# Document Embedding for Scientific Articles: Validation of word embeddings

Arjan Meijer, 11425555

May 7, 2018

## 1 Abstract

[TODO: Abstract]

# 2 Background

**Embedding**
Machine learning (ML) tasks rely on a numerical (vectorial) representation of text which we refer to as an embedding. These can be calculated for texts of different lengths such as a title, sentence, paragraph or an entire document[1]. Word embeddings are these numerical representations of a word, these vectors are an distributed representation of the word over the multiple (vector) dimensions(Mikolov et al. [2]). The word embeddings can be used to construct embedding of larger texts. Word embeddings can capture both the semantic and syntactic information of words. The advantage of the machine learning models is that it can be done without human-interaction(Lai et al. [3]). Word embeddings have improved various Natural Language Processing (NLP) areas such as named entity recognition, part-of-speech tagging, parsing, and semantic role labelling (Luong et al. [4]).

*Word2Vec*
Word2vec word embeddings are created using neural network, Word2vec learns word embeddings via maximizing the log conditional probability of the word given the context word(s) occurring within a fixed-sized window. Therefore the learnt embeddings contain useful knowledge about word co-occurrence[5]. There are multiple input/output possibilities for the neural network, best known are Skip-gram and the Continuous Bag-of-Words model (CBOW). The Skip-gram model takes a target word as input and outputs the predicted output words, while CBOW takes the context words as input and outputs the predicted target word[5, 6].

*Paragraph vectors*
Variations on the word2vec model have also been proposed, Le and Mikolov [7] introduced the paragraph vector, based on the word2vec model. The paragraph vector model uses additional variables to improve the accuracy of the word-embeddings. An advantage of the paragraph vector model is that it takes the word order into consideration, atleast in a small context [7].

*GloVe*
Pennington et al. [6] introduced the GloVe (Global Vectors) model. This model captures the global corpus statistics. The model transforms the word co-occurrences of all words in the corpus to chances, it excludes all the zero values and uses that as initial input for the neural network.

*TF-IDF*
TF-IDF is an abbreviation for Term Frequency - Inversed Document Frequency. This method does not rely on a neural network, and does not require training. According to a paper by Beel et al. [8] from 2016, "TF-IDF was the most popular weighting scheme (70%) among those approaches for which the scheme was specified" (in the recommendation class 'Content-based filtering'). The TFIDF score is the product of the term frequency in a text and the inversed document frequency of the same term in a corpus of texts. Both of which can be calculated in a variety of ways.

| | | | |
|---|---|---|---|
| Man | Women | King | Queen |
| Athens | Greece | Oslo | Norway |
| great | greater | tough | tougher |

Table 1: Word analogies used in word embedding validation

**Embedding validation**
Embedding validation techniques are methods that are used to validate the quality[1] of an embedding for a specific task(Schnabel et al. [9]). Multiple validations of word embeddings have been used, including: Word analogy, text similarity, categorization and positional visualization.

*Word Analogy*
Word analogy validation is based on a labelled validation set, containing, commonly, word pairs of four, that can be logically divided into two parts. As Table 1 shows, each last word can be derived from the three words before. The score is the fraction of correctly given fourth words, given the first three words. This validation metric is used in multiple studies[2, 6, 10, 11]. Both this validation technique and the Word Similarity technique use vector distance calculations to validate the embeddings, this can therefore also be written as:

$$X_{Man} - X_{King} \approx X_{Women} - X_{Queen}$$

*Word Similarity*
A method to test the quality of word embeddings is the word similarity test. For these test, the distance between the word embeddings (vectors) is measured and compared to similarity scores defined by humans. Multiple non domain specific validation sets are publicly available including: the Rare-word dataset introduced in the paper "Better Word Representations with Recursive Neural Networks for Morphology" by Luong et al. [4], the MEN test collection by Bruni [12] and the WordSimilarity-353 test collection by Gabrilovich [13]. These sets, among others, have been used in multiple studies of word embeddings[6, 10]. This validation metric also relies on labelled data.

