Pattern Recognition 81 (2018) 254–267

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Learning bag-of-embedded-words representations for textual information retrieval

Nikolaos Passalis*, Anastasios Tefas

Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece

ARTICLE INFO

Article history:
Received 23 October 2017
Revised 9 February 2018
Accepted 8 April 2018
Available online 10 April 2018

Keywords:
Word embeddings
Bag-of-words
Bag-of-features
Dictionary learning
Relevance feedback
Information retrieval

ABSTRACT

Word embedding models are able to accurately model the semantic content of words. The process of extracting a set of word embedding vectors from a text document is similar to the feature extraction step of the Bag-of-Features (BoF) model, which is usually used in computer vision tasks. This gives rise to the proposed Bag-of-Embedded Words (BoEW) model that can efficiently represent text documents overcoming the limitations of previously predominantly used techniques, such as the textual Bag-of-Words model. The proposed method extends the regular BoF model by a) incorporating a weighting mask that allows for altering the importance of each learned codeword and b) by optimizing the model end-to-end (from the word embeddings to the weighting mask). Furthermore, the BoEW model also provides a fast way to fine-tune the learned representation towards the information need of the user using relevance feedback techniques. Finally, a novel spherical entropy objective function is proposed to optimize the learned representation for retrieval using the cosine similarity metric.

1. Introduction

The textual *Bag-of-Words* (BoW) representation [1], is among the prevalent techniques used for textual Information Retrieval (IR). In the textual BoW model a set of predefined words, called *dictionary*, is selected and then each document is represented by a histogram vector that counts the number of appearances of each word in the document. Its great success in IR tasks has led a great deal of research to be devoted to improve the textual BoW model. For example, some techniques focused on pruning the dictionary [2], while other methods on improving the extracted histograms by applying a weighting scheme, such as the tf–idf (term frequency/inverse document frequency) method [1,3,4]. Furthermore, other more advanced techniques, such as the Latent Dirichlet Allocation (LDA) [5], also use word occurrence statistics to model each document as a *mixture of topics*.

Word embedding models are capable of extracting semantically-enriched representations of words. However, it is not straightforward to use them to encode whole documents. Perhaps the most commonly used technique to overcome this limitation is to simply calculate the average word embedding vector of a document [6–10]. Also, a few sophisticated techniques, such as the Paragraph Vector [11,12], have been proposed to directly calculate embeddings of documents. However, the averaging process ignores part of the information that the document contains, while the paragraph embedding requires a computationally intensive inference step to provide out-of-sample embeddings, which limits its applicability.

In this paper we propose a method that is capable of overcoming the aforementioned limitations by using an *efficient end-to-end trainable text representation* scheme that exploits the representation power of semantic-enriched word embeddings and is inspired by the well known Bag-of-Features (BoF) model. The proposed method also aims at providing a link between the textual Bag-of-Words model, mainly used by the natural language processing and information retrieval communities, and the feature-based Bag-of-Features model, mainly used by the computer vision community. That way, this work paves the way for developing powerful text representation machines for information retrieval building upon the extensive existing research on the BoW-based techniques [1–5], as well as on the BoF-based methods [13–18].

To better understand the link between the BoW and BoF methods, note that the process of extracting a word embedding (feature) vector for each word of a document is similar to the feature extraction step that is used in order to represent multimedia objects, such as images and videos [19,20]. For example, for image recognition/retrieval tasks it is common to extract multiple SIFT vectors [21], from an image and then use the Bag-of-Features technique, also known as Bag-of-Visual Words (BoVW), to extract a constant dimensionality vector from each image [13,22]. Thus, a

* Corresponding author.
E-mail addresses: passalis@csd.auth.gr (N. Passalis), tefas@aiia.csd.auth.gr (A. Tefas).

https://doi.org/10.1016/j.patcog.2018.04.008
0031-3203/

text document comprises of a set of feature vectors (word embeddings) in a similar way to an image that comprises of a set of visual feature vectors (e.g., SIFT vectors). The pipeline of the BoF model can be summarized as follows:

1. *Feature extraction*, in which multiple features, such as SIFT descriptors [21], are extracted from each object, e.g., image. That way, the *feature space* is formed where each object is represented as a set of features.
2. *Codebook learning*, in which the extracted features are used to learn a *codebook* of representative features (also called *codewords*),
3. *Feature quantization* and *encoding*, in which each feature vector is represented using a codeword from the learned codebook and a histogram is extracted for each object. That way, the *vector space* (also referred to as *representation space* or *histogram space* in BoF-based dictionary learning literature) is formed, where each object is represented by a constant dimensionality term/histogram vector, similarly to the vector space model [1].

The similarity between extracting a set of word embedding vectors from a text document and extracting a set of feature vectors from a multimedia object was noticed quite recently [7,23,24]. Exploiting this similarity allows us to use the BoF model to extract representations from text documents using the extracted word embedding vectors as feature vectors. The application of the BoF model in the context of text representation is called Bag-of-Embedded Words (BoEW) model. In [24] the BoF model is used to encode text documents using the extracted word embedding vectors, while in [23], and [7], Fisher vector encoding was used instead to represent each document.

