Aggregating Continuous Word Embeddings for Information Retrieval

Stéphane Clinchant
Xerox Research Centre Europe
stephane.clinchant@xrce.xerox.com

Florent Perronnin
Xerox Research Centre Europe
florent.perronnin@xrce.xerox.com

Abstract

While words in documents are generally treated as discrete entities, they can be embedded in a Euclidean space which reflects an *a priori* notion of similarity between them. In such a case, a text document can be viewed as a bag-of-embedded-words (BoEW): a set of real-valued vectors. We propose a novel document representation based on such continuous word embeddings. It consists in non-linearly mapping the word-embeddings in a higher-dimensional space and in aggregating them into a document-level representation. We report retrieval and clustering experiments in the case where the word-embeddings are computed from standard topic models showing significant improvements with respect to the original topic models.

1 Introduction

For many tasks such as information retrieval (IR) or clustering, a text document is represented by a vector, where each dimension corresponds to a given word and where each value encodes the word importance in the document (Salton and McGill, 1983). This Vector Space Model (VSM) or bag-of-words (BoW) representation is at the root of topic models such as Latent Semantic Indexing (LSI) (Deerwester, 1988), Probablistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) or Latent Dirichlet Allocation (LDA) (Blei et al., 2003). All these topic models consist in "projecting" documents on a set of topics generally learned in an unsupervised manner. During the learning stage, as a by-product of the projection of the training documents, one also obtains an embedding of the words in a typically small-dimensional continuous space. The distance between two words in this space translates the measure of similarity between words which is captured by the topic models. For LSI, PLSA or LDA, the implicit measure is the number of co-occurrences in the training corpus.

In this paper, we raise the following question: if we were provided with an embedding of words in a continuous space, how could we best use it in IR/clustering tasks? Especially, could we develop probabilistic models which would be able to benefit from this *a priori* information on the similarity between words? When the words are embedded in a continuous space, one can view a document as a Bag-of-Embedded-Words (BoEW). We therefore draw inspiration from the computer vision community where it is common practice to represent an image as a bag-of-features (BoF) where each real-valued feature describes local properties of the image (such as its color, texture or shape). We model the generative process of embedded words using a mixture model where each mixture component can be loosely thought of as a "topic". To transform the variable-cardinality BoEW into a fixed-length representation which is more amenable to comparison, we make use of the Fisher kernel framework of Jaakkola and Haussler (Jaakkola and Haussler, 1999). We will show that this induces a non-linear mapping of the embedded words in a higher-dimensional space where their contributions are aggregated.

We underline that our contribution is **not** the application of the FK to text analysis (see (Hofmann, 2000) for such an attempt). Knowing that words can be embedded in a continuous space, our main contribution is to show that we can *consequently represent a document as a bag-of-embedded-words*. The FK is just *one possible way* to subsequently transform this bag representation into a fixed-length vector which is more amenable to large-scale processing.

The remainder of the article is organized as fol-

Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality, pages 100–109,
Sofia, Bulgaria, August 9 2013.

lows. In the next section, we review related works. In section 3, we describe the proposed framework based on embedded words, GMM topic models and the Fisher kernel. In section 4, we report and discuss experimental results on clustering and retrieval tasks before concluding in section 5.

2 Related Works

We provide a short review of the literature on those topics which are most related to our work: topic models, word embeddings and bag-of-patches representations in computer vision.

Topic models. Statistical topic models build on the idea of Latent Semantic Indexing (LSI) in a probabilistic way. The PLSA model proposed by Hoffman (Hofmann, 1999) can be thought of as a constrained matrix factorization problem equivalent to NMF (Lee and Seung, 1999; Gaussier and Goutte, 2005). Latent Dirichlet Allocation (LDA) (Blei et al., 2003) , the generative counterpart of PLSA, has played a major role in the development of probabilistic models for textual data. As a result, it has been extended or refined in a countless studies (Griffiths et al., 2004; Eisenstein et al., 2011; Hoffman et al., 2010). Statistical topic models are often evaluated with the perplexity measure on a held-out dataset but it has been shown that perplexity only correlates weakly with human preference (Chang et al., 2009). Moreover, several studies reported that LDA does not generally outperform LSI in IR or sentiment analysis tasks (Wang et al., 2011; Maas et al., 2011).

