Do Scaling Algorithms Preserve Word2Vec Semantics? A Case Study for Medical Entities

Janus Wawrzinek1[0000-0002-8733-2037], José María González Pinto1[0000-0002-2908-3466],
Philipp Markiewka1[0000-0003-2099-7568] and Wolf-Tilo Balke1[0000-0002-5443-1215]

[1] IFIS TU-Braunschweig, Mühlenpfordstrasse 23, 38106 Braunschweig, Germany
{wawrzinek, pinto, balke}@ifis.cs.tu-bs.de,
p.markiewka@tu-braunschweig.de

Abstract. The exponential increase of scientific publications in the bio-medical field challenges access to scientific information, which primarily is encoded by *semantic relationships* between medical *entities,* such as active ingredients, diseases, or genes. Neural language models, such as Word2Vec, offer new ways of automatically learning semantically meaningful entity relationships even from large text corpora. They offer high scalability and deliver better accuracy than comparable approaches. Still, first the models have to be tuned by testing different training parameters. Arguably, the most critical parameter is the *number of training dimensions* for the neural network training and testing individually different numbers of dimensions is time-consuming. It usually takes hours or even days per training iteration on large corpora. In this paper we show a more efficient way to determine the optimal number of dimensions concerning quality measures such as precision/recall. We show that the quality of results gained using simpler and easier to compute scaling approaches like MDS or PCA correlates strongly with the expected quality when using the same number of Word2Vec training dimensions. This has even more impact if after initial Word2Vec training only a limited number of entities and their respective relations are of interest.

Keywords: information extraction, neural language models, scaling approaches.

1 Introduction

The current exponential growth of scientific publications in the medical field requires innovative methods to structure the information space for important medical entities, such as active ingredients, diseases, genes, and their relationships to each other. For instance, a term-based search for a common disease such as diabetes in the medical digital library PubMed leads to a search result of over 600,000 publications. Here the automated extraction of high-quality relationships between entities contained in medical literature would provide a useful tool to facilitate an exploration of large datasets. Moreover, such an extraction could serve as a basis for numerous innovative medical applications such as Drug Repurposing [3, 7], the discovery of drug-drug interactions

[4], the creation of biomedical databases [5], and many more. Previous work has recognized this and proposed several methods to calculate similarities between entities to infer their relationships. These include popular approaches such as the computation of chemical (sub-)structure similarity based on bit-fingerprints [8] or methods relying on entity networks [6]. Recent approaches even try to calculate similarity based on word contexts using distributional semantic models (DSMs) [1, 9-11]: here, a similar word context points to an implicitly expressed relationship. This property is often transferred to entities: two entities in similar linguistic contexts point to an intrinsic relationship between these entities and possibly also to similar entity properties. According to Baroni et al. [12], DSMs can generally be divided into count-models and predictive models. For count-models, first word-context matrices are generated from a text corpus, followed by matrix optimization steps such as re-weighting and dimensional scaling [12]. In contrast, predictive models (also known as embedding models or neural language models) try to predict an expected context based on numerous training examples. Studies show that state-of-the-art predictive models, such as Word2Vec, outperform count models in performance and scalability, in particular in semantics and analogy tasks [12, 13].

Recently researchers [28-30] have tried to uncover the theoretical principles of Word2Vec to reveal what is behind the embedding vectors' semantics. In particular, the work of [30] has demonstrated that a reformulation of the objective of the skip-gram negative sampling implementation (SGNS) of Word2Vec leads to a mathematical demonstration that SGNS is, in fact, an explicit matrix factorization, where the matrix to be factorized is the co-occurrence matrix. However, little is known about the effect of scaling algorithms on Word2Vec: do we lose its appealing semantics, or do we filter out noise [19]? Among the popular scaling algorithms that exist, which one can preserve the original semantics better? Does it make a difference which scaling algorithm is chosen? Answering these questions can help researchers to find the optimal number of dimensions of semantic spaces efficiently. In fact, the usually accepted '200-400 dimensions' chosen when training Word2Vec (see e.g., [13-14]) has yet to spark a more in-depth investigation.

In this paper, we pragmatically investigate these questions to provide first insights into the fundamental issues. We focus on a case study for medical entities motivated by our findings in previous work. In [1] we investigated the semantic properties of Word2Vec for pharmaceutical entity-relations and subsequently utilized them as an alternative access path for the pharmaceutical digital library PubPharm[1]. In brief, we found that semantically meaningful drug-to-drug relations are indeed reflected in the high-dimensional word embeddings. Here, we aim to identify the effect of scaling methods such as Multidimensional Scaling (MDS) and Principal Component Analysis (PCA) on active substance embeddings learned by Word2Vec.

