Mining Semantic Subspaces to Express Discipline-Specific Similarities

Janus Wawrzinek
Institute for Information Systems
Technische Universität Braunschweig
Braunschweig, Germany
wawrzinek@ifis.cs.tu-bs.de

José María González Pinto
Institute for Information Systems
Technische Universität Braunschweig
Braunschweig, Germany
pinto@ifis.cs.tu-bs.de

Wolf-Tilo Balke
Institute for Information Systems
Technische Universität Braunschweig
Braunschweig, Germany
balke@ifis.cs.tu-bs.de

ABSTRACT

Word embeddings enable state-of-the-art NLP workflows in important tasks including semantic similarity matching, NER, question answering, and document classification. Recently also the biomedical field started to use word embeddings to provide new access paths for a better understanding of pharmaceutical entities and their relationships, as well as to predict certain chemical properties. The central idea is to gain access to knowledge *embedded, but not explicated* in biomedical literature. However, a core challenge is the *interpretability* of the underlying embeddings model. Previous work has attempted to interpret the semantics of dimensions in word embeddings models to ease model interpretation when applied to semantic similarity task. To do so, the original embedding space is transformed to a sparse or a more condensed space, which then has to be interpreted in an exploratory (and hence time-consuming) fashion. However, little has been done to assess in real-time whether specific *user-provided semantics* are actually reflected in the original embedding space. We solve this problem by extracting a semantic subspace from large embedding spaces that better fits the query semantics defined by a user. Our method builds on least-angle regression to rank dimensions according to given semantics properly, i.e. to uncover a subspace to ease both *interpretation* and *exploration* of the embedding space. We compare our methodology to querying the original space as well as to several other recent approaches and show that our method consistently outperforms all competitors.

CCS CONCEPTS

• Information systems~Data mining • Information systems~Content analysis and feature selection • Computing methodologies~Learning latent representations

JCDL '20, August 1–5, 2020, Virtual Event, China

ACM ISBN 978-1-4503-7585-6/20/06$15.00
https://doi.org/10.1145/3383583.3398523

KEYWORDS

Word embeddings, Semantic subspaces, Pharmaceutical entities, Medical digital libraries, Neural language models

ACM Reference format:

Janus Wawrzinek, José M. G. Pinto, and Wolf-Tilo Balke. 2020. Mining Semantic Subspaces to Express Discipline-Specific Similarities. In *Proceedings of ACM/IEEE-CS* Joint Conference on Digital Libraries *(JCDL'20). ACM, New York, NY, USA, 10 pages.*

1 INTRODUCTION

Today word embeddings [5, 31-33] are an often used natural language processing (NLP) technique in biomedicine [34-37]. Consider for instance Word2Vec [5, 6, 22], a state-of-the-art neural-language model based on the idea of embedding words into some high-dimensional space based on their surrounding word contexts. The distance between two words in the embedded space is smaller the more similar word contexts they share. This contextualization property is of great interest for the biomedical domain [23], where entities like active substances, diseases, genes, and their interrelations play an essential role. The distance between embedded biomedical entities expresses a certain similarity according to some predominant semantics as provided by the training corpus, e.g., a therapeutic similarity between active substances [17]. Moreover, based on such lexical similarities, also predictions of complex chemical properties are possible [23]. This makes word embeddings interesting for important downstream tasks such as drug repositioning [17, 41], medical hypotheses generation [16], disease diagnosis [18], and many others. However, when taking large and diverse corpora (such as Medline) for training, entities are bound to appear in many *different* word contexts. For example, active ingredients may have chemical, therapeutic, pharmacological, or anatomical relationships or a combination of these. This raises the question: *which semantics do entities actually have in the embedding space?*

Based on this interpretation problem and the growing popularity of NLMs like Word2Vec, newer approaches try to shed light on this semantic black box. On one hand, research focuses on general relationships between the corpus and the semantic quality of models [19, 20], where similarity analysis is mainly applied to the entire feature space. On the other hand different semantics might exist in different subspaces [1, 21], which would

