Word Embedding techniques for Content-based Recommender Systems: an empirical evaluation

Cataldo Musto, Giovanni Semeraro, Marco De Gemmis, Pasquale Lops
University of Bari Aldo Moro, Italy
{name.surname}@uniba.it

ABSTRACT

This work presents an empirical comparison among three widespread *word embedding* techniques as Latent Semantic Indexing, Random Indexing and the more recent Word2Vec. Specifically, we employed these techniques to learn a low-dimensional vector space *word representation* and we exploited it to represent both items and user profiles in a *content-based* recommendation scenario. The performance of the techniques has been evaluated against two state-of-the-art datasets, and experimental results provided good insights which pave the way to several future directions.

1. MOTIVATIONS AND METHODOLOGY

Word Embedding techniques learn in a totally unsupervised way a *low-dimensional* vector space representation of *words* by analyzing their usage in (very) large corpora of textual documents. These approaches are recently gaining more and more attention, since they showed very good performance in a broad range of natural language processing-related scenarios, ranging from *sentiment analysis* and *machine translation* to more challenging ones as learning a textual description of a given image[1].

In a nutshell, all these techniques employ a large corpora of documents to encode the co-occurences between the terms, in order to learn both linguistic regularities as well as *semantic nuances*, according to their usage. Next, given this huge co-occurrences matrix, each technique use a different approach to obtain a smaller low-dimensional representation of each *word* occurring in the original corpus. An important feature which is common to all these technique is that the dimension of the representation (that is to say, the size of the vectors) is just a parameter of the model, so it can be set according to specific constraint or peculiarities of the data.

However, although the effectiveness of such techniques (especially when combined with deep neural network architectures) is already taken for granted, just a few work investigated how well they do perform in recommender systems-related tasks. To this aim, in this work we defined a very simple content-based recommendation framework based on *word embeddings*, in order to assess about the effectiveness of such techniques in these scenarios as well. Specifically, we first exploited word embedding techniques to represent *words* in vector spaces. Next, we inferred a vector-space representation of the *items* by summing the representation of the words occurring in the document. Similarly, user profiles are represented by summing the document representation of the items the user liked. Finally, by exploiting classic similarity measures the available items can be ranked according to their descending similarity with respect to the user profile, and recommendations can be provided, in a typical *Top-N recommendation* setting.

Clearly, this is a very basic formulation, since more *fine-grained representations* can be learned for both items and users profiles. However, this work just aims to preliminarily evaluate the effectiveness of such representations in a simplified recommendation framework, in order to pave the way to several future research directions in the area.

Overview of the techniques. Latent Semantic Indexing (LSI) [1] is a word embedding technique which applies Singular Value Decomposition (SVD) over a word-document matrix. The goal of the approach is to *compress* the original information space through SVD in order to obtain a smaller-scale word-*concepts* matrix, in which each column models a *latent concept* occurring the original vector space. Specifically, SVD is employed to unveil the latent relationships between terms according to their usage in the corpus.

Next, Random Indexing (RI) [3], is an incremental technique to learn a low-dimensional word representation relying on the principles of the Random Projection. It works in two steps: first, a *context vector* is defined for each context (the definition of the context is typically scenario-dependant, it may be a paragraph, a sentence or the whole document). Each context vector is ternary (it contains values in $\{-1, 0, 1\}$) very sparse, and its values are *randomly distributed*. Given such context vectors, the vector space representation of each word is obtained by just summing over all the representations of the contexts in which the word occurs. An important peculiarity of this approach is that it is incremental and scalable: if any new documents come into play, the vector space representation of the terms is updated by just adding the new occurrences of the terms in the new documents.

Finally, Word2Vec (W2V) is a recent technique proposed by Mikolov et al. [2]. The approach learns a vector-space representation of the terms by exploiting a two-layers neu-

[1] http://googleresearch.blogspot.it/2014/11/a-picture-is-worth-thousand-coherent.html

RecSys 2015 Poster Proceedings, September 16-20, 2015, Vienna, Austria.

Table 1: Results of the experiments. The best word embedding approach is highlighted in bold. The best overall configuration is highlighted in bold and underlined. The baselines which are overcame by at least a word embedding are put in italics.

