Proposal for Master's Thesis

Neural Word Embeddings in Practical Information Retrieval

Lukas Galke

Knowledge Discovery Group
Department for Computer Science
Faculty of Engineering
Kiel University

1 Motivation

Beside the notable success of deep neural networks on speech recognition and computer vision tasks, recent progress in the field of Representation Learning also shows that several natural language processing tasks including language modeling, part-of-speech tagging, chunking, named entity recognition, semantic role labeling, syntactic parsing, large scale text classification benefit from neural network based word embeddings. [3, 5] Moreover, the recently proposed neural network based word embeddings *Word2Vec* [8] and *Doc2Vec* [7] lead to superior classification results compared to classical bag-of-words based approaches.[1]

The major advantage of these neural word embedding model over bag-of-words based methods is the ability of capturing the semantics of the input text. While word n-gram models also try to capture those semantics within a certain window, the neural word embeddings are capable of learning a *meaningful* representation. Semantically related terms are located close to each other in the representational space. Moreover, even mathematical computations such as $\text{vec}(\text{'King'}) - \text{vec}(\text{'Man'}) + \text{vec}(\text{'Woman'}) =_{\text{nearest}} \text{vec}(\text{'Queen'})$ can be performed in a properly trained model.

Transferring neural network based techniques successfully to the Information Retrieval (IR) task is a active research field. The IR task mainly consists of two steps: matching and scoring. In the matching step, documents of the corpus are matched against a certain query. The scoring step consists of scoring and therefore ranking the retrieved documents according to their relevancy to the query. As these core IR tasks are quite different from other NLP tasks, the respective adaption of the techniques is not straight-forward.

However, the distributed representations of words, phrases, sentences and documents (*Word2Vec* and *Doc2Vec*, or: neural word embeddings) tend to cluster semantically related representations. Extending on this property, the distance between those vectors could provide a reasonable similarity metric for the scoring step in IR. Therefore, it is of interest to investigate, how well these neural word embeddings perform in practical information retrieval tasks, especially considering the ranking of the results.

2 Related work

Traditional retrieval models (*tf-idf*, *BM25*) rely on bag-of-words representations. While still considered a strong baseline, these models (along with others) struggle with two typical difficulties of the IR task: *term dependencies* and *vocabulary mismatch*. The former means that the independence assumption of terms does not hold in natural language, the latter describes the problem of disregarding semantically related terms, when exact matching fails. Early approaches to tackle the term dependency problem involved word n-gram models [6]. However, Fagan et al. showed that these approaches are not so successful, most probably caused by higher sparsity of the more complex units. [10].

In 2012, Bengio et al. first introduced a statistical language model [4], based on neural networks, so-called neural net language models [2], which form the basis for More recently, Mikolov et al. [9, 8] proposed a neural network based word embedding (*Word2Vec*), in which the representations are learned by training to reconstruct each word's context (skip-gram model) or to reconstruct a word given its context (continuous bag-of-words). The success of this model relies on very efficient training, that does not involve dense matrix multiplication. Le and Mikolov [7] further extend this approach by additionally modeling representations of whole documents (*Doc2Vec*). Their experiments indicate that these distributed representations are useful for information retrieval tasks. However, the conducted information retrieval task differs from practical information retrieval in several aspects:

- The conducted information retrieval task in the experiments is narrowed down to a binary classification task (matching), disregarding any quality of the obtained ranking (scoring).
- In this binary classification task, the models are given 80% supervised training data at their disposal, **which is typically not available in practice**.
- In practice,the binary decision about relevancy is less important than the actual scoring of the relevant documents.

Hence, it is left to investigate how well the retrieved documents can be ranked (scoring step) with respect to the meaningful distance to the query in the representational space.

3 Goals

3.1 Evaluating the Ranking Quality of Neural Word Embeddings

In order to inspect the ranking quality of neural word embeddings, an experiment will be conducted, in which the quality of the obtained rankings is evaluated. Since queries to practical retrieval engines are rather short, or even just single word, it is still questionable, how *Doc2Vec* *Word2Vec* perform in terms of ranking quality (measured in nDCG and MAP) compared to each other and *tf-idf* as a baseline. For evaluation we will conduct some of the well-known TREC data sets.

3.2 Full-text vs Titles-Only

We will answer the question of how good titles can be when compared to full-text using neural word embeddings. One could assume, that a carefully trained neural word embeddings could be capable of circumventing the lack of full-text by capturing enough semantics of the presumably quite important words occurring in the titles. For the experiments consult a data set consisting of 288344 scientific publications from the economics domain and split it up into titles-only and full-text data sets. The gold-standard is computed by querying the full-text *tf-idf* model, and the different retrieval models are evaluated against this artificial gold-standard using nDCG and MAP metrics.

3.3 Integrating Neural Word Embeddings in an IR Framework

In order to accomplish the previous two objectives, neural word embeddings need to be integrated in a practical information retrieval framework.

On the one hand, *Word2Vec* is a pure unsupervised model, that learns to reconstruct the word's context. Apart from some hyper-parameters, such as layer-size and window-size, no additional information about labels is required.

On the other hand, *Doc2Vec* is designed to associate given labels with arbitrary documents, so that in this case, we need a set of labels that may be used to describe a document corpus, for which we will use the information about concepts in a prevalent thesaurus. It is left to investigate, whether it is computationally affordable to use all of these concepts or to make a cut-off at a certain level of occurrence counts.