*Classification*
The classification validation method is a simple task which compares multiple texts. Lau and Baldwin [14] used data from StackExchange and tried to determine if a pair was a duplicate. Even though the validation method is simple, it too used labelled data to validate the acquired results.

*Categorization*
Le and Mikolov [7] used for their research a dataset of IMDB with 100,000 movie review. They validated their proposed paragraph vector model by determining whether a review was positive or negative.

*Position Visualization*
(Unlabelled, Needs human validation)Dai et al. [11] and Hinton and Roweis [15] mapped their word embeddings to a two dimensional vector to be able to display them in a graph

[1]With quality we mean the extend to which the task is completed correctly

and applied colors to various categories. The advantage of this is that a human can directly see that the embeddings make sense, however this approach is not applicable by a computer.

Even though these validation methods are not limited to domains, the labelled data they use are, since they do not consist of words of specific domains. At this moment, there are no sets for every domain which make it difficult to compare the accuracy of domain specific word embeddings to non domain specific word embeddings.

# 3 Motivation

**Domain specific**
Earlier work on this topic, concerning domain specific articles, by (Truong [16]), found that in-domain training of the word embeddings can drastically improve the process of document clustering. In fact, the effect is even stronger than the number of training examples and the model architecture. However, too isolated training can lead to a failure of several clustering algorithms, such as DBSCAN, due to a too dense vocabulary[16]. Lai et al. [3] found that the corpus domain is more important than the corpus size. Using an in-domain corpus significantly improves the performance for a given task, whereas using a corpus in an unsuitable domain may decrease performance. These findings both indicate that an in-domain corpus improves the performance of word embeddings for the specific domains.

| | WordSim | Men | RareWords |
|---|---|---|---|
| Lowest | 0.38 | 0.54 | 0.29 |
| Highest | 0.49 | 0.61 | 0.32 |
| Average | 0.45 | 0.59 | 0.32 |
| Baseline | **0.64** | **0.68** | **0.34** |

Table 2: Results for the different validation sets of word simularity validations on domain specific texts from the study by Truong [16]

**Problems in validation**

Unpublished results of the study by Truong encounter this problem, they show high error rates on the validation scores, presented in Table 2. However, the word embeddings created correct document clusterings[16], this seems to indicate that the word-vectors are able to represent the words correctly but that the available validation sets cannot confirm this.

Furthermore, a study by Schnabel et al. [9] found that the quality of embeddings are tasks specific, *different tasks favour different embeddings.* They also found that the embeddings encode information about word frequency, even in models that are created to prevent this. *This casts doubt on the common practice of using vanilla cosine similarity as a similarity measure.*

Therefore, we propose the validation of domain specific word-embeddings through a classification tasks, using multiple vector-distance calculations. This eliminates the need for labelled data in the validation of these domain specific word embeddings, will validate the quality of word embeddings for domain specific texts, and will validate the impact of different vector-distance measures on a categorization task.

# References


[1] R. Karimi. Go deep or go out. Technical report, Elsevier, 2017.

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

[3] Siwei Lai, Kang Liu, Shizhu He, and Jun Zhao. How to generate a good word embedding. *IEEE Intelligent Systems*, 31(6):5–14, 2016.

[4] Thang Luong, Richard Socher, and Christopher Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, 2013.

[5] Eric Nalisnick, Bhaskar Mitra, Nick Craswell, and Rich Caruana. Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 83–84. International World Wide Web Conferences Steering Committee, 2016.

[6] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[7] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[8] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4):305–338, 2016.

[9] Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, 2015.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[11] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.

[12] Elia Bruni. Men test collection, 2012. URL `https://staff.fnwi.uva.nl/e.bruni/MEN`.

[13] Evgeniy Gabrilovich. Wordsimilarity-353 test collection, 2002. URL `http://www.cs.technion.ac.il/ gabr/resources/data/wordsim353/`.

[14] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.

[15] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 857–864, 2003.

[16] J. Truong. An evaluation of the word mover's distance and the centroid method in the problem of document clustering, 2017.