Also, it has been well established that using unsupervised algorithms, such as k-means [25], to learn the codebook of the BoF/BoEW representation leads to suboptimal results [14,26]. Therefore, the codebook of the BoF/BoEW model must be optimized towards the task at hand. Although a wide range of methods exist for learning discriminative dictionaries, e.g., [14,27–29], many of them produce highly discriminative representations that are not always optimal for retrieval tasks. This phenomenon was studied and explained in [13], where an entropy-based retrieval-oriented objective function was proposed. In the case of [13], and [24], the Euclidean distance was used to calculate the entropy. Therefore, the learned representation was optimized for retrieval using the Euclidean distance. However, in most cases using the cosine similarity instead of the Euclidean distance significantly increases the retrieval precision. Motivated by this observation a new type of entropy is proposed in this work, the *spherical entropy*, that optimizes the representation for retrieval using the cosine similarity. In Section 4 it was experimentally demonstrated that this can lead to significant improvements in the retrieval precision.

Furthermore, the BoEW model is extended using a weighting mask that allows us to alter the importance of each codeword. Note that this is similar to the weighting schemes used in the classical BoW schemes, such as the tf–idf. The purpose of using the proposed weighting mask is two-fold: a) it allows for further optimizing the learned representation towards the task at hand and b) allows for quickly fine-tuning the representation towards the information need of the user using relevance feedback techniques [30–32]. The latter is especially important, since a) it provides a very fast way to adjust the representation using the user's feedback without having to re-encode the whole database and b) allows for optimizing the representation when only a few annotated documents are available.

The main contributions of this paper are briefly summarized below. First, the BoF model is adjusted towards representing text documents using word embeddings leading to the proposed BoEW model. The proposed model utilizes a histogram-space weighting mask, inspired by the weighting schemes used in the BoW models, that increases the flexibility of the model and allows for further fine-tuning the representation towards different tasks. Also, the proposed BoEW model is optimized end-to-end, i.e., all the parameters of the model (the word embedding, the codebook, the scaling factor and the weighting mask) are simultaneously learned using the proposed spherical entropy objective, which optimizes the learned representation for retrieval using the cosine similarity. Furthermore, two different optimization algorithms are proposed for the BoEW model: a) an offline algorithm for optimizing the representation using an annotated set of documents and b) a relevance feedback algorithm that allows for quickly optimizing the representation and re-querying the database using the feedback from the user. Finally, both algorithms are evaluated using three collections of text documents from a diverse range of domains and it is demonstrated that they can both increase the retrieval precision and reduce the size of the extracted representation (increasing the retrieval speed and reducing the storage requirements) for both in-domain and out-of-domain retrieval tasks.

The rest of the paper is structured as follows. The related work is discussed in Section 2 and the proposed method is presented in Section 3. The experimental evaluation of the proposed method is presented in Section 4 and conclusions are drawn in Section 5. Finally, note that a reference implementation of the proposed method will be provided at http://github.com/passalis/boew to enable other researcher easily use and extend the proposed technique.

2. Related work

Early text retrieval approaches used the term frequency/inverse document frequency (tf–idf) method to represent documents as vectors. Then, relevant documents can be retrieved by measuring the similarity between a query vector and document vectors stored in the database (vector space model) [1]. Several methods were subsequently developed building upon this model, ranging from tf–idf variants and extensions, such as [3,4], to more advanced topic-based analysis techniques, e.g,. Latent Semantic Indexing (LSI) [33], Probablistic Latent Semantic Indexing (PLSI) [34], and Latent Dirichlet Allocation (LDA) [5].

Even though the aforementioned techniques were used with great success for several information retrieval tasks [1], they ignore part of the semantic relationships between the words that compose the dictionary (the term vectors are equidistant to each other and, thus, fail to capture the semantic similarity between different words). This problem gave rise to word embedding models, such as [35] and [36], where each word is mapped to a dense real-valued vector that captures the semantic properties of the corresponding word and encode the underlying linguistic patterns. That is, vectors that correspond to words with similar meaning are closer to each other than vectors for words with irrelevant semantic content. Among the most well known word embedding models is the word2vec model [35], and the GloVe model [36]. Both use unsupervised algorithms to learn word embeddings either by predicting the words in a given window or by using word co-occurrence statistics. In contrast to these methods, the proposed approach concerns document representation instead of learning embeddings of individual words.

However, it is not straightforward to use word embeddings to represent a document that is composed of multiple words. Among the most commonly used, yet naive, approaches is to average all the word embedding vectors that correspond to the words that a document contains. Even though this approach is widely used in many natural language processing tasks [6–10], the averaging process leads to loss of valuable information that a document contains. To overcome this limitation, Paragraph Vector [11,12,37], and