In order to make neural word embeddings applicable in practice, we have to consider the analyzing process, the matching operation, and the similarity scoring. Typical information retrieval engines are organized as follows:

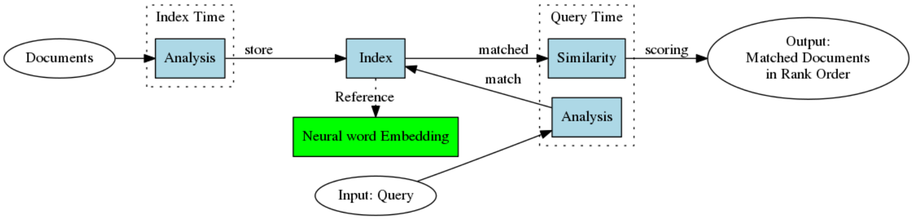

Fig. 1. Architecture of a typical retrieval engine (blue) and possible extensions (green)

To incorporate neural word embeddings in this generic information retrieval architecture, the plugin will provide an analyzer to shift the raw texts into representation space (possibly by reference to the respective word vector) and several similarity modules to provide a scoring function with respect to the distance in representational space of *Word2Vec* and *Doc2Vec*.

Training and Analysis

True incremental (online) training would require to change the whole matrix on each out-of-vocabulary word, which is unfeasible for a practical information retrieval engine, because of computational expenses. Thus, the plugin will provide some of the following alternatives:

- A method to consult a pre-trained model (or store a trained model).
- A REST endpoint to train a model on the whole indexed corpus. This operation would be separated from both, the index time and the query time.
- An analyzer for incremental training after one of the initial setup variants above, but ignoring out-of-vocabulary words.

The proposed methodology will allow the user to either work with a completely precomputed model, such as Google's *Word2Vec* model, or to train a model after indexing some corpus, and refining it while indexing more documents. After training the model, the words occurring in a raw text can be shifted into representational space by a custom analyzer, which will be provided by the plugin.

Similarity Scoring

The plugin will also feature certain similaritiy metrics to score the retrieved documents based on their distance to the query. While *Doc2Vec* natively provides a similarity metric between documents, the most prominent approach for emphWord2Vec is to apply word averaging over documents and queries.

3.4 Semantic Search (Optional)

The *Word2Vec* model allows to perform meaningful mathematical operations on the word's representations. This feature could also be safely passed down to the user in terms of a dedicated query language.

For example, the query 'economic - crisis' would trigger the actual mathematical computation vec('economics') – vec('crisis'), and retrieve documents as if the query was composed of the k nearest neighbors of the computation's result (in the vector space). In our example, we would expect to see results about growing economies without any crisis.

4 Schedule

Table 1. Time Management

Task	**Time Span**
Documentation and Paper	2017-10-01 – 2017-03-30 (6 months)
Implementation	2016-10-01 – 2016-12-31 (3 months)
Experiments and Evaluation	2016-01-01 – 2017-02-28 (2 months)

5 Frameworks

Desired Solution: Native Elasticsearch Plugin

As Elasticsearch (built on top of Lucene) is currently one of the most popular information retrieval frameworks, a native Elasticsearch plugin would be the desired solution. However, due to certain limitations of both, Elasticsearch and Deeplearning4J, the integration might become difficult and inefficient.

- Deeplearning4J
- Elasticsearch
- Lucene

Alternative Solution: The Pythonic Approach
In contrast to Deeplearning4J, the Deep Learning frameworks in python are very flexible and extensible. Therefore the addition of new methods might require to drop Deeplearning4J in favor of either Keras or Tensorflow. The python bindings of PyLucene to Lucene would retain compatibility with existing retrieval engines, while it is still possible to switch to a pure python retrieval framework if necessary.

- PyLucene or Whoosh
- Tensorflow or Keras

6 Preliminary Outline

6.1 Paper

Inspecting the Ranking Quality of Word2Vec and Doc2Vec in Practical Information Retrieval on Titles-Only and Full-Texts

1. Introduction
2. Related Work
3. Neural Word Embeddings
  3.1. *Word2Vec*
  3.2. *Doc2Vec*
  3.3. ...
4. Experimental Setup
  4.1. Task
  4.2. Data Sets
  4.3. Evaluation
5. Results
  5.1. *Word2Vec* vs *Doc2Vec*
  5.2. Full-Text vs Titles-Only
6. Discussion
7. Conclusion

6.2 Documentation

Neural Word Embeddings in Practical Information Retrieval

References

[1] Georgios Balikas and Massih-Reza Amini. “An empirical study on large scale text classification with skip-gram embeddings”. In: *CoRR* abs/1606.06623 (2016).

[2] Yoshua Bengio. “Neural net language models”. In: *Scholarpedia* 3.1 (2008), p. 3881.

[3] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. “Representation Learning: A Review and New Perspectives”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 35.8 (2013), pp. 1798–1828.

[4] Yoshua Bengio et al. “A Neural Probabilistic Language Model”. In: *Journal of Machine Learning Research* 3 (2003), pp. 1137–1155.

[5] Ronan Collobert et al. “Natural Language Processing (Almost) from Scratch”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2493–2537.

[6] Joel L. Fagan. “Automatic Phrase Indexing for Document Retrieval: An Examination of Syntactic and Non-Syntactic Methods”. In: *Proceedings of the Tenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, USA, June 3-5, 1987.* 1987, pp. 91–101.

[7] Quoc V. Le and Tomas Mikolov. “Distributed Representations of Sentences and Documents”. In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014.* 2014, pp. 1188–1196.

[8] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.* 2013, pp. 3111–3119.

[9] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *CoRR* abs/1301.3781 (2013).

[10] ChengXiang Zhai. “Statistical Language Models for Information Retrieval: A Critical Review”. In: *Foundations and Trends in Information Retrieval* 2.3 (2008), pp. 137–213.

September 7, 2016

____

Lukas Galke

______________________________	______________________________
Prof. Dr. Ansgar Scherp	Ahmed Saleh