In the following, we show that scaling has a high correlation with the number of Word2Vec training dimensions. This finding means that by using scaling, we can find where the optimal number of training dimensions regarding purity, precision, and recall is located. Our results can be of interest for all approaches in which Word-Embedding

[1] https://www.pubpharm.de/vufind/

training has to be applied to massive amounts of data (Big Data) and thus exploring different numbers of dimensions with re-training is not a practical option.

The paper is organized as follows: Section 2 revisits related work accompanied by our extensive investigation of scaling approaches to embedded drug-clusters in section 3. We close with conclusions in section 4.

2 Related Work

Neural Language Model Representation of Words. Semantic embeddings of words in vector spaces have sparked interest, especially Word2Vec [13-14] and related methods [15-18]. Researchers have demonstrated that words with similar meanings are embedded nearby, and even 'word arithmetic' can be convincingly applied. For example, the calculated difference in the embedding vector space between 'Berlin' and 'Germany' is similar to the one obtained between 'Paris' and 'France'. Word2Vec representations are learned in an unsupervised manner from large corpora and are not explicitly constrained to abide by such regularities. In a nutshell, Word2Vec is a technique for building a neural network that maps words to real number vectors. What is unique about these number vectors is that words with similar meaning will map to similar vectors. At its core, Word2Vec constructs a log-linear classification network. More specifically, in [13-14] researchers proposed two such networks: the Skip-gram and the Continuous Bag-of-Words (CBoW). In our experiments we used the Skip-gram architecture, which is considered preferable according to the experiments reported by [14].

Multidimensional Scaling (MDS). Multidimensional Scaling [19] deals with the problem of representing a set of n objects in a low-dimensional space in which the distances respect the distances in the original high-dimensional space. In its classical formalization MDS takes as input a dissimilarity matrix between pairs of objects and outputs a coordinate matrix whose configuration minimizes a loss function called stress or strain [19]. In our experimental setting, given a matrix of the Euclidean distances between entities represented by Word2Vec vectors, $M = [ed_{i,j}]$ where $ed_{i,j}$ is the distance between the pair of entitiesi, j. MDS uses eigenvalue decomposition on the matrix M using double centering [20]. In our experiments we used the Scikit-Learn Python implementation [21] with default parameters except for the number of dimensions that we exhaustively tested.

Principal Component Analysis (PCA). Principal Component Analysis is a popular data mining technique for dimensionality reduction [27]. Given a set of data points on n dimensions, PCA aims to find a linear subspace of dimension d lower than n such that the data points lie mainly on this linear subspace. In our case we take the matrix M_e of Word2Vec vectors where the rows represent medical entities and columns to the dimensions of the Word2Vec semantic space. The idea of PCA then is to treat the set of tuples in this matrix and find the eigenvectors for $M_e {M_e}^T$. When you apply this transformation to the original data, the axis corresponding to the principal eigenvector is the one along which the points are most spread out. In other words, this axis is the one along which the variance of the data is maximized. Thus, the original data is approximated by data that has many fewer dimensions and that summarizes well the original data.

Orthogonal Procrustes. We use Orthogonal Procrustes [25] – also known as rotational alignment – to evaluate the relative quality of two different scaling approaches. The general idea here is to evaluate two scaling techniques without considering any specific metric related to the clustering task. Instead it is assessed by measuring pointwise differences, which of the two scaling approaches can better approximate the original Word2Vec space. Orthogonal Procrustes was used before to align word embeddings created at different time periods, i.e., to analyze semantic changes of words in diachronic embeddings [23].

3 Investigation of Effect of the Dimensionality Reduction

First, we describe the methodology for generating our ground truth dataset. After this, we describe our ground truth evaluation corpus followed by experimental set-up and implementation decisions. Then we examine with the help of our ground truth dataset whether the number of Word2Vec training dimensions and the number of scaling dimensions correlate with purity, precision, recall, and F-Score. We will then perform a mathematical analysis between MDS, PCA, and Word2Vec results based on statistical t-test and matrix approximation methods. Afterwards we compare the runtime of MDS, PCA and the training with different Word2Vec dimensions. Since our current study is based on the dataset of our previous work [1], we use almost the same methodology, evaluation corpus, implementation, and set-up decisions:

Methodology for building our ground-truth dataset

After the initial crawling step the following process can be roughly divided into four sub-steps:

