bit of progress in language modeling," *Computer Speech and Language*, vol. 15, no. 4, pp. 403–434, October 2001.

[3] R. Rosenfeld, *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*, Ph.D. Thesis, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA, April 1994.

[4] C. Beccetti and L. P. Ricotti, *Speech Recognition Systems: Theory and C++ Implementation*, Wiley, Chichester: England, 2nd edition, 1999.

[5] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1-2, pp. 177–196, 2001.

[6] N. Bassiou and C. Kotropoulos, "Interpolated distanced bigram language models for robust word clustering," in *Proc. Nonlinear Signal and Image Processing*, Sapporo, Japan, May 2005, pp. 12–15.

[7] D. Graff, J. Kong, K. Chen, and K. Maeda, *English Gigaword*, Linguistic Data Consortium, PA, 2nd edition, 2005.

[8] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, N.J., 1988.

[9] Y. C. Park and K. S. Choi, "Automatic thesaurus construction using bayesian networks," *Information Processing Management*, vol. 32, no. 5, pp. 543–553, 1996.

[10] V. J. Hodge and J. Austin, "Hierarchical word clustering - automatic thesaurus generation," *Neurocomputing*, vol. 48, no. 1-4, pp. 819–846, October 2002.

[11] G. Grefenstette, *Explorations in Automatic Thesaurus Discovery*, Kluwer Academic Publishers, Dordrecht: The Netherlands, 1st edition, 1994.

[12] D. Lin and P. Pantel, "Induction of semantic classes from natural language text," in *Proc. 7th ACM Int. Conf. Knowledge Discovery and Data mining*, N.Y., 2001, pp. 317–322, ACM Press.

[13] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. 33rd Annu. Meeting Assoc. for Computational Linguistics*, 1995, pp. 189–196.

[14] D. Lin, "Using syntactic dependency as local context to resolve word sense ambiguity," in *Proc. 35th Annu. Meeting Assoc. for Computational Linguistics*, 1997, pp. 64–71.

[15] T. Pedersen, "A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation," in *Proc. Conf. Applied Natural Language Processing*, Seattle, WA, May 2000, pp. 63–69.

[16] P. Pantel and D. Lin, "Discovering word senses from text," in *Proc. 8th ACM Int. Conf. Knowledge Discovery and Data Mining*, N.Y., 2002, pp. 613–619, ACM Press.

[17] H. Li, "Word clustering and disambiguation based on co-occurence data," *Natural Language Engineering*, vol. 8, no. 1, pp. 25–42, 2002.

[18] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA, 1998.

[19] M. Ciaramita, T. Hofmann, and M. Johnson, "Hierarchical semantic classification: Word sense disambiguation with world knowledge," in *Proc. Int. Conf. Artificial Intelligence*, 2003, pp. 9–15.

[20] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[21] J. McMahon and F. Smith, "Improving statistical language model performance with automatically generated word hierarchies," *Computational Linguistics*, vol. 22, no. 2, pp. 217–247, 1996.

[22] S. Martin, J. Liermann, and H. Ney, "Algorithms for bigram and trigram word clustering," *Speech Communication*, vol. 24, no. 1, pp. 19–37, April 1998.

[23] J. Uszkoreit and T. Brants, "Distributed word clustering for large scale class-based language modeling in machine translation," in *Proc. 46th Annu. Meeting Assoc. for Computational Linguistics: Human Language Technologies*, 2008, pp. 755–762.

[24] A. Emami and F. Jelinek, "Random clusterings for language modeling," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, March 2005, vol. 1, pp. 581–584.

[25] J. Gao, J. T. Goodman, G. Cao, and H. Li, "Exploring asymmetric clustering for statistical language modeling," in *Proc. 40th Annu. Meeting Assoc. for Computational Linguistics*, 2001, pp. 183–190.

[26] J. R. Bellegarda, "A multispan language modeling framework for large vocabulary speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 6, no. 5, pp. 456–467, September 1998.

[27] F. Pereira, N. Tishby, and L. Lee, "Distributional clustering of english words," in *Proc. Annu. Meeting Assoc. for Computational Linguistics*, 1993, pp. 183–190.

[28] L. D. Baker and A. K. McCallum, "Distributional clustering of words for text classification," in *Proc. 21st ACM Int. Conf. Research and Development in Information Retrieval*, W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, Eds., N.Y., 1998, pp. 96–103, ACM Press.

[29] C. Wenliang, C. Xingzhi, W. Huizhen, Z. Jingbo, and Y. Tianshun, "Automatic word clustering for text categorization using global information," in *Information Retrieval Technology*, vol. LNCS 3411, pp. 1–11. Springer, Berlin, 2005.

[30] N. Slonim and N. Tishby, "The power of word clusters for text classification," in *Proc. 23rd European Colloq. Inf. Retrieval Research*, 2001.

[31] I. S. Dhillon, S. Mallela, and R. Kumar, "Enhanced word clustering for hierarchical text classification," in *Proc. 8th ACM Int. Conf. Knowledge Discovery and Data Miniming*, N.Y., 2002, pp. 191–200, ACM Press.

[32] X. Huang, F. Alleva, H. Hon, M. Hwang, K. Lee, and R. Rosenfeld, "The SPHINX-II speech recognition system: An overview," *Computer Speech and Language*, vol. 2, pp. 137–148, 1993.

[33] A. Stolcke, "Entropy-based pruning of backoff language models," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, 1998, pp. 270–274, Morgan Kaufmann Publishers.

[34] F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambridge, MA, 1998.

[35] H. Ney, S. Martin, and F. Wessel, "Statistical language modeling using leaving-one-out," in *Corpus-Based Methods in Language and Speech Processing*, S. Young and G. Bloothooft, Eds., pp. 174–207. Kluwer Academic Publishers, Dordrecht: The Netherlands, 1997.

[36] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw Hill, N.Y., 3rd edition, 1991.

[37] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *Journal Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.

[38] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, July 1980.

[39] C. Tillmann and H. Ney, "Word trigger and the EM algorithm," in *Proc. Workshop Computational Natural Language Learning*, 1997, pp. 117–124.

[40] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, pp. 193–218, 1985.

[41] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107–145, 2001.

[42] M. Meilă, "Comparing clusterings by the variation of information," in *Proc. 16th Annu. Conf. Computational Learning Theory*, 2003, pp. 173–187.

[43] J. R. Deller Jr., J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*, J. Wiley and Sons, N.Y, 2000.