Nevertheless, LSI has known a resurging interest. Supervised Semantic Indexing (SSI) (Bai et al., 2009) learns low-rank projection matrices on query-document pairs so as to minimize a ranking loss. Similarly, (Wang et al., 2011) studies the influence of ℓ_1 and ℓ_2 regularization on the projection matrices and shows how to distribute the algorithm using Map-Reduce.

Word embeddings. Parrallel to the large development of statistical topic models, there has been an increasing amount of literature on word embeddings where it has been proposed to include higher-level dependencies between words, either syntactic or semantic. We note that topic models such as LSI, PLSA or LDA implicitly perform such an embedding (jointly with the embedding of documents) and that the measure of similarity is the co-occurrence of words in the training corpus.

A seminal work in this field is the one by Collobert and Weston (Collobert and Weston, 2008) where a neural network is trained by stochastic gradient descent in order to minimize a loss function on the observed n-grams. This work has later then been refined in (Bengio et al., 2009). Probabilistic methods have also been proposed to learn language models such as the HLBL embedding (Mnih and Hinton, 2007).

Similarly, (Maas et al., 2011) parametrizes a probabilistic model in order to capture word representations, instead of modeling individually latent topics, which lead to significant improvements over LDA in sentiment analysis. Furthermore, (Dhillon et al., 2011) uses the Canonical Correlation Analysis technique between the left and right context vectors of a word to learn word embeddings. Lastly, (Turian et al., 2010) proposes an empirical comparison of several word embedding techniques in a named entity recognition task and provides an excellent state-of-the-art of word representation. Except (Maas et al., 2011) , there has been very little work to your knowledge bridging the statistical topic models with the word embedding techniques.

Computer vision. In modern computer vision, an image is usually described by a set of local descriptors extracted from small image patches such as SIFT. This local representation provides some invariance to changes in viewpoint, lighting or occlusion. The local descriptors characterize the low-level properties of the image such as its color, texture or shape. Since it is computationally intensive to handle (e.g. to match) sets of descriptors of variable cardinality, it has been proposed to aggregate the local descriptors into a global vector which is more amenable to retrieval and classification.

The most popular aggregation mechanism was directly inspired by the work in text analysis. It makes use of an intermediate representation – the *visual vocabulary* – which is a set of prototypical descriptors – the *visual words* – obtained through a clustering algorithm such as k-means (Leung and Malik, 1999; Sivic and Zisserman, 2003; Csurka et al., 2004). Given an image, each of its descriptors is assigned to its closest visual word and the image is described by the histogram of visual words frequencies. This representation is referred to as the *Bag-of-Visual-words* (BoV).

Some works pushed the analogy with text analysis even further. For instance, in large-scale re-

trieval, Sivic and Zisserman proposed to use a tf-idf weighting of the BoV vector and an inverted file for efficient matching (Sivic and Zisserman, 2003). As another example, pLSA, LDA and their many variations have been extensively applied to problems such as image classification (Quelhas et al., 2005) or object discovery (Russell et al., 2006). However, it has been noted that the quantization process mentioned above incurs a loss of information since a *continuous descriptor* is transformed into a *discrete value* (the index of the closest visual word). To overcome this limitation, several improvements have been proposed which depart from the pure discrete model. These improvements include the soft assignment of descriptors to visual words (Farquhar et al., 2005; Philbin et al., 2008; Gemert et al., 2008) or the use of more advanced coding techniques than vector quantization such as sparse coding (Yang et al., 2009) or locality-constrained linear coding (Wang et al., 2010).

All the previous techniques can be understood (with some simplifications) as simple counting mechanisms (computation of 0-order statistics). It has been proposed to take into account higher-order statistics (first and second order for instance) which encode more descriptor-level information and therefore incur a lower loss of information. This includes the Fisher vector (Perronnin and Dance, 2007; Perronnin et al., 2010), which was directly inspired by the Fisher kernel of Jaakkola and Haussler (Jaakkola and Haussler, 1999). In a nutshell, the Fisher vector consists in modeling the distribution of patches in any image with a Gaussian mixture model (GMM) and then in describing an image by its deviation from this average probability distribution. In a recent evaluation (Chatfield et al., 2011), it has been shown experimentally that the Fisher vector was the state-of-the-art representation for image classification. However, in this work *we question the treatment of words as discrete entities*. Indeed, intuitvely some words are closer to each other from a semantic standpoint and words can be embedded in a continuous space as is done for instance in LSA.

3 The Bag-of-Embedded-Words (BoEW)

In this work, we draw inspiration from the work in the computer vision community: we model the generation process of words with continuous mixture models and use the FK for aggregation.