MovieLens	**W2V**		**RI**		**LSI**		**U2U**	**I2I**	**BPRMF**
Vector Size	300	500	300	500	300	500			
F1@5	**0.5056**	0.5054	0.4921	0.4910	0.4645	0.4715	**0.5217**	*0.5022*	0.5141
F1@10	**0.5757**	0.5751	0.5622	0.5613	0.5393	0.5469	**0.5969**	0.5836	0.5928
F1@15	0.5672	**0.5674**	0.5349	0.5352	0.5187	0.5254	**0.5911**	0.5814	0.5876
DBbook	**W2V**		**RI**		**LSI**		**U2U**	**I2I**	**BPRMF**
	300	500	300	500	300	500			
F1@5	0.5183	**0.5186**	0.5064	0.5039	0.5056	0.5076	0.5193	*0.5111*	**0.5290**
F1@10	0.6207	0.6209	0.6239	0.6244	0.6256	**0.6260**	*0.6229*	*0.6194*	**0.6263**
F1@15	0.5829	0.5828	0.5892	0.5887	0.5908	**0.5909**	*0.5777*	*0.5776*	*0.5778*

ral network. In the first step, weights in the network are randomly distributed as in RI. Next, the network is trained by using the Skip-gram methodology in order to model fine-grained regularities in word usage. At each step, weights are updated through Stochastic Gradient Descent and a vector-space representation of each term is obtained by extracting the weights of the network at the end of the training.

2. EXPERIMENTAL EVALUATION

In the experimental evaluation the performance of word embedding representations were compared against two state-of-the-art datasets as MovieLens (ML) and DBbook (DB)[2]. Moreover, we also compared the effectiveness of the best-performing configurations to some widespead baselines.

Experimental Design. Experiments were performed by adopting different protocols: as regards ML, we carried out a 5-folds cross validation, while a single training/test split was used for DB. Textual content was obtained by mapping items to Wikipedia pages. For each word embedding technique we compared two different size of learned vectors: 300 and 500. As regards the baselines, we exploited MyMediaLite library[3]. We evaluated User-to-User (U2U-KNN) and Item-to-Item Collaborative Filtering (I2I-KNN) as well as the Bayesian Personalized Ranking Matrix Factorization (BPRMF). U2U and I2I neighborhood size was set to 80. while BPRMF was run by setting the factor parameter equal to 100. In both cases we chose the optimal values for the parameters. Finally, statistical significance was assessed by exploiting Wilcoxon and Friedman tests, chosen after running the Shapiro-Wilk test which revealed the non-normal distribution of the data.

Discussion of the results. The first six columns of Table 1 provide the results of the comparison among the word embedding techniques. As regards ML, W2V emerged as the best-performing configuration for all the metrics took into account. The gap is significant when compared to both RI and LSI. Moreover, results show that the size of the vectors did not significantly affect the overall accuracy of the algorithms (with the exception of LSI). This is an interesting outcome since with an even smaller word representation, word embeddings can obtain good results. However, the outcomes emerging from this first experiments are controversial, since DBbook data provided opposite results: in this dataset W2V is the best-performing configuration only for F1@5. On the other side, LSI, which performed the worst on MovieLens data, overcomes both W2V and RI on F1@10 and F1@15. On a first sights these results indicate non-generalizable outcomes. However, it is likely that such behavior depends on specific pecularities of the datasets which in turn influence the way the approaches learn their vector-space representations. A more throrough analysis is needed to obtain general guidelines which drive the behavior of such approaches.

Next, we compared our techniques to the above described baselines. Results clearly show that the effectiveness of word embedding approaches is directly dependent on the sparsity of the data. This is an expected behavior since content-based approaches can better deal with cold-start situations. In highly sparse dataset as DBbook (99.13% against 93.59% of MovieLens), content-based approaches based on word embedding tend to overcome the baselines. Indeed, all the approaches overcome I2I and U2U on F1@10 and F1@15 (W2V also overcomes I2I on F1@5). Furthermore, it is worth to note that on F1@10 and F@15 word embeddings can obtain results which are comparable (or even better on F1@15) to those obtained by BPRMF. This is a very important outcome, which definitely confirms the effectiveness of such techniques. Conversely, on less sparse datasets as MovieLens, CF algorithms overcome their content-based counterpart.

However, the overall outcomes emerging from this preliminary investigations are very promising: given that no specific NLP task was performed on the data, it is likely that a more thorough processing of the content can lead to even better results. Thus, this investigation showed that word embedding approaches can represent a very interesting alternative to widespread CF approaches. In the following, we will further validate our results by also further investigating the effectiveness of novel and richer textual *data silos*, as those coming from the Linked Open Data cloud.

3. REFERENCES

[1] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41:391–407, 1990.

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[3] Marcus Sahlgren. An introduction to Random Indexing. In *Methods and Applications of Semantic Indexing Workshop, TKE 2005*, 2005.

[2] http://challenges.2014.eswc-conferences.org/index.php/RecSys

[3] http://www.mymedialite.net/