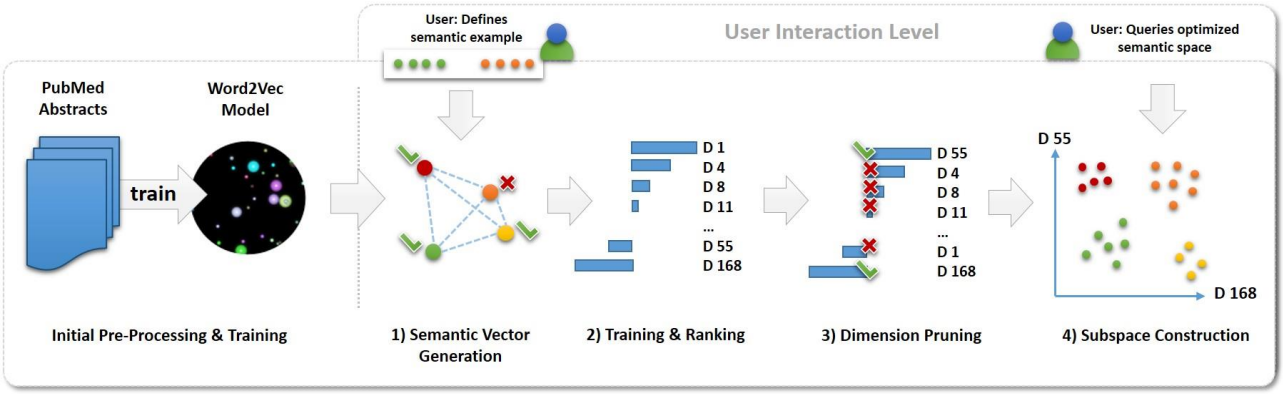

Figure 1. Method overview: we start with a pre-trained model, and then we apply our approach. First, it optimizes the user-input and generates a semantic vector for training and ranking the dimensions. After that, it prunes the dimensions and finally it generates an interpretable semantic subspace.

then be preferable for query processing. The general problem in this context is that in a d-dimensional embedding space $2^d - 1$ possible subspaces exist: the search for the best subspace fulfilling specific semantic properties is therefore prohibitive.

Due to these problems, recent works [1, 21] try to create a semantically changed (sub-) space W_s from the original word embedding space W. Generally speaking, the model learns a transformation matrix using categories defined by experts. Afterwards, this transformation matrix is applied to create an optimized (sub-) space from the original embedding space. Such approaches, thus, require a large amount of expert knowledge per semantics, i.e. a transformation without expert knowledge is not possible, and semantic information may be lost [1, 21]. On the other hand, unsupervised methods exist, such as subspace clustering [38-40] or node embedding approaches [24, 25], which have the potential to assign several different contexts to an entity. Here, however, the semantics in the resulting clusters and subspaces are unknown and thus need to be labelled afterwards. Furthermore, the number of contexts is fixed for all entities, which is difficult without a detailed knowledge of all aspects of the training corpus. Here a user-driven approach would be helpful to choose desired semantics.

In this paper, we develop a method that allows the user to express desirable semantics for active ingredient similarity search at query time. Based on this input, users query an automatically generated embedding subspace, which corresponds best to the given semantics, i.e., a therapeutic subspace where active substances are grouped according to their similar therapeutic properties (Figure 1). To summarize, the contributions of this paper are as follows:

- We introduce a method that given examples exhibiting certain semantics, automatically discovers a sub-space that expresses semantic similarity pursued by the query. Our method empowers users, since it does not require a predefined semantics of the embeddings space (Sections 2 and 3).
- We prove that our method can ease the interpretation of embedding spaces with substantial improvements in similarity tasks (Section 5).

2 FUNDAMENTALS

In this section, we first explain and define the relationships between different (semantic) similarities that exists between a pair of pharmaceutical entities. Finally, we define the problem finding a semantic subspace followed by a measure to calculate a mean squared error for a given subspace.

2.1 Relationship between Word-embedding and Semantic Similarity-Measures

The semantic subspace searched for, expresses a given semantic, which can be measured by similarity measures. In this context, given a set of n entities $e_1, \dots, e_n \in E$ and a set of representations for these entities, such as (but not limited to):

- $fingerprint_{F_m}(e_i) \coloneqq (f_1, \dots, f_m) \in \{0,1\}^m$, i.e. bit vector for e_i according to some fingerprint definition F_m (i.e. [43])
- $embedding_{W_k}(e_i) \coloneqq (w_1, \dots, w_k) \in \{0,1\}^k$, i.e. word vector v of a word embedding matrix W_k
- $taxonomic\_class_{T_l}(e_i) \coloneqq (t_1, \dots, t_l) \in \{0,1\}^l$, i.e. the set of classes entity e_i belongs to in taxonomy T_l

then the similarity between two entities e_i and e_j can be expressed by different similarity measures $sim(e_i, e_j)$ working on their respective representations and evaluating different features like e.g. Euclidean distances, cosine similarities, overlap of taxonomic classes, etc.

