Document Embedding for Scientific Articles: Validation of word embeddings

Arjan Meijer, 11425555

April 10, 2018

1 Abstract

[TODO: Abstract]

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

Table 1: Word analogies used in word embedding validation

Embedding
ML tasks rely on a numerical (vectorial) representation of text which we refer to as an embedding. This embedding 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, which, since the representation is a vector, is an distributed word representation over multiple (vector) dimensions(Mikolov et al. [2]). Word embeddings can capture both the semantic and syntactic information of words from a large unlabeled corpus (Lai et al. [3]). Techniques based on word embeddings have improved various NLP areas such as named entity recognition, part-of-speech tagging, parsing, and semantic role labelling (Luong et al. [4]). The word2vec model is a way of creating word embeddings. The model converts words via a learned lookuptable into real valued vectors[5]. Mikolov et al. [5] show that calculations with these vectors is possible:

$$\mathrm{X}_{\mathrm{apple}} - \mathrm{X}_{\mathrm{apples}} \approx \mathrm{X}_{\mathrm{car}} - \mathrm{X}_{\mathrm{cars}}$$

Furthermore they show that de distance in the vector space between "king" and "man" approximates the distance between "queen" and "women". Variations on the word2vec model have also been proposed, Le and Mikolov [6] 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 [6]. For this research, only word2vec will be used.

Domain specific
Earlier work on this topic, concerning domain specific articles, by (Truong [7]), 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[7]. 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.

Embedding validation
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, 8, 9, 10].

Word Similarity

	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 [7]

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 [11] and the WordSimilarity-353 test collection by Gabrilovich [12]. These sets, among others, have been used in multiple studies of word embeddings[8, 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 [13] 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 [6] 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. [9] and Hinton and Roweis [14] mapped their word embeddings to a two dimensional vector to be able to display them in a graph 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 this validation method is not limited to domains, the validation sets are, since they do not contain 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. 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[7], this seems to indicate that the word-vectors are able to represent the words correctly. The problem is then that the available validation sets cannot confirm this. Therefore, we propose the validation of word-embeddings through [TODO: SOLUTION - classification - abstract/text/title matching - keyword categorization - ...] to eliminate the need for labelled data in the validation of word embeddings. This will enable us to find optimal parameters for the embedding of domain specific articles.

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] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.

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

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

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

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

[10] 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.

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

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

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

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