The proposed bag-of-embedded-words proceeds as follows: **Learning phase.** Given an unlabeled training set of documents:

1. Learn an embedding of words in a low-dimensional space, *i.e.* lower-dimensional than the VSM. After this operation, each word w is then represented by a vector of size e:
$$w \rightarrow E_w = [E_{w,1}, \ldots, E_{w,e}]. \quad (1)$$
2. Fit a probabilistic model – *e.g.* a mixture model – on the continuous word embeddings.

Document representation. Given a document whose BoW representation is $\{w_1, \ldots, w_T\}$:

1. Transform the BoW representation into a BoEW:
$$\{w_1, \ldots, w_T\} \rightarrow \{E_{w_1}, \ldots, E_{w_T}\} \quad (2)$$
2. Aggregate the continuous word embeddings E_{w_t} using the FK framework.

Since the proposed framework is independent of the particular embedding technique, we will first focus on the modeling of the generation process and on the FK-based aggregation. We will then compare the proposed continuous topic model to the traditional LSI, PLSA and LDA topic models.

3.1 Probabilistic modeling and FK aggregation

We assume that the continuous word embeddings in a document have been generated by a "universal" (i.e. document-independent) probability density function (pdf). As is common practice for continuous features, we choose this pdf to be a Gaussian mixture model (GMM) since any continuous distribution can be approximated with arbitrary precision by a mixture of Gaussians. In what follows, the pdf is denoted u_λ where $\lambda = \{\theta_i, \mu_i, \Sigma_i, i = 1 \ldots K\}$ is the set of parameters of the GMM. θ_i, μ_i and Σ_i denote respectively the mixture weight, mean vector and covariance matrix of Gaussian i. For computational reasons, we assume that the covariance matrices are diagonal and denote σ_i^2 the variance vector of Gaussian i, i.e. $\sigma_i^2 = \text{diag}(\Sigma_i)$. In practice, the GMM is estimated offline with a set of continuous word embeddings extracted from a representative set of documents. The parameters λ are estimated through the optimization of a Maximum

Likelihood (ML) criterion using the Expectation-Maximization (EM) algorithm.

Let us assume that a document contains T words and let us denote by $X = \{x_1, \ldots, x_T\}$ the set of continuous word embeddings extracted from the document. We wish to derive a fixed-length representation (i.e. a vector whose dimensionality is independent of T) that characterizes X with respect to u_λ. A natural framework to achieve this goal is the FK (Jaakkola and Haussler, 1999). In what follows, we use the notation of (Perronnin et al., 2010).

Given u_λ one can characterize the sample X using the score function:

$$G_\lambda^X = \nabla_\lambda^T \log u_\lambda(X). \quad (3)$$

This is a vector whose size depends only on the number of parameters in λ. Intuitively, it describes in which direction the parameters λ of the model should be modified so that the model u_λ better fits the data. Assuming that the word embeddings x_t are iid (a simplifying assumption), we get:

$$G_\lambda^X = \sum_{t=1}^{T} \nabla_\lambda \log u_\lambda(x_t). \quad (4)$$

Jaakkola and Haussler proposed to measure the similarity between two samples X and Y using the FK:

$$K(X, Y) = G_\lambda^{X\prime} F_\lambda^{-1} G_\lambda^Y \quad (5)$$

where F_λ is the Fisher Information Matrix (FIM) of u_λ:

$$F_\lambda = E_{x \sim u_\lambda} \left[\nabla_\lambda \log u_\lambda(x) \nabla_\lambda \log u_\lambda(x)' \right]. \quad (6)$$

As F_λ is symmetric and positive definite, it has a Cholesky decomposition $F_\lambda = L'_\lambda L_\lambda$ and $K(X, Y)$ can be rewritten as a dot-product between normalized vectors $\mathcal{G}_\lambda$ with:

$$\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X. \quad (7)$$

(Perronnin et al., 2010) refers to $\mathcal{G}_\lambda^X$ as the *Fisher Vector* (FV) of X. Using a diagonal approximation of the FIM, we obtain the following formula for the gradient with respect to μ_i [1]:

$$\mathcal{G}_i^X = \frac{1}{\sqrt{\theta_i}} \sum_{t=1}^{T} \gamma_t(i) \left(\frac{x_t - \mu_i}{\sigma_i} \right). \quad (8)$$