1. *Preprocessing of crawled documents.* After the relevant documents were crawled, classical IR-style text pre-processing is needed, i.e., stop-word removal and stemming. The pre-processing helps mainly to reduce vocabulary size, which leads to improved performance, as well as improved accuracy. Due to their low discriminating power, all words occurring in more than 50% of the documents are removed. Primarily, these are often used words in general texts such as '*the*' or '*and*', as well as terms frequently used within a domain (as expressed by the document base), e.g., '*experiment*', '*molecule'*, or '*cell*' in biology. Stemming further reduces the vocabulary size by unifying all flections of terms. A variety of stemmers for different applications is readily available.
2. *Creating word embeddings for entity contextualization.* Currently, word embeddings [12] are the state-of-the-art neural language model technique to map terms into a multi-dimensional space (usually about 200-400 dimensions are created), such that terms sharing the same context are grouped more closely. According to the distributional hypothesis, terms which often share the same context in larger samples of language data, in general also share similar semantics (i.e., have a similar meaning).

In this sense, word embeddings group entities sharing the same context and thus collecting the nearest embeddings of some search entity leads to a group of entities sharing similar semantics.

3. *Filtering according to entity types.* The computed word embeddings comprise at this point a significant portion of the corpus vocabulary. For each vocabulary term there is precisely one-word vector representation as the output of the previous step. Each vector representation starts with the term followed by individual values for each dimension. In contrast, classical facets only display information of the same type, such as publication venues, (co-)authors, or related entities like genes or enzymes. Thus, for the actual building of facets, we need only vector representations of the same entity type. Here, dictionaries are needed to sort through the vocabulary for each type of entity separately. The dictionaries either can be directly gained from domain ontologies, like, e.g., MeSH for illnesses, can be identified by named entity recognizers, like e.g., the Open Source Chemistry Analysis Routines (OSCAR, see [26]) for chemical entities, or can be extracted from open collections in the domain, like the DrugBank for drugs.

4. *Clustering entity vector representations.* The last step is preparing the actual grouping of entities closely related to each other. To do this, we apply a k-means clustering technique on all embedded drug representations and decide for optimal cluster sizes: in our approach optimal cluster sizes are decided according to the *Anatomical Therapeutic Chemical (ATC) Classification System*[2]. Here ATC subdivides drugs according to their anatomical properties, therapeutic uses, and chemical features.

Experimental ground-truth dataset setup.

Evaluation corpus. With more than 27 million document citations, PubMed[3] is the largest and most comprehensive digital library in the bio-medical field. However, since many documents citations do not feature full texts, we relied solely on abstracts for learning purposes. As an intuition, the number of abstracts matching each pharmaceutical entity under consideration should be ‘high enough’ because with more training data, more accurate contexts can be learned, yet the computational complexity grows. Thus, we decided to use the 1000 most relevant abstracts for each entity according to the relevance weighting of PubMed’s search engine [31].

Query Entities. As query entities for the evaluation, we randomly selected 275 drugs[4] from the *DrugBank*[5] collection. We ensured that each selected drug featured at least one class label in ATC and occurred in at least 1000 abstracts on PubMed. Thus, our final document set for evaluation contained 275,000 abstracts. Therefore, these drugs usually have a *one-word* name, which makes it straightforward to filter them out after a Word2Vec training iteration. However, besides our specific case, pharmaceutical entities often consist of several words (e.g., diabetes mellitus) and can also have many

[2] https://www.whocc.no/atc_ddd_index/

[3] https://www.ncbi.nlm.nih.gov/pubmed/

[4] The complete list can be downloaded under: http://www.ifis.cs.tu-bs.de/webfm_send/2295

[5] https://www.drugbank.ca/

synonyms (e.g., aspirin/ acetylsalicylic acid). Phrases and synonyms are a general problem for word embedding algorithms because they are usually trained on single words, resulting in one vector per word and not per entity. A possible solution for such cases is 1) applying named entity recognition in documents and 2) placing a unique identifier at the entity's position in the text. Here, entity recognition can be done using PubTator[6], which is a tool that can recognize pharmaceutical entities as well as their position in text and that returns a unique MeSH-Id for each of them.