Please note that depending on the representation and the similarity measure used, the similarity value between the same two entities may strongly vary. For instance, two active ingredients may be to treat the same illness, i.e. show a high therapeutic similarity, while being very different, i.e. dissimilar, on their chemical structure. A similarity with respect to a certain representation is a proper subspace similarity, if the similarity measure is evaluated on a projection $\pi(\cdot)$ of the original representation, i.e. a subset of dimensions for vector-based representations or a subset of classes for taxonomical representations, etc. where

the cardinality of dimensions, classes, etc. is strictly smaller than in the original representations.

2.2 Problem Definition

Now, the problem of semantically focused subspace selection for word embedding spaces can be framed as: Given a similarity measure $sim(e_i, e_j)$ for $e_i, e_j \in E$, find a projection $\pi(\cdot)$ on the k-dimensional word embedding space W_k so that $|\pi(W_k)| \ll k$ and a similarity measure of entity pairs in word embedding space W_k is proportional to $sim(e_i, e_j)$ with a suitable quality threshold:

$$\frac{1}{n(n-1)} \sum_{e_i,e_j \in E} (sim_{\pi(W_k)} - sim(e_i, e_j))^2 \leq t \quad (*)$$

i.e. the mean squared error (MSE) is bounded by threshold t. Since this problem does not have always valid solutions, we approach the related problem of "best effort" semantic focused selection where the error term (*) is minimized. The resulting projected subspace can be subsequently used for best effort similarity query processing following the semantics expressed by $sim(e_i, e_j)$.

3 METHODOLOGY

Our method is based on regression analysis that has long been used in the field of chemometrics [14] to reveal (linear) relationships between the dependent variable (target variable) and one or multiple independent variables (predictors). For regression analysis and feature selection it is helpful to use dependent variables whose semantics/relationships are well-understood [14]. Moreover, it is extremely beneficial to identify and exclude factors that have little or no influence on predicting dependent variables, because this can lead to the optimization of the accuracy of a regression model [15]. We apply these ideas to our problem to determine an optimized semantic subspace with fewer dimensions. In this context, the Multiple Linear Regression Model describes the target variable y through the linear line (matrix notation):

$$y = X\beta + \varepsilon$$

where y represents a vector of observations y_i $(1 \leq i \leq n)$, X represents a matrix with the predictor variables in the columns, β represents a $(n+1)$-dimensional vector with regression coefficients, and ε represents an n-dimensional vector with the error terms. According to our problem, we adjust the notation for the following steps:

$$\mathrm{y_s} = \mathrm{W}\beta + \varepsilon$$

Here we call y_s the semantic vector because it expresses a user-specified semantics between entities. In our specific case y_s is an n-dimensional vector, where each entry of y_s contains the relative position of some entity e_v. On the other hand, we have as input the word-embedding matrix W whose dimensions represent the independent variables. After an initial document preprocessing and training step, in which the result is the word-embedding matrix W, we can divide our approach into four steps (Figure 1). In the first step, we prepare the user-input for the following dimension-learning and ranking step. Afterward, we remove dimensions that do probably not contain the preferred semantics. In the last step, we construct the semantic subspace and prepare it for k-Nearest-Neighbor queries. In the next sections, we will describe each of the steps in detail.

3.1 *Semantic Vector Generation*

The user input can be seen as a one-dimensional space i.e. the user *places* the entities on a line (user-interface) relative to each other. Here, the user expresses a certain semantic between the entities, i.e., semantically similar entities are positioned close to each other while semantically dissimilar entities are separated as far as possible. These positions have nothing to do with the positions (values) in the word-embedding matrix W. If the values of the observations represented by y_s have nothing in common with the values of the predictors (matrix W), then our approach will probably not be able to find a connection between y_s. and W. We will prove this in chapter 5.1 (Influence of the semantic vector). To determine the entity positions for y_s we use two steps. First, we identify a dimension from W for which the MSE between user input and the dimension is lowest. Since we measure in one-dimensional spaces, we use the Euclidean distance for this task. This dimension reflects the user input nearest. After this step we remove this dimension from W and select it as the vector y_s. In the second step we optimize the user input by identifying the k-entities that cause the largest MSE and remove them from y_s. Formally, this procedure can be defined as follows.