[1] we only consider the partial derivatives with respect to the mean vectors since the partial derivatives with respect to the mixture weights and variance parameters carry little additional information (we confirmed this fact in preliminary experiments).

where the division by the vector σ_i should be understood as a term-by-term operation and $\gamma_t(i) = p(i|x_t, \lambda)$ is the soft assignment of x_t to Gaussian i (*i.e.* the probability that x_t was generated by Gaussian i) which can be computed using Bayes' formula. The FV $\mathcal{G}_\lambda^X$ is the concatenation of the $\mathcal{G}_i^X$, $\forall i$. Let e be the dimensionality of the continuous word descriptors and K be the number of Gaussians. The resulting vector is $e \times K$ dimensional.

3.2 Relationship with LSI, PLSA, LDA

Relationship with LSI. Let n be the number of documents in the collection and t be the number of indexing terms. Let A be the $t \times n$ document matrix. In LSI (or NMF), A decomposes as:

$$A \approx U \Sigma V' \quad (9)$$

where $U \in \mathbb{R}^{t \times e}$, $\Sigma \in \mathbb{R}^{e \times e}$ is diagonal, $V \in \mathbb{R}^{n \times e}$ and e is the size of the embedding space. If we choose $V\Sigma$ as the LSI document embedding matrix – which makes sense if we accept the dot-product as a measure of similarity between documents since $A'A \approx (V\Sigma)(V\Sigma)'$ – then we have $V\Sigma \approx A'U$. This means that the LSI embedding of a document is approximately the sum of the embedding of the words, weighted by the number of occurrences of each word.

Similarly, from equations (4) and (7), it is clear that the FV $\mathcal{G}_\lambda^X$ is a sum of non-linear mappings:

$$x_t \rightarrow L_\lambda \nabla_\lambda \log u_\lambda(x_t) = \left[\frac{\gamma_t(1)}{\sqrt{\theta_1}} \frac{x_t - \mu_1}{\sigma_1}, \ldots, \frac{\gamma_t(K)}{\sqrt{\theta_K}} \frac{x_t - \mu_K}{\sigma_K} \right] \quad (10)$$

computed for each embedded-word x_t. When the number of Gaussians $K = 1$, the mapping simplifies to a linear one:

$$x_t \rightarrow \frac{x_t - \mu_1}{\sigma_1} \quad (11)$$

and the FV is simply a whitened version of the sum of word-embeddings. Therefore, if we choose LSI to perform word-embeddings in our framework, the Fisher-based representation is similar to the LSI document embedding in the one Gaussian case. This does not come as a surprise in the case of LSI since Singular Value Decomposition (SVD) can be viewed as a the limite case of a probabilistic model with a Gaussian noise assumption (Salakhutdinov and Mnih, 2007). Hence, the proposed framework enables to model documents when the word embeddings are non-Gaussian.

Another advantage is that the proposed framework is rotation and scale invariant. Indeed, while it "makes sense" to use $V\Sigma$ as the document embedding, in practice better results can be obtained when using simply V. Our framework is independent of such an arbitrary choice.

Relationship with PLSA and LDA. There is also a strong parallel between topic models on discrete word occurrences such as PLSA/LDA and the proposed model for continuous word embeddings. Indeed, both generative models include a latent variable which indicates which mixture generates which words. In the LDA case, each topic is modeled by a multinomial distribution which indicates the frequency of each word for the particular topic. In the mixture model case, each mixture component can be loosely understood as a "topic".

Therefore, one could wonder if the proposed framework is not somehow equivalent to topic models such PLSA/LDA. The major difference is that PLSA, LDA and other topic models on word counts *jointly* perform the embedding of words and the learning of the topics. A major deficiency of such approaches is that they cannot deal with words which have not been seen at training time. In the proposed framework, these two steps are *decoupled*. Hence, we can cope with words which have not been seen during the training of the probabilistic model. We will see in section 4.3.1 that this yields a major benefit: the mixture model can be trained efficiently on a small subset of the corpus and yet generalize to unseen words.

In the same manner, our work is significantly different from previous attempts at applying the FK framework to topic models such as PLSA (Hofmann, 2000; Chappelier and Eckard, 2009) or LDA (Chandalia and Beal, 2006) (we will refer to such combinations as FKPLSA and FKLDA). Indeed, while FKPLSA and FKLDA can improve over PLSA and LDA respectively, they inherit the deficiencies of the original PLSA and LDA approaches, especially their unability to deal with words unseen at training time. We note also that FKPLSA is extremely computationally intensive: in the recent (Chappelier and Eckard, 2009), the largest corpus which could be handled contained barely 7,466 documents. In contrast, we can easily handle on a single machine corpora containing hundreds of thousands of documents (see section 4.2).