As ground truth, all class labels were crawled from *DrugBank*. Since the ATC classification system shows a fine-grained hierarchical structure, we remove all finer levels before assigning the respective class label to each drug. For example, one of the ATC classes for the drug 'Acyclovir' is '*D06BB53*'. The first letter indicates the main anatomical group, where '*D*' stands for 'dermatological'. The next level consists of two digits '*06*' expressing the therapeutic subgroup 'antibiotics and chemotherapeutics for dermatological use'. Each further level classifies the object even more precisely, until the finest level usually uniquely identifies a drug. In our active ingredient collection there are 13 different ATC class labels of the highest level. We use these 13 different labels to divide the 275 active ingredients into 13 (*ground truth*) clusters.

Ground Truth dataset implementation and parameter settings.

1. *Text Preprocessing:* Stemming and stop-word removal were performed using a *Lucene*[7] index. For stemming we used Lucene's *Porter Stemmer* implementation.
2. *Word Embeddings*: After preprocessing, word embeddings were created with Gensims's *Word2Vec*[8]-implementation. To train the neural network, we used a minimum word frequency of 5 occurrences. We set the word window size to 20 and the initial layer size to 275 features per word. Training iterations were set to 4.
3. *Entity filtering.* While Word2Vec generated a comprehensive list of word vector representations, we subsequently filtered out all vectors not related to any DrugBank entity (resulting in 275 entity-vectors).
4. *Clustering vector representations*. In this step we clustered the 275 entity vector representations obtained in the previous filtering step in 13 clusters. For the clustering step we used Python [21] Multi-KMean ++ implementation.

3.1 Experimental Investigation

First, we need to clarify how a correlation between the different approaches can be measured. We also need to determine whether the scaling approaches are faster. In this context, the following quality criteria should be fulfilled:

- *Empirical Correlation accuracy*: The result of a scaling approach should be comparable to the result of a Word2Vec training for a fixed number of training dimensions. Therefore, we will always determine the 'semantic quality' of a semantic space by

[6] https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/PubTator/
[7] https://lucene.apache.org/
[8] https://radimrehurek.com/gensim/models/word2vec.html

evaluating purity, F-Score, precision, and recall against the ground truth expressed by the ATC classification. After scaling down the original Word2Vec space trained on 275 dimensions to *n* dimensions (where *n* < 275) the semantic quality of this space needs to be compared to the respective quality of a Word2Vec space directly trained using only *n* dimensions. Are the respective qualities correlated for different values of *n*?

- *Mathematical Accuracy:* The result of a scaling should resemble the vectors of a Word2Vec training. A similarity between the vectors would underpin the results of our empirical study as well as help us to find possible differences between PCA and MDS. To test our hypothesis, we perform a mathematical analysis based on statistical t-test and matrix approximation using orthogonal Procrustes.
- *Scaling performance*: Performing a scaling iteration for some number of dimensions should on average be significantly faster than training a Word2Vec model using the same number of dimensions.

3.2 Empirical correlation accuracy

In our first experiment we investigate if scaling with MDS and PCA correlates with the number of Word2Vec training dimensions regarding the following quality measures: F-Score, precision, recall, and purity. We determine the quality measures for our clusters using the method described in Manning et al. [22]. Initially we train Word2Vec using 275 dimensions, and we choose the maximum of 275 dimensions because of the technical implication for calculating PCA. Technically speaking there exist a Principal Component for each variable in the data set. However, if there are fewer samples than variables, then the number of samples puts an upper bound on the number of Principal Components with eigenvalues greater than zero [27]. Therefore, for this experiment, we perform the following steps for each number of dimensions *n* (where *n* < 275):

- *Scaling Step*: First we scale the initially trained and filtered 275 active substance Word2Vec vectors with dimensions *n* using MDS and PCA. Also, we train *Word2Vec* with *n* dimensions on the evaluation corpus and then filter out the 275 active substance vectors.
- *Clustering Step*: For each of the three results from the previous step, we assign each active ingredient to one of the possible 13 ATC class labels. Then we perform clustering with *k*=13 and a total of 50 iterations. In each clustering iteration we calculate the quality measures mentioned above and calculate the mean values for purity, precision, recall, and F-Score.

Figure 1 shows the respective mean values regarding each quality measure for the different choices of dimensions. Table 1 lists the correlation (Pearson correlation coefficient) values between the different methods. The mean values of the individual dimensions were used for the correlation calculation. As can be seen, there is a strong correlation for all values, whereby the values for MDS correlate best with the Word2Vec result. Thus, scaling approaches indeed lead to similar results as Word2Vec training.

Table 1. Correlation (Pearson correlation coefficient) values between the different approaches. Where PCC is the correlation coefficient between precision values, RCC between recall values, F1CC between F1-Values, and PuCC is the correlation coefficient between purity values.