Sampling set-up: Given two similarity measures $sim_A(e_i, e_j)$ and $sim_B(e_i, e_j)$, (in practice one of them reflects a similarity on a word embedding space, while the other reflects the semantics desired by some user) let us define a fully connected undirected graph K_n over the entity set E as nodes:

$$V = \{e_{v_1}, e_{v_2}, \dots, e_{v_n}\}$$

Then we label each edge between each two nodes e_{v_i}, e_{v_j} by the squared error $|sim_A(e_{vi}, e_{vj}) - sim_B(e_{vi}, e_{vj})|^2$ between both similarities.

Sampling: To select a sample of entities for the respective semantics, we proceed as follows:

1. From the word embedding matrix W, a dimension is chosen in which the MSE is minimal, this is our y_s first.

2. To lower the MSE in the selected dimension further, we first remove all edges that do not fulfill the condition $|sim_A(e_{vi}, e_{vj}) - sim_B(e_{vi}, e_{vj})|^2 \leq \lambda$. Then iteratively k entities (including edges) are removed which have the smallest number of remaining edges to other entities. The remaining $n - k$ entities that appear most often together form the semantic vector y_s.

We use this method to construct the semantic vectors in our experiments, and we use the parameter λ to regulate how many entities should be selected for y_s. We demonstrate the efficiency of this graph-pruning in our evaluation (section 5.1).

3.2 *Dimension Ranking*

Considering the learning objective of NLMs like Word2Vec [5], we do not assume an independence of the dimensions in the matrix W. Furthermore, we assume that values of some dimensions correlate linearly stronger with y_s than others, and thus have probably a stronger correlation with the semantics defined in y_s. We hypothesized that the predictors, therefore dimensions that have a linear correlation with the entity positions in y_s, indicate the semantic subspace W_s. We will prove this assumption later in our experiments (section 5.1). In order not to have the problem to try $2^d - 1$ combinations arbitrarily to calculate an optimized space, we need a ranking of the dimensions by correlation strength with y_s. This ranking enables us to decide in which order we can combine the dimensions. Furthermore, we have to keep in mind that the number of rows can significantly exceed the number of columns. This can hurt many regression approaches, such as the inaccuracy of the model [15]. In order to achieve such a ranking of dimensions under the given conditions, we use Least Angle Regression (LAR). The LAR algorithm is an efficient version of forward stepwise regression and consists of the following steps [11, 42]:

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $r = y - \hat{y},\ \beta_1, \beta_2, \dots, \beta_m = 0$
2. Find the predictor x_j most correlated with r.
3. Move β_j from 0 towards its least-squares coefficient $\langle x_j, r \rangle$, until some other competitor x_k has as much correlation with the current residual as does x_j
4. Move β_j and β_k in the direction defined by their joint least squares of the current residual on (x_j, x_k), until some other Competitor x_l has as much correlation with the current residual.
5. Continue in this way until all m predictors have been entered. After $min(N - 1, m)$ steps, we arrive at the full least-squares solution.

Conclusions: LAR reduces the residual error in each step. The dependent variable $\hat{y}$ is the semantic vector y_s, whose relative entity positions are to be predicted. Therefore, the reduction of the residual error should correlate with the Mean Squared Semantic Error (*MSE, Section 2.2)*. Besides LAR determines which predictors (dimensions) have the strongest correlation with y_s. After this step, the predictors can be listed according to their absolute coefficient value. The lower the value, the smaller the influence of a predictor in predicting the dependent variable [15]. We combine LAR with Cross-Validation and in Lasso-Mode, as this can lead to an optimized result [10].

3.3 *Dimension Pruning*

Next, we have to decide how many dimensions to choose for the subspace. On the one hand, Anscombee's Quartet [9] shows that different sets of data points can have almost the same statistical properties as, e.g., the same linear regression line, but their position in space can be completely different. However, the position affects the semantic quality and the *MSE* in the subspace. On the other side, a LAR ranking alone will probably not be expressive enough, because we can assume that a low number of dimensions (i.e., only one dimension) leads to a loss of semantic information. On the other hand, if the coefficients of some dimensions are near to zero or set to zero, then these dimensions probably lead to an increase of noise [15] and therefore probably to lower semantic subspace quality. We confirm this assumption in section 5.1. In this context, we need an approach that approximates an optimal number of dimensions in acceptable query runtime. We decided to approximate the optimal number of dimensions as follows:

1. Remove all predictors $x_i\ (1, \dots, n)$ where $\beta_i = 0$.
2. Rank all remaining predictors x_i according to their absolute coefficient value $|\beta_i|$, where $|\beta_i| > 0$.
3. Calculate the standard deviation sd of all coefficients.
4. Take all predictors x_i where $|\beta_i| \geq \alpha * sd, \alpha \in R$

The remaining predictors form the dynamic generated semantic subspace W_s.