Collection	#docs	#terms	#classes
20NG	19,995	32,902	20
TDT	4,214	8,978	18

(a) Clustering

Collection	#docs	#terms	#queries
ROBUST	490,779	87,223	250
TREC1&-3	741,856	108,294	150
CLEF03	166,754	79,008	60

(b) IR

Table 1: Characteristics of the clustering and IR collections

4 Experiments

The experiments aim at demonstrating that the proposed *continuous* model is competitive with existing topic models on *discrete* words. We focus our experiments on the case where the embedding of the continuous words is obtained by LSI as it enables us to compare the quality of the document representation obtained originally by LSI and the one derived by our framework on top of LSI. In what follows, we will refer to the FV on the LSI embedding simply as the FV.

We assessed the performance of the FV on clustering and ad-hoc IR tasks. We used two datasets for clustering and three for IR. Using the Lemur toolkit (Ogilvie and Callan, 2001), we applied a standard processing pipeline on all these datasets including stopword removal, stemming or lemmatization and the filtering of rare words to speed up computations. The GMMs were trained on 1,000,000 word occurences, which represents roughly 5,000 documents for the collections we have used. In what follows, the cosine similarity was used to compare FVs and LSI document vectors.

4.1 Clustering

We used two well-known and publicly available datasets which are 20 NewsGroup (20NG) and a subset of one TDT dataset (http://www.ldc.upenn.edu/ProjectsTDT2004, 2004). The 20NG is a classical dataset when evaluating classifiers or clustering methods. For the TDT dataset we retain only topics with more than one hundred documents, which resulted in 18 classes. After preprocessing, the 20NG collection has approximately 20,000 documents and 33,000 unique words and the TDT has approximately 4,000 documents and 9,000 unique words. Table

Collection	Model	ARI	NMI
20NG	PLSA	41.0	57.4
	LDA	40.7	57.9
	LSI	41.0	59.5
	FV	45.2	60.7
TDT	PLSA	64.2	84.5
	LDA	69.4	86.4
	LSI	72.1	88.5
	FV	70.4	88.2

Table 2: Clustering experiments on 20NG and the WebKB TDT Corpus: Mean performance over 20 runs (in %).

1 (a) gives the general statistics of the two datasets after preprocessing.

We use 2 standard evaluation metrics to assess the quality of the clusters, which are the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) and Normalized Mutual Information (NMI) (Manning and Schütze, 1999). These measures compare the clusters with respect to the partition induced by the category information. The ARI and NMI range between 0 and 1 where 1 indicates a perfect match with the categories. For all the clustering methods, the number of clusters is set to the true number of classes of the collections and the performances were averaged over 20 runs.

We compared spherical k-means on the FV document representations to topic models such as PLSA and LDA[2]. We choose a priori an embedding of size $e = 20$ for both datasets for LSI and therefore for the FV. LDA and PLSA were trained on the whole dataset. For the FV, we varied the number of Gaussians (K) to analyze the evolution of performances. Table 2 shows the best results for the FV and compares them to LSI, PLSA and LDA. First, LDA has lower performance than LSI in our experiments as reported by several studies which showed that LDA does not necessarily outperform LSI (Wang et al., 2011; Maas et al., 2011). Overall, the FV outperforms all the other models on 20NG and probabilistic topic models on TDT.

[2]We use Blei's implementation available at http://www.cs.princeton.edu/ blei/lda-c/

4.2 Retrieval

We used three IR collections, from two evaluation campaigns: TREC[3] and CLEF[4]: Table 1 (b) gives the statistics of the collections we retained: (i) ROBUST (TREC), (ii) the English subpart of CLEF03 AdHoc Task and (iii) the TREC 1&2 collection, with 150 queries corresponding to topics 51 to 200. For the ROBUST and TREC 1&2 collections, we used standard Porter stemming. For CLEF, words were lemmatized. We removed rare words to speed up the computation of LSI. Performances were measured with the Mean Average Precision (MAP) over the top 1,000 retrieved documents. All the collections have more than 80,000 unique words and approximately 166,000 documents for CLEF, 500,000 for ROBUST and 741,000 for TREC. LSI was computed on the whole dataset and the GMMs were trained on a random subset of 5,000 documents. We then computed the FVs for all documents in the collection. Note that we did not compute topic models with LDA on these datasets as LSI provides similar performances to LDA (Wang et al., 2011; Bai et al., 2009).