Correlation between	PCC	RCC	F1CC	PuCC
MDS-W2V	0.90	0.80	0.85	0.87
PCA-W2V	0.85	0.78	0.81	0.69

Can the optimal training dimension be determined using a scaling method? As can be seen, the highest mean values (Figure 1) of the different methods are almost precisely in the same dimension range (e.g., precision). This observation allows us to predict the optimum number of training dimensions quite accurately using scaling approaches. *Is the quality comparable?* Surprisingly, a Word2Vec training does not always lead to the best result. For example, we can observe that scaling for most dimensions (~200) leads to a better result. In particular, we achieve the best purity-value with PCA. In short, it probably pays off to use a scaling approach. The differences for the other quality measures are rather small. For example, MDS can only achieve a ~2% worse precision result, but on the other hand, MDS scaling alone can increase the precision values by up to ~60%, and F1-Values up to 20%. What we can also see is that our optimum for all quality measures lies at about 25 dimensions. This value, in turn, deviates quite far from the recommended 200-400 dimensions for a Word2Vec training. Our finding indicates that for a particular problem domain, as in our case, a standard choice of dimensions for a Word2Vec training can be a disadvantage.

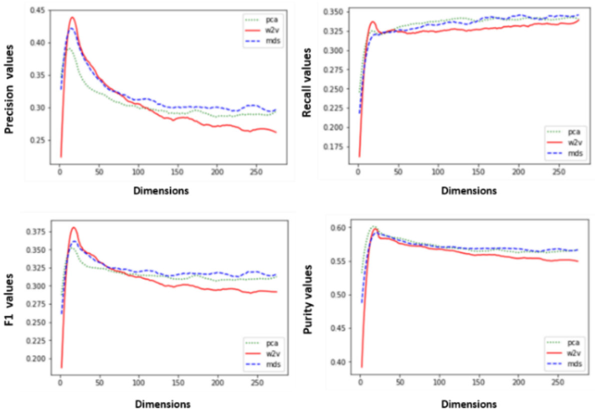

Fig. 1. Precision, recall, F1, and purity mean values for PCA, MDS, and Word2Vec.

3.3 Mathematical accuracy

We performed two evaluations to assess the quality of the scaling approaches that we compared with Word2Vec. The first evaluation corresponds to what we called metric-based analysis because specific metrics that depend on the task at hands such as precision, recall, and F1 are needed. In contrast, non-metric based evaluation considers only the approximation quality of the scaling algorithms regarding the original Word2Vec space.

Metric-based analysis. In this first evaluation we used precision, recall, and F-Score to perform a pair-wise t-test comparison. With a 95% confidence interval the differences between PCA and MDS are not statistically significant for precision and F-Score. However, in recall, the differences between PCA and MDS are statistically significant.

To provide the reader with a visual interpretation of the results found in the recall t-test, we show in Figures 3, 4, and 5 the Bland-Altman Plots [24] that compares MDS with PCA, MDS with Word2Vec, and PCA with Word2Vec, respectively. Bland-Altman plots compare in a simple plot two measurements to ascertain if indeed differences exist between them. In the x-axis the graph shows the mean of the measurements and on the y-axis their differences. Thus, if there are no differences, we should observe on the y-axis that most of the values are near zero. This type of plot makes it easier to see if there are magnitude effects, for example when small values of x and y are more similar than large values of x and y. We can observe in that differences between PCA and Word2Vec are negligible regarding recall values. Moreover, we can observe that the higher the values of recall, the better PCA is in approximating Word2Vec. In summary, the plot (Fig. 5) shows that PCA leads to a slightly better approximation of recall values than MDS.

Non-metric based analysis. Finally to evaluate the differences between MDS and PCA, we decided to assess the approximation power of the two methods using Procrustes analysis. This analysis complements our previous metric-based analysis by introducing an evaluation of the MDS and PCA spaces regarding how good each of them can approximate the original Word2Vec space. What we mean here by Procrustes analysis is the following: given two identical sized matrices, Procrustes tries to apply transformations (scaling, rotation and reflection) on the second matrix to minimize the sum of squares of the pointwise differences between the two matrices (disparity hereafter). Put another way, Procrustes tries to find out what transformation of the second matrix can better approximate the first matrix. The output of the algorithm is not only the transformation of the second matrix that best approximates matrix one but also the disparity between them.