3.4 *Subspace Construction*

To compute the MSE, we determine the similarity values for all pairs in y_s. This can be computed fast because y_s contains only a small set of entities. To provide for the user *k*-Nearest-Neighbor queries in a dynamically generated subspace, we need to be able to calculate the k-Neighbors at query time [12] and this for thousands of entities efficiently. For this task, and in contrast to the MSE for computing y_s, a pre-computation of all pairwise similarities is not strictly necessary. Therefore, we use for this task the k-dimensional tree algorithm, which shows excellent performance compared to other approaches [13].

4 EXPERIMENTAL SET-UP

In this section, we will first describe our pharmaceutical text corpus and the necessary experimental set-up decisions followed by implementation details.

4.1 Document Corpus and Query Entities

Evaluation corpus. For our document corpus, we used all abstracts from the bio-medical digital library *PubMed*[1]. Specifically, we used all annotated abstracts from *PubTator*[2]. In this context, PubTator applies named entity recognition to identify pharmaceutical entities such as drugs, diseases and genes in PubMed publications uniquely. We used the annotated documents because word-embedding algorithms generally create an embedding per word and not per entity. However, this is a problem for drugs consisting of several words (e.g. "acetylsalicylic acid"). In addition to a unique entity-id, PubTator also provides the exact position of the entity in the text. We used this information and

[1] https://www.ncbi.nlm.nih.gov/pubmed/
[2] https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/PubTator/

inserted the entity id into the text to get/learn an entity embedding.

Query Entities. As query entities for the evaluation, we selected all *approved* drugs from the *DrugBank*[3] collection, which can also be found in the documents as well have a MeSH-Id listed in the Comparative Toxogenomics Database (CTD[4]) Database. Thus, our final entity set for evaluation contains 1124 approved active substances. In total, we can identify ~75.000 chemicals in PubMed. After our core method evaluation, we will use these 75.000 entities to show that our method can scale to that amount of entities (Section 5.4).

4.2 Experimental Implementation and Parameter Settings

In the following, we describe experimental implementation details and parameter settings.

Text Pre-processing. To improve the embedding-quality and to reduce the training time, we first removed stop-words and applied stemming (*Lucene's*[5] *Porter Stemmer*) afterwards.

Word-Embeddings. We used DeepLearning4J's *Word2Vec*[6] skip-gram implementation to train the word-embeddings. We set the window-size to 50, as a larger window size can lead to improved results in learning (pharmaceutical) associations [20, 34]. Further, according to [34] we set the layer size to 200 features per word, and we removed all words with a lower word frequency than five occurrences.

Word-Embedding (WE) Similarity-Measure. As the similarity measure between the drug-entity embedding vectors, we choose Euclidian based similarity in all experiments. We decided for Euclidean based similarity because the user input is a one-dimensional space. We rescale and rearrange the distance so that a value of 1 between two vectors means a perfect similarity and the value 0 means a maximum distance between two vectors.

LAR-Training. We use Scikit's[7] implementation of LAR with Cross-Validation and in Lasso-Mode. Here, we use the default settings.

4.3 Semantic Similarity Measures

To simulate a (pharmaceutical) user input, we have decided to test our approach with three different main used drug semantics (pharmacologic, therapeutic, and anatomical). Thus, we will group active substances according to pharmacologic, therapeutic and anatomical properties. The computation of these semantics is primarily based on active substance similarity. In the following, we describe how we compute the different similarities, which we use in our experiments, using taxonomical information.

Active Substance Similarity. The *taxonomical similarity* approach to compute active ingredient similarity is based on mostly manually curated semantic classification systems. Considering pharmacy, the Anatomical Therapeutic Chemical (ATC) Classification System[8] is one of the most used classification systems. ATC first groups drugs according to anatomical properties (at level 1), then to therapeutic (level 2), and then to therapeutic-pharmacologic (level 3) properties. If drugs are in the same group and at the same level, they are to some degree similar. We use this classification to calculate the different similarities. For example, to calculate the anatomical similarity, we first generate a drug-drug adjacency matrix $A = [a_{i,j}]$ and set the values as follows:

$$a_{i,j} \begin{cases} 1 \; if \; drug \; i \; is \; in \; the \; same \; group \; as \; drug \; j, \\ 0 \; else. \end{cases}$$

After that, we determine the anatomical similarity by measure the similarity between two drug-vectors using a similarity measure (e.g. cosine similarity). We repeat this approach, using different group levels, to measure therapeutic and pharmacologic similarity.