Table 3 shows the evolution of the MAP for the LSI baseline with respect to the size of the latent space. Note that we use Matlab to compute singular valued decompositions and that some numbers are missing in this table because of the memory limitations of our machine. Figure 1 shows the evolution of the MAP for different numbers of Gaussians (K) for respectively the CLEF, TREC and ROBUST datasets. For all these plots, FV performances are displayed with a circle and LSI with crosses. We tested an embedding of size $e = 50$ and $e = 200$ for the CLEF dataset, an embedding of size $e = 100$ and $e = 200$ for the TREC dataset and $e = 100$ and $e = 300$ for ROBUST. All these figures show the same trend: a) the performance of the FV increases up to 16 Gaussians and then reaches a plateau and b) the FV significantly outperforms LSI (since it is able to double LSI's performance in several cases). In addition, the LSI results in table 3 (a) indicate that LSI with more dimensions will not reach the level of performance obtained by the FV.

[3]trec.nist.gov

[4]www.clef-campaign.org

e	50	100	200	300	400	500
CLEF	4.0	6.7	9.2	11.0	13.0	13.9
TREC-1 &2	2.2	4.3	6.5	8.3	-	-
ROBUST	1.3	2.4	3.6	4.5	-	-

Table 3: LSI MAP (%) for the IR datasets for several sizes of the latent subspace.

4.3 Discussion

In the previous section we validated the good behavior of the proposed continuous document representation. In the following parts, we conduct additional experiments to further show the strengths and weaknesses of the proposed approach.

IR Baselines. If the FV based on LSI word embeddings significantly outperforms LSI, it is outperformed by strong IR baselines such as Divergence From Randomness (DFR) models (Amati and Rijsbergen, 2002) or Language Models (Ponte and Croft, 1998). This is what we show in table 4 with the PL2 DFR model compared to standard TFIDF, the best FV and LSI.

Collection	PL2	TFIDF	FV	LSI
CLEF'03	35.7	16.4	23.7	9.2
TREC-1&2	22.6	12.4	10.8	6.5
ROBUST	24.8	12.6	10.5	4.5

Table 4: Mean Average Precision(%) for the PL2 and TFIDF model on the three IR Collections compared to Fisher Vector and LSI

These results are not surprising as it has been shown experimentally in many studies that latent-based approaches such as LSI are generally outperformed by state-of-the-art IR models in Ad-Hoc tasks. There are a significant gap in performances between LSI and TFIDF and between TFIDF and the PL2 model. The first gap is due to the change in representation, from a vector space model to latent based representation, while the second one is only due to a 'better' similarity as both methods operate in a similar space. In a way, the FV approach offers a better similarity for latent representations even if several improvements could be further proposed (pivoted document length normalization, combination with exact representation).

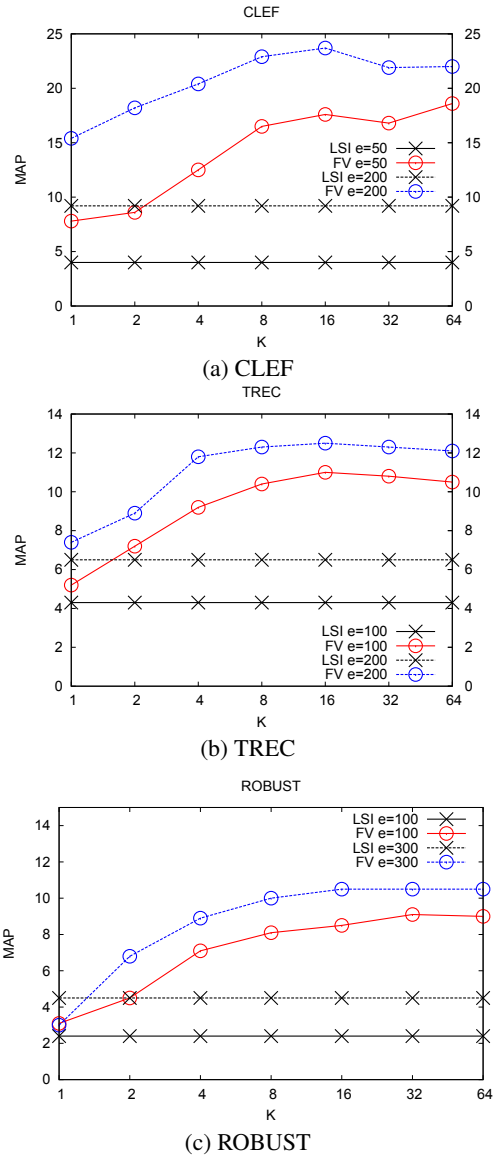

Figure 1: MAP(%) for the FV with different numbers of Gaussians against LSI on the CLEF, TREC and ROBUST datasets