We use the disparity value in our analysis to determine which of the scaling algorithms can better approximate Word2Vec original space. Low disparity values are better by definition. In Figure 2 we plot the disparity values using dimensions up to 275 which is the maximum number that we can use because we have only 275 active substances as our input matrix. To generate the plot, we train Word2Vec for dimensions two up to 275. We used the original space of 275 dimensions from Word2Vec to apply MDS and PCA using dimension from 2 up to 275. Thus, each point in the plot shows the disparity value between the corresponding scaling algorithm and Word2Vec. We can see that PCA outperforms MDS because it shows lower disparity values for each of the dimensions calculated. In other words, PCA preserves the quality of the semantics of the original Word2Vec space better than MDS.

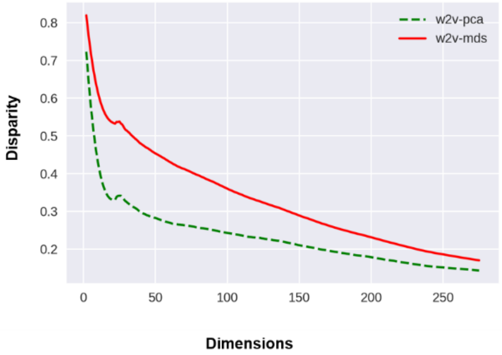

Fig. 2. Disparity Comparison (lower values are better) between Word2Vec-PCA and Word2Vec-MDS using Procrustes Analysis.

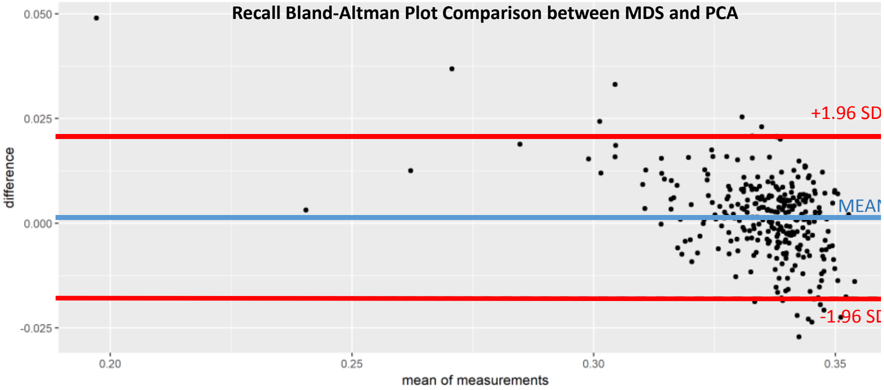

Fig. 3. Bland-Altman plot using Recall measures PCA vs MDS.

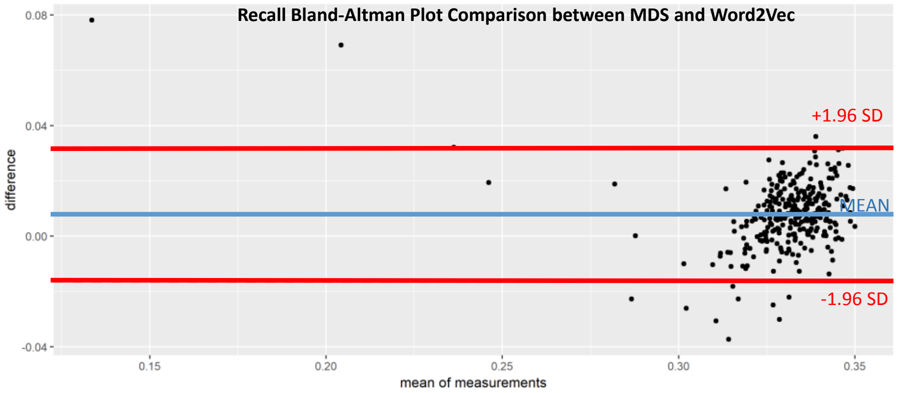

Fig. 4. Bland-Altman plot using Recall measures MDS vsWord2Vec.

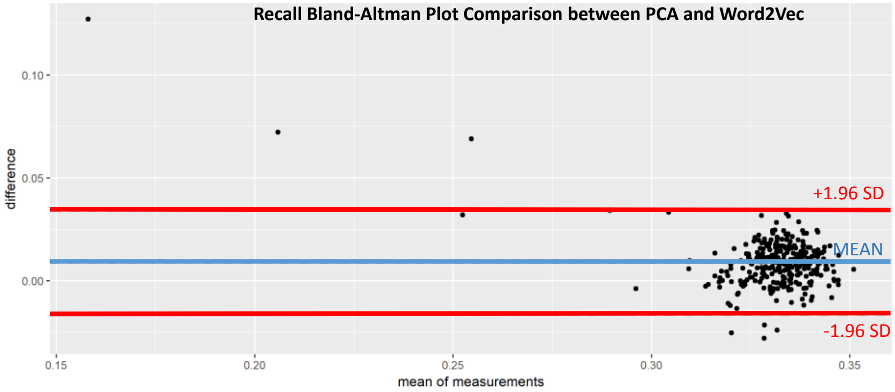

Fig. 5. Bland-Altman plot using Recall measures PCA vs Word2Vec.