5 EXPERIMENTAL EVALUATION

For the experimental evaluation, we first have to determine what quality criteria our proposed method should meet to be useful for dynamically creating and querying semantic subspaces. In this context, the following quality criteria have to be fulfilled. In our evaluation, we will investigate each of the defined quality criteria.

- **Subspace-Quality**: First, we have to confirm the hypothesis that the LAR rank correlates with the subspace quality, and thus an LAR rank enables us to generate an optimized semantic subspace. A group of dimensions with a lower rank should lead to a lower subspace quality than the dimensions of a higher rank. The quality can be measured by determining whether semantically similar active substances are more likely to be grouped in a subspace.
- **Subspace Diversity and Stability**: The identified semantic subspaces should be pairwise different as possible; therefore, a high amount of different dimensions per semantic should be found. The differences can be determined using the Jaccard coefficient. On the other side, *Subspace Stability* indicates that probably no much better subspaces exist and that the desired semantics are actually found in the subspace. Therefore, the subspace should be stable when using different entities for y_s. For a given semantics and varying y_s, the majority of equal/similar combinations of dimensions should always be found so that:

$$y_s \sim \acute{y_s} \iff W_s \sim \acute{W_s}$$

We use the Jaccard coefficient to measure how many identical dimensions occur in different subspaces.

[3] https://www.drugbank.ca/
[4] http://ctdbase.org/
[5] https://lucene.apache.org/
[6] https://deeplearning4j.org/word2vec
[7] https://scikit-learn.org
[8] https://www.whocc.no/atc_ddd_index/

- **State-of-the-Art Approach Comparison**: We compare our subspaces with State of the Art approaches from the paper [2]. In this context, our method should lead to improved or to comparable results.
- **Subspace Construction Performance**: Given a user-defined semantic, the method should be able to construct a subspace for all existing chemical substances (~75.000) in query-time and thus in maximal few seconds.

5.1 Subspace Quality

In our first experiment, we investigate several hypotheses, which we have formulated in our method description. In this context, first we describe how we determine semantic quality in a (sub-) space. After that, we investigate whether a LAR ranking in combination with a standard deviation threshold leads to a qualitatively improved subspace and to what extent. We then investigate the influence of the semantic y_s-vector generated by our pruning method and show that our Graph-based pruning technique *sole* leads to the improved results. Finally, we prove the hypothesis that the standard deviation approach helps us to control the semantic quality. With this, we prove the hypothesis that with too few dimensions too much semantic information is removed as well as with too many dimensions noise is added. The implementation details (e.g., parameter settings) described as well as the subspaces generated here will be part of subsequent investigations.

To start our experiments, we must first clarify how to measure semantic quality for pharmaceutical entities existing in different (sub-) spaces.

Measuring Subspace Quality. To measure the semantic quality for pharmaceutical entities, we apply the clustering approach described in [29] to measure the *semantic cluster coherence* in a (sub-) space. Here, *semantic coherence* means that the entities of a cluster should be grouped under a common theme, e.g., they have similar therapeutic properties. On the other side, to compute and to compare the semantic similarity quality of an entity cluster, first, each entity (active substance) needs a unique class label. For this task, we use the ATC Classification System. *How many clusters should we expect?* Since most drugs for our evaluation are assigned to only one ATC Root-Group, there will be, in most cases a zero similarity between drugs from different ATC-Groups. Therefore, we select the number of the highest-level groups as the number of clusters we expect, which are, in our case, all 14 ATC root groups. As an example, we investigate whether acetylsalicylic acid, with the anatomical group "N", occurs in the anatomical subspace of a cluster in which most drugs have the same group "N". For the therapeutic subspace, we investigate the quality of the next ATC level (for acetylsalicylic acid = N02) in one of the 14 clusters. Finally, to compute the cluster quality, we determine the clusters Precision, Recall, and F1 Score using the method described in [28]. These measures allow us to compare the different (sub-) spaces with each other. On the other side, a minority of drugs can have several classes and this for different ATC-groups (e.g., Like acetylsalicylic acid). Since we have to determine a unique label for each active substance, we use the majority label approach described in [29] to determine a unique label for each drug.

Experimental Implementation Details. For our experiments, we generate three different y_s-vectors that represent three different user-inputs. First, we calculate an anatomical, therapeutic, and pharmacological pairwise drug similarity with the approach described in Section 4.3. Then, we apply our *Sampling Approach* (described in Section 3.1) to generate an anatomical, therapeutic and pharmacological y_s-vector. In total, we have for each semantic a different dataset which is represented by the different semantic vectors.

In our evaluation, we use half of all drug entities (with the lowest MSE) of each y_s-vector to train a subspace and the other half to evaluate the subspace. The partitioning was done to have enough training entities on the one hand and to keep the MSE in the y_s-vectors as low as possible on the other hand. We set the semantic vector threshold parameter λ to 0.0081, as this led to improved results. We set the standard deviation threshold parameter α for the optimal choice of dimensions to 0.65 for the anatomical subspace, 1.45 for the therapeutic subspace, and 1.35 for the pharmacological subspace. The above parameter settings also led to improved results. For the LAR ranking in Lasso Mode, we use the *LassoLars* implementation of the Python library *sklearn* [30]. For the clustering of the drug entities, we used the K-Means++ Clusterer of the Java library Apache-Math[9]. We perform the clustering for the expected 14-ATC clusters in total 200 times and calculate the AVG F1 score.

Table 1. Anatomic Semantic (ATC)

Space	#Dims.	F1-Score	% Imp.
Original space (W)	200	0.377	0
SemSub(>SD)	74	**0.395**	**+4%**
SemSub(<SD)	126	0.354	-4%

Table 2. Therapeutic Semantic (ATC)

Space	#Dims.	F1-Score	% Imp.
Original space(W)	200	0.286	0
SemSub(>SD)	35	**0.333**	**+16.43%**
SemSub (<SD)	165	0.279	-2.74

Table 3. Pharmacologic Semantic (ATC)

Space	#Dims.	F1-Score	% Imp.
Original space(W)	200	0.128	0
SemSub(>SD)	29	**0.150**	**+17.19%**
SemSub(<SD)	171	0.126	-1.56%

Results and Result Interpretations. In Tables 1-3, we present the F1 scores and percentage changes for the various semantics and subspaces, including the number of dimensions. W refers to the original space, *SemSubspace (> SD)* is the space gen-

[9] https://commons.apache.org/proper/commons-math/

erated by our method and the standard-deviations threshold approach.

SemSubspace (< SD) is the subspace that contains all dimensions that fall under the SD threshold and have been filtered out by our approach. First, in all clusterings, the semantic subspace generated by us shows increased values between 4% and 17% compared to the original space. As expected, the F1-Scores are decreasing, when calculating F1 on a finer ATC Level but for the same number of clusters. Besides, the subspace with the filtered dimensions is the worst in comparison. This experiment underpins our hypotheses: 1) Our method leads to improved semantic subspaces, and 2) our approach ranks dimensions containing a searched semantics higher. Since semantics are separated in our semantically "distilled" subspaces, this can lead to improvements in the interpretability of a neural-language model output. We can observe the highest percentage improvement in the therapeutic and pharmacologic subspace. This finding could indicate that the therapeutic/pharmacologic contexts are the most pronounced. This could be useful for all approaches using entity embeddings in subsequent applications, e.g., to predict new therapeutic (drug) properties as currently being tested in drug repositioning [17, 41]. In our next experiment, we investigate the role of the semantic Vector.

What influence do the semantic vector and its generation process have? To answer this question, we generate the models under two aspects and measure the Cluster-Coherence (F1-Score) if 1) Pruning is omitted and the entities are chosen randomly and 2) if we do not use the best dimension (lowest MSE) for Pruning but randomly choose a dimension with a higher MSE. For training, we used the same number of entities as in the previous experiment. We again run the clustering 200 times and measure the AVG F1 scores on the entities not used for training. We show the results of this experiment for the different semantics in Table 4.

Table 4. Semantic Vector Generation

	Pruning +Best Dim.	No Pruning +Best Dim.	No Pruning +Worse Dim.
An.SubSp.	**0.395**	0.392	0.354
Th. SubSp.	**0.333**	0.310	0.296
Ph. SubSp.	**0.150**	0.142	0.127

Results and Result Interpretation: We can observe that if we omit the pruning step, the F1-Scores per semantics decreases for all models. Furthermore, we can observe that if a dimension with a higher MSE is chosen, the F1 decreases further until the results correspond to the original space (compare with Tables 1-3) or even become worse. We conclude from this that without our core y-vector generation approach, LAR alone cannot lead to improved results.

What effect does the SD-Approach have? Finally, and for the sake of completeness, we examine the influence of the standard deviation step (see Dimension Pruning Step in section 3.3. The lower the parameter α is set, the more dimensions we select. We want to confirm the hypothesis that we achieve a semantic loss with too few and add semantic noise with too many dimensions. For this experiment, we use the settings from the first experiment but change the SD parameter. Here we test α in a range between [0, 2] and increase α by 0.05 iteratively. Again, we measure the cluster coherence AVG F1 scores. Figure 2 shows the results for the different semantics.

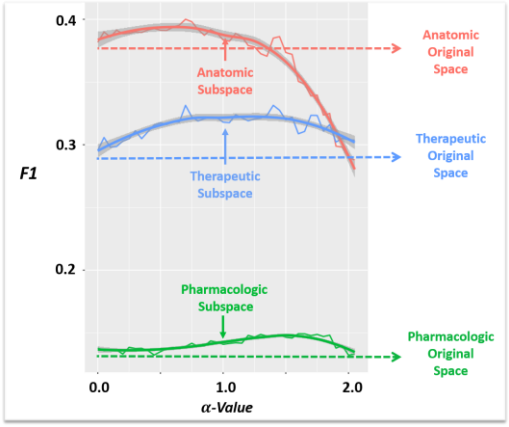

Figure 2. Results for different Standard Deviations.

Results and Results-Interpretation: Rather, less surprisingly, our hypothesis applies here. For all semantics, we see a peak in the F1 scores (Figure 2). Therefore, it is advantageous to use the SD approach; otherwise, the values can fall so far that there is hardly any difference to the original space. A good guideline is to set the parameter α to 1. This setting leads to improved results in all models.

5.2 Subspace Diversity and Stability

Have we indeed learned different semantics, or do we have some semantics that leads to improvements for other subspaces? This is an important point because if there exists no diversity, there is again the problem of interpretability. As an example, it should not be the case that a therapeutic subspace has almost the same dimensions as the anatomical subspace. On the other hand, we should expect that with similar y-vectors, we will find/learn the same dimensions, and thus we can achieve certain subspace stability.

For ***Subspace Diversity***, we compared the subspaces from the previous experiment and determined the proportion of identical dimensions using the Jaccard coefficient. Table 5 shows the pairwise Jaccard coefficients.

For ***Subspace Stability***, we removed randomly 5% of the training-entities of a y-Vector. We repeated this step 20 times to retrieve 20 different y-Vector samples. Afterwards, we learned a subspace for each y-Vector, and (using the resulting dimensions) we compared the pairwise Jaccard coefficient for each semantic. The results are presented in Table 6, where column >SD shows the results for the subspaces with the same parameter setting as for Subspace Diversity experiment and column $|x| > 0$ shows the results for all dimensions that have a coefficient value > 0.

Results and Result Interpretation. For *Subspace-Diversity*, the Jaccard coefficients show only a moderate dimension overlap. The most significant overlap exists with a Jaccard coefficient of 0.60 between the pharmaceutical and therapeutic subspace. This could be because we trained the pharmacologic semantic with ATC-level 3, which is also known as the therapeutic-pharmacological level[10]. Overall, however, the subspaces have a moderate proportion of different dimensions and, therefore, a moderate diversity. On the other side, for *Subspace-Stability* the dimension overlap is with Jaccard coefficient values up to 82 high. Based on these results, we assume that we learned different semantics and probably stable subspaces with our approach.

Table 5. Subspace Diversity (Jaccard coefficient)

	An. SubSp.	Th. SubSp.	Ph. SubSp.
An. SubSp.	1	0.48	0.40
Th. SubSp.		1	0.60
Ph. SubSp.			1

Table 6. Subspace Stability (Jaccard coefficient)

Subspace	>=SD	|x| > 0
An. SubSp.	0.76	0.82
Th. SubSp.	0.70	0.81
Ph. SubSp.	0.69	0.79

5.3 Subspace Quality Comparison

To compare the quality of the subspace, we introduce the task of entity intrusion. Similar to word intrusion [2, 26, 27] the goal is to assess the semantic coherence of a space by measuring how easy it is to identify an intruder from a set of semantically similar entities. The idea is as follows: given an entity e_l , we extract the k nearest neighbors $e_1, e_2, .., e_k$ and add an 'intruder' entity $e_{intruder}$ to the set. Because the entities except the intruder have similar meanings to each other, one can easily select the intruder to conclude that the k entities are sharing coherent meanings. How do we choose an intruder for a specific entity e_l ? Following the work of [2, 26, 27] for word intrusion, we adapt the idea to our work as follows. An intruder must satisfy two criteria: firstly, it is in the lower half of all the possible neighbors of a given entity e_l, and secondly, it is in the top 10