Textual Similarity with a Bag-of-Embedded-Words Model

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 experiments in the case where the word-embeddings are computed from standard topic models showing significant improvements with respect to the original topic models.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Theory, Algorithms, Experimentation

Keywords

IR Theory, Probabilistic Models, Word Embeddings,Fisher Kernel

1. INTRODUCTION

The Vector Space Model (VSM) or bag-of-words (BoW) representation is at the root of topic models such as Latent Semantic Indexing (LSI) [8], Probablistic Latent Semantic Analysis (PLSA) [11] or Latent Dirichlet Allocation (LDA) [3]. 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). Using the Fisher Kernel framework [12], 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 [5] 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 follows. In section 2, we describe the proposed framework. In section 3, we report and discuss retrieval tasks. Then, we provide a discussion of the proposed framework and related works in section 4, before concluding in section 5.

2. THE BAG-OF-EMBEDDED-WORDS

One source of inspiration for this work has been the recent progress in computer vision. Indeed, state-of-the-art image retrieval and classification systems describe an image as an unordered set (*i.e.* a bag) of descriptors extracted from small image patches. Since it is computationally intensive to handle (*e.g.* to match) sets of descriptors, it has been proposed to aggregate the patch descriptors into a global vector which is more amenable to retrieval and classification. While image representations were initially inspired by the BoW [7] of text analysis, they have been improved in many ways to take into account the continuous nature of the local descriptors [15, 17]. While words are typically treated as discrete entities in the IR commmunity, we saw that they

ICTIR '13, September 29 - October 02 2013, Copenhagen, Denmark

http://dx.doi.org/10.1145/2499178.2499180

can be embedded in a Euclidean space which reflects some a priori notion of similarity between them. Therefore, it only seems natural to try to understand whether the improvements proposed in computer vision could be translated back to the field of text analysis. The solution we propose learns probabilitic models for embedded words. It consists of the following steps:

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 using the GMM generative model

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.

2.1 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 [12]. We note that the FK has already been applied to GMMs in other fields than text analysis, for instance in computer vision (see [15, 16]). In what follows, we use the notation of [16]. 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:

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

[16] refers to $\mathcal{G}_\lambda^X$ as the *Fisher Vector* (FV) of X. We obtain the following formula for the gradient with respect to μ_i [1]

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

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.

2.2 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:

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

[1] Following [16] we only consider the partial derivatives with respect to the mean vectors.

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 Decmposition (SVD) can be viewed as a the limite case of a probabilistic model with a Gaussian noise assumption [18]. Hence, the proposed framework enables to model documents when the word embeddings are non-Gaussian.

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". 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 (assuming they can be embedded). The, the mixture model can be trained efficiently on a small subset of the corpus and yet generalize to unseen words which is necessary for transfer learning applications.

In the same manner, our work is significantly different from previous attempts at applying the FK framework to topic models such as PLSA [5] or LDA [4] (we will refer to such combinations as FKPLSA and FKLDA). Indeed, while FKPLSA and FKLDA can improve over PLSA and LDA respectively, they are extremely computationally intensive: in the recent [5], he 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 3).

3. 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 used three IR collections, from two evaluation campaigns: TREC[2] and CLEF[3]: We used a standard preprocessing pipeline and performances were measured with the Mean Average Precision (MAP). 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.

[2]trec.nist.gov

[3]www.clef-campaign.org

e	50	100	200	300	400	500
CLEF	6.0	8.1	12.1	11.0	13.0	15.1
TREC-1 &2	2.2	4.8	7.15	8.6	-	-
ROBUST	1.5	3.0	3.6	5.3	-	-

Table 1: LSI MAP (%) for the IR datasets for several sizes of the latent subspace.

			Number of Gaussians (M)				
Dataset	e	LSI	2	4	8	16	32
clef	50	6.0	8.6	12.5	16.5	**17**.6	16.8
	200	12.1	18.2	20.4	22.9	**23.7**	21.9
trec	100	4.8	7.2	9.2	10.4	**11.0**	10.8
	200	7.15	8.9	11.8	12.3	**12.5**	12.3
robust	100	3.0	4.5	7.1	8.1	8.5	**9.1**
	300	5.3	6.8	8.9	10	10.5	**10.5**

Table 2: FV significantly outperforms LSI. MAP(%) for FV with several embedding size e and Gaussians M on the IR datasets.

Table 1 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. Table 2 shows the evolution of the MAP for different numbers of Gaussians (K) for respectively the CLEF, TREC and ROBUST datasets. 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 numbers 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 1 (a) indicate that LSI with more dimensions will not reach the level of performance obtained by the FV.

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 [1]. This is what we show in table 3 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 3: 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 such as pivoted document normalization, combination with exact representation etc.

4. DISCUSSION

In the previous section we validated the good behavior of the proposed continuous document representation. We have shown that it is beneficial to introduce non-linear mappings.

Our work is clearly related to textual dimensionality reduction techniques. Of course, many extensions of NMF [13] and LDA have been proposed [10, 9]. Similarly, LSI has been refined in [19] where the influence of ℓ_1 and ℓ_2 regularization was studied.

Parallel 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. A seminal work in this field is the one by Collobert and Weston [6] (later refined in [2]) where a neural network is trained by stochastic gradient descent in order to minimize a loss function on the observed n-grams. [14] parametrizes a probabilistic model in order to capture word representations, instead of modeling individually latent topics. Except [14], there has been very little work to your knowledge bridging the statistical topic models with the word embedding techniques. If our work can also be viewed as a Kernel for LSI, our initial motivation was to be able to inject prior knowledge or metric into textual probabilitic models. We underline again that our contribution is **not** the application of the FK to text analysis. We propose a generic probabilistic model that rely on word embedding and can therefore benefit from a priori information on the similarity between words.

5. CONCLUSION

In this work, we proposed to treat documents as bags-of-embedded-words (BoEW) and to learn probabilistic topic 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 ad-hoc IR collections and the experiments showed that our model is able to yield effective descriptors of textual documents: 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. 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 [6] which has been shown to scale well in several NLP tasks. In addition, Canonical Correlation Analysis, as an embedding technique, could enable us to deal seamlessly with multilingual corpora. 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.

Acknowledgements

This research was partly supported by the European Project FUPOL FP7-ICT-287119.

6. REFERENCES

[1] G. Amati and C. J. V. Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, 2002.

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

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 2003.

[4] G. Chandalia and M. J. Beal. Using fisher kernels from topic models for dimensionality reduction, 2006.

[5] J.-C. Chappelier and E. Eckard. Plsi: The true fisher kernel and beyond. In *ECML/PKDD (1)*, 2009.

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

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

[8] S. Deerwester. Improving Information Retrieval with Latent Semantic Indexing. In *Proceedings of (ASIS '88)*, 1988.

[9] J. Eisenstein, A. Ahmed, and E. P. Xing. Sparse additive generative models of text. In L. Getoor and T. Scheffer, editors, *ICML*, pages 1041–1048. Omnipress, 2011.

[10] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum. Integrating topics and syntax. In *NIPS*, 2004.

[11] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*. ACM, 1999.

[12] T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *NIPS*, Cambridge, MA, USA, 1999. MIT Press.

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

[14] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150, 2011.

[15] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

[16] F. Perronnin, Y. Liu, , J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, 2010.

[17] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *European Conference on Computer Vision (ECCV)*, 2010.

[18] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, 2007.

[19] Q. Wang, J. Xu, H. Li, and N. Craswell. Regularized latent semantic indexing. In *SIGIR'11*, 2011.