4.3.1 Influence of Training Set Size and Unseen Words.

One of the main benefits of our method is its ability to cope with unseen words: our framework allows to assign probabilities for words unseen while training the topic model assuming that they can be embedded in the Euclidean space. Thus, one can train the probabilistic model on a subpart of the collection without having to discard unseen words at test time. Therefore, we can easily address large-scale collections as we can restrict the GMM learning step on a subset of a collection of documents. This is something that LDA cannot cope with as the vocabulary is frozen at training

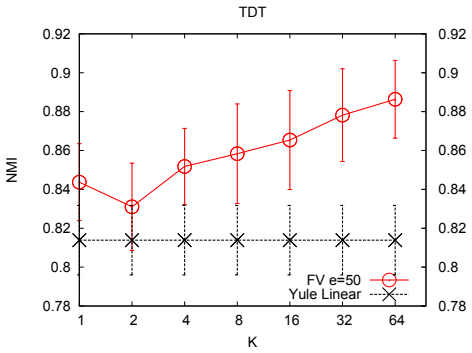

Figure 2: NMI for the FV with different Number of Gaussians against averaging word embeddings on TDT with the Yule measure

time. We show in figure 5 that our model is robust to the number of word occurences used for training the GMM. We illustrate this point using the TREC 1-&2 collection with an embedding of size $e = 100$. We varied the number of documents to train the GMM. Figure 5 shows that increasing the number of documents does not lead to improvements and the performance remains stable. Therefore, these empirical results indeed confirm that we can adress large-scale collections as we can restrict the learning step on a small subset of a collection of documents.

4.3.2 Beyond LSI Embedding

While we focused in the experimental section on word embeddings obtained with LSI, we now show that the proposed framework can be applied to other word embeddings. To do so, we use a word embedding based on the Yule association measure (Jagarlamudi et al., 2011) which is closely related to the Mutual Information but relies on the raw frequencies rather than on probabilities. We use this measure to compute a similarity matrix between words. Then, we applied a spherical kmeans on this matrix to find $e = 50$ word clusters and used the cluster centroids as the word embedding matrix. A simple baseline is to use as document representation the average word embedding as is the case of LSI. The baseline gets 82% NMI wherease the FV with 32 Gaussians reaches 88%. The non-linear mapping induced by the FV always outperforms the simple averaging. Therefore, it is worthwhile to learn non-linear mappings.

5 Conclusion

In this work, we proposed to treat documents as bags-of-embedded-words (BoEW) and to learn

M	# docs	MAP TREC
0.5M	$\approx$ 2,700	11.0
1M	$\approx$ 5,400	11.0
5M	$\approx$ 27,000	10.6
10M	$\approx$ 54,000	10.6

Table 5: Model performance for different subsets used to train the GMM. M refers to a million word occurences

probabilistic mixture models *once* words were embedded in a Euclidean space. This is a significant departure from the vast majority of the works in the machine learning and information retrieval communities which deal with words as discrete entities. We assessed our framework on several clustering and ad-hoc IR collections and the experiments showed that our model is able to yield effective descriptors of textual documents. In particular, the FV based on LSI embedding was shown to significantly outperform LSI for retrieval tasks.

There are many possible applications and generalizations of our framework. In this study, we focused on the LSI embedding and showed preliminary results with the Yule embedding. Since we believe that the word embedding technique is of crucial importance, we would like to experiment with recent embedding techniques such as the Collobert and Weston embedding (Collobert and Weston, 2008) which has been shown to scale well in several NLP tasks.

Moreover, another significant advantage of the proposed framework is that we could deal seamlessly with collections of multilingual documents. This requires the ability to embedd the words of different languages and techniques exist to perform such an embedding including Canonical Correlation Analysis. Finally, the GMM still has several theoretical limitations to model textual documents appropriately so that one could design a better statistical model for bags-of-embedded-words.