3.4 Scaling Performance

After having shown that there is both a strong empirical as well as a robust mathematical correlation between scaling approaches and a Word2Vec training using the same number of dimensions, we then compare the runtime performance of the different approaches. Here, we first train Word2Vec on our ground truth corpus with 275 dimensions and extract the 275 active substances vectors again. Then we scale the result with PCA and MDS to dimensions n (where $n < 275$), after which we measure the cumulative time which was required for scaling to all number of dimensions. Also, we train Word2Vec with the different number of dimensions n ($n < 275$) and measure the cumulative training time for comparison with the scaling approaches. This kind of Word2Vec training corresponds to the usual procedure to determine an optimal result (e.g., regarding F-Score). All three calculations are performed one after the other on the same computer with the following characteristics: 16 Xeon E5/Core i7 Processors with 377 GB of RAM. The results of our experiments are shown in Table 2:

Table 2. Runtime *(seconds)*: Sum of the runtimes of the different approaches in seconds. *Runtime reduction*: Reduction in % of run times compared to a Word2Vec (W2V) training

Approach	Runtime (seconds)	Runtime reduction
PCA	17	99.83%
MDS	1162	88.22%
W2V	9865	–

As can be seen in Table 2, scaling approaches need significantly less time on our active substance dataset. Here, a runtime reduction of up to 99% can be achieved. PCA was much faster in scaling compared to MDS. Given the observed runtime reduction, it pays off to use scaling approaches when training on a large corpus.

4 Conclusions

We have conducted an experimental analysis of scaling algorithms applied over a set of entities using neural language models for clustering purposes. Indeed, one of the most critical parameters of implementations such as Word2Vec is the number of training dimensions for the neural network. Because different testing numbers are time-consuming and thus can take hours or even days per training iteration on large text corpora, we have investigated an alternative using scaling approaches. In particular, we used the implementation provided by Word2Vec and contrasted Multidimensional Scaling and Principal Component Analysis quality. We conclude here by summarizing our main findings for researchers and practitioners looking to use Word2Vec in similar problems.

Our experiments indicate that there exists a strong correlation (up to 90%) regarding purity, F1, as well as precision and recall. We have shown that for a particular problem domain, as in our active substance case, a standard choice of dimensions for a Word2Vec training can be a disadvantage. Moreover, by mathematical analysis we have shown that the spaces after scaling strongly resemble the original Word2Vec semantic spaces. Indeed, the quality of the scaling approaches is quite comparable to the original Word2Vec space: they achieve almost the same precision, recall, and F1 measures.

As a performance bonus, we have shown that performance of scaling approaches regarding execution times is several orders of magnitude superior to Word2Vec training. For instance, we obtained more than 99% of time-saving when computing PCA instead of Word2Vec training. Researchers could thus rely on initial Word2Vec training or pre-trained (Big Data) models such as those available for the PubMed[9] corpus or

[9] https://github.com/RaRe-Technologies/gensim-data/issues/28

Google News[10] with high numbers of dimensions and afterward apply scaling approaches to quickly find the optimal number of dimensions for any task at hand.

