LEVERAGING FEATURE VECTORS TO EXTRACT DISCRIMINATIVE WORDS

Márius ŠAJGALÍK, Michal BARLA, Mária BIELIKOVÁ

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
{sajgalik,barla,bielik}@fiit.stuba.sk

Abstract. Feature vectors are becoming more popular alternative to traditional methods based on word ontologies such as WordNet or BabelNet. Their usage is much simpler on tasks like word similarity. We research keyword extraction problem and focus on utilisation of feature vectors. We present a novel method to extract discriminative words from text documents, where we represent each word as a vector of its latent features. We evaluate our method on text categorisation task using a well-known 20-newsgroups dataset and achieve state-of-the-art results.

1. Introduction

A few years ago, ontologies and taxonomies were prevalent for understanding the meaning of text. With growing amounts of data being generated, unsupervised approaches for learning word meaning are becoming more popular [4]. They do not require us to manually label a training set like traditional ontology-based approaches, but skip the word disambiguation task altogether. They learn distributed representation of words, which encodes multiple senses (and other properties like syntactic features of a word [5]) into vector of latent features. Such word embedding makes tasks like word similarity as simple as measuring cosine similarity of word vectors [5].

In our approach, we leverage feature vectors of words to extract discriminative words from text documents and evaluate such document representation on text categorization task. Using feature vectors, we can easily search for similar words by measuring vector similarity. Apart from words sense disambiguation, ontologies would also require us to use only some heuristics to measure word similarity. Despite their expressive force, ontologies lack the weight of relations (e.g. [2] describes a relation by type, but cannot determine the weight), since that is hard and laborious task even for human experts.

2. Extracting feature vectors

First, we pre-process each article with Stanford CoreNLP [6] to transform raw text into sequence of words labelled with part-of-speech tags and designed a finite automaton (see Figure 1) accepting candidate phrases. We experimented with choosing only noun phrases after parsing each sentence, however, it proved to be rather impractical. Besides the fact that it skipped the phrases beginning with adjectives, which sometimes significantly influence the meaning of a phrase, it could not handle more complex documents, which contained more structured text formatting like lists or other decorative elements. We also tried choosing only nouns, adjectives, verbs or adverbs, but it yielded worse results than using phrases accepted by the proposed automaton. By using this automaton, we also avoided stopwords and retained only potentially discriminative words.

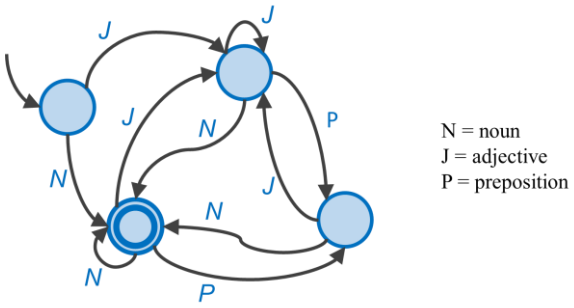

Figure 1. Finite automaton for generating candidate phrases.

We leverage feature vectors to simulate the understanding of word semantics [4]. Feature vector representation maps each word to vector of latent features. We use pre-trained feature vectors[1] trained on part of Google News dataset (about 100 billion words), which were obtained using a simple data-driven approach described in [4] and transform each phrase into the corresponding feature vector. We build on the evidence [4] that multiple word vectors can be summed up to obtain the meaning of a longer phrase. Although the model contains also some phrases, there is no direct mapping for every possible phrase. We use a simple dynamic programming technique described in [1] to minimise number of concatenated unit phrases and thus, prefer the longer unit phrases present in the model.

We use the corpus of feature vectors to substitute the vectors of candidate phrases in the article with vectors of k nearest neighbours present in the corpus. Thus, we get broader understanding of the meaning of each candidate phrase in the article. For each of those substitution feature vectors, we increase its relevance relative to its discriminative force. This force is equal to TF-RF score [3], which has been reported to give better results in text categorisation task than TF-IDF. To obtain k nearest neighbours, we do not use the whole corpus of feature vectors, but reduce it to only 200,000 most frequent words based on Google N-

[1] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing

gram statistics[2]. Since there are many noisy words like typos in the corpus, we can use this reduction, which speeds up the algorithm and does not have negative influence.

3. Evaluation by text categorisation

To evaluate proposed method, we chose text categorisation problem and evaluated our approach on 20-newsgroups dataset. For each document, we computed one feature vector as a normalised sum of vectors of top relevant words output from our method. Thus, each document was encoded into 300-dimensional feature vector. We experimented with different number of top words to create the feature vector of a document. Using linear discriminant analysis we achieved F1 score 82.85% for sum of top 1200 word vectors and using linear SVM we achieved state-of-the-art performance of 84.5% micro-averaged F1 score. We showed that feature vectors can improve extraction of discriminative words and succeeded to achieve state-of-the-art results, which we consider the main contribution of our work. We showed that extracted discriminative words are good for text categorisation, but that also implies the semantic quality of the underlying representation.

Acknowledgement: This work was partially supported by grants No. VG1/0675/11, APVV-0208-10 and it is the partial result of the Research and Development Operational Programme project "University Science Park of STU Bratislava", ITMS 26240220084, co-funded by the European Regional Development Fund.

References

to other papers publishing the results that are summarized here

[1] Šajgalík, M., Barla, M., Bieliková, M.: Exploring Multidimensional Continuous Feature Space to Extract Relevant Words. In: *Proceedings of the Second International Conference on Statistical Language and Speech Processing, SLSP 2014*, Springer-Verlag, (2014), (to appear).

Other references

[2] Barla, M., Bieliková, M.: On Deriving Tagsonomies: Keyword Relations Coming from Crowd. In: *Proc. of the 1st Int. Conf. on Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, Springer-Verlag, (2009), pp. 309–320.

[3] Lan, M., Tan, C., Low, H.: Proposing a New Term Weighting Scheme for Text Categorization. In: *Proc. of the 21st national conf. on AI – Vol. 1*, AAAI Press, (2008), pp. 763-768.

[4] Mikolov, T. et al.: Distributed Representations of Words and Phrases and their Compositionality. In: *Advances in Neural Information Processing Systems 26*, Curran Associates, (2013), pp. 3111-3119.

[5] Mikolov, T., Yih, W., Zweig, G.: Linguistic Regularities in Continuous Space Word Representations. In: *Proc. of NAACL HLT*, ACL, (2013), pp. 746-751.

[6] Socher, R. et al.: Parsing With Compositional Vector Grammars. In: *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL, (2013), pp. 455-465.

[2] `http://storage.googleapis.com/books/ngrams/books/datasetsv2.html`