References

Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389.

B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Weinberger. 2009. Supervised semantic indexing. In *Proceeding of the 18th ACM CIKM*.

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. 2009. Curriculum learning. In *ICML*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *JMLR*.

Gaurav Chandalia and Matthew J. Beal. 2006. Using fisher kernels from topic models for dimensionality reduction.

Jonathan Chang, Jordan Boyd-graber, Sean Gerrish, Chong Wang, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *NIPS*.

Jean-Cédric Chappelier and Emmanuel Eckard. 2009. Plsi: The true fisher kernel and beyond. In *ECML/PKDD (1)*.

K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. 2011. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*.

G. Csurka, C. Dance, L Fan, J. Willamowski, and C. Bray. 2004. Visual categorization with bags of keypoints. In *Proc. of ECCV Workshop on Statistical Learning for Computer Vision*.

Scott Deerwester. 1988. Improving Information Retrieval with Latent Semantic Indexing. In *Proceedings of (ASIS '88)*.

Paramveer S. Dhillon, Dean Foster, and Lyle Ungar. 2011. Multi-view learning of word embeddings via cca. In *NIPS*, volume 24.

Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. 2011. Sparse additive generative models of text. In Lise Getoor and Tobias Scheffer, editors, *ICML*, pages 1041–1048. Omnipress.

Jason Farquhar, Sandor Szedmak, Hongying Meng, and John Shawe-Taylor. 2005. Improving "bag-of-keypoints" image categorisation: Generative models and pdf-kernels. Technical report, University of Southampton.

Éric Gaussier and Cyril Goutte. 2005. Relation between PLSA and NMF and implications. In *SIGIR*.

Jan Van Gemert, Jan-Mark Geusebroek, Cor Veenman, and Arnold Smeulders. 2008. Kernel codebooks for scene categorization. In *European Conference on Computer Vision (ECCV)*.

Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. 2004. Integrating topics and syntax. In *NIPS*.

Matthew D. Hoffman, David M. Blei, and Francis Bach. 2010. Online learning for latent dirichlet allocation. In *In NIPS*.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *SIGIR*. ACM.

T. Hofmann. 2000. Learning the similarity of documents: An information geometric approach to document retrieval and categorization. In *Neural Information Processing Systems*.

http://www.ldc.upenn.edu/ProjectsTDT2004. 2004. TDT: Annotation manual - version 1.2.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification*.

Tommi S. Jaakkola and David Haussler. 1999. Exploiting generative models in discriminative classifiers. In *NIPS*, Cambridge, MA, USA. MIT Press.

Jagadeesh Jagarlamudi, Raghavendra Udupa, Hal Daumé III, and Abhijit Bhole. 2011. Improving bilingual projections via sparse covariance matrices. In *EMNLP*, pages 930–940.

D. D. Lee and H. S. Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October.

T. Leung and J. Malik. 1999. Recognizing surfaces using three-dimensional textons. In *IEEE International Conference on Computer Vision (ICCV)*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.

Andriy Mnih and Geoffrey E. Hinton. 2007. Three new graphical models for statistical language modelling. In Zoubin Ghahramani, editor, *ICML*, volume 227 of *ACM International Conference Proceeding Series*, pages 641–648. ACM.

Paul Ogilvie and James P. Callan. 2001. Experiments Using the Lemur Toolkit. In *TREC*.

Florent Perronnin and Christopher R. Dance. 2007. Fisher kernels on visual vocabularies for image categorization. In *CVPR*.

Florent Perronnin, Yan Liu, , Jorge Sánchez, and Hervé Poirier. 2010. Large-scale image retrieval with compressed fisher vectors. In *CVPR*.

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2008. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281. ACM.

P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. 2005. Modeling scenes with local descriptors and latent aspects. In *IEEE International Conference on Computer Vision (ICCV)*.

B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. 2006. Using multiple segmentations to discover objects and their extent in image collections. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

R. Salakhutdinov and A. Mnih. 2007. Probabilistic matrix factorization. In *NIPS*.

G. Salton and M. J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

J. Sivic and A. Zisserman. 2003. Video google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision (ICCV)*.

Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*, pages 384–394.

J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. 2010. Locality-constrained linear coding for image classification. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Quan Wang, Jun Xu, Hang Li, and Nick Craswell. 2011. Regularized latent semantic indexing. In *SIGIR'11*.

J. Yang, K. Yu, Y. Gong, and T. Huang. 2009. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.