References

1. Wawrzinek, J., & Balke, W. T. (2017, November). Semantic Facettation in Pharmaceutical Collections Using Deep Learning for Active Substance Contextualization. In *International Conference on Asian Digital Libraries* (pp. 41-53). Springer, Cham.
2. Nikfarjam, A., Sarker, A., O'Connor, K., Ginn, R., & Gonzalez, G. (2015). Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, *22*(3), 671-681.
3. Wang, Z. Y., & Zhang, H. Y. (2013). Rational drug repositioning by medical genetics. *Nature biotechnology*, *31*(12), 1080.
4. Abdelaziz, I., Fokoue, A., Hassanzadeh, O., Zhang, P., & Sadoghi, M. (2017). Large-scale structural and textual similarity-based mining of knowledge graph to predict drug–drug interactions. *Web Semantics: Science, Services and Agents on the World Wide Web*, *44*, 104-117.
5. Leser, U., & Hakenberg, J. (2005). What makes a gene name? Named entity recognition in the biomedical literature. *Briefings in bioinformatics*, *6*(4), 357-369.
6. Lotfi Shahreza, M., Ghadiri, N., Mousavi, S. R., Varshosaz, J., & Green, J. R. (2017). A review of network-based approaches to drug repositioning. *Briefings in bioinformatics*, bbx017.
7. Dudley, J. T., Deshpande, T., & Butte, A. J. (2011). Exploiting drug–disease relationships for computational drug repositioning. *Briefings in bioinformatics*, *12*(4), 303-311.
8. Willett, P., Barnard, J. M., & Downs, G. M. (1998). Chemical similarity searching. *Journal of chemical information and computer sciences*, *38*(6), 983-996.
9. Abdelaziz, I., Fokoue, A., Hassanzadeh, O., Zhang, P., & Sadoghi, M. (2017). Large-scale structural and textual similarity-based mining of knowledge graph to predict drug–drug interactions. *Web Semantics: Science, Services and Agents on the World Wide Web*, *44*, 104-117.
10. Ngo, D. L., Yamamoto, N., Tran, V. A., Nguyen, N. G., Phan, D., Lumbanraja, F. R., & Satou, K. (2016). Application of word embedding to drug repositioning. *Journal of Biomedical Science and Engineering*, *9*(01), 7.
11. Lengerich, B. J., Maas, A. L., & Potts, C. (2017). Retrofitting Distributional Embeddings to Knowledge Graphs with Functional Relations. *arXiv preprint arXiv:1708.00112*.
12. Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vol. 1, pp. 238-247).
13. Mikolov, T., Yih, W. T., & Zweig, G. (2013). 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* (pp. 746-751).
14. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *NIPS*. (2013).

[10] https://code.google.com/archive/p/word2vec/

15. Levy, O., Goldberg, Y.: Neural Word Embedding as Implicit Matrix Factorization. *Adv. Neural Inf. Process. Syst*. 2177–2185 (2014).
16. Pennington, J., Socher, R., Manning, C.: Glove: Global Vectors for Word Representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 1532–1543 (2014).
17. Bengio, Y., Courville, A., Vincent, P., Collobert, R., Weston, J., et al.: Natural Language Processing (almost) from Scratch. *IEEE Trans. Pattern Anal. Mach. Intell*. 35, 384–394 (2014).
18. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of Tricks for Efficient Text Classification. 2, 427–431 (2016). *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain, April 3-7, 2017.
19. Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media. (2005).
20. Weinberg, S.L.: An introduction to multidimensional scaling. Meas. Eval. Couns. Dev. 24, 12–36 (1991).
21. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., et al.: Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011).
22. Manning, C. D., Raghavan, P., Schütze, H. (2008). *Introduction to information retrieval.* Cambridge University Press.
23. Hamilton, W.L., Leskovec, J., Jurafsky, D.*: Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* pages 1489–1501, Berlin, Germany, August 7-12, 2016 (2016).
24. Altman, D.G., Bland, J.M.: Measurement in Medicine : the Analysis of Method Comparison Studies. Statistician. 32, 307–317 (1983).
25. Schönemann, P.H.: A generalized solution of the orthogonal procrustes problem. *Psychometrika*. 31, 1–10 (1966).
26. Jessop, D.M., Adams, S.E., Willighagen, E.L., Hawizy, L., Murray-Rust, P. (2011) OSCAR4: a flexible architecture for chemical text-mining. *Journal of Cheminformatics*, Vol. 3(1), Springer.
27. Leskovec, J., Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. Cambridge *University Press* (2014).
28. Levy, O., Goldberg, Y.: Neural Word Embedding as Implicit Matrix Factorization. *Adv. Neural Inf. Process. Syst.* 2177–2185 (2014).
29. Gittens, A., Achlioptas, D., Mahoney, M.W.: Skip-Gram - Zipf + Uniform = Vector Additivity. *Proc. 55th Annu. Meet. Assoc. for Comput. Linguist*. (Volume 1 Long Pap. 69–76 (2017).
30. Li, Y., Xu, L., Tian, F., Jiang, L., Zhong, X., Chen, E.: Word embedding revisited: A new representation learning and explicit matrix factorization perspective. *IJCAI Int. Jt. Conf. Artif. Intell*. 2015–January, 3650–3656 (2015).
31. Canese, K. (2013). PubMed relevance sort. *NLM Tech. Bull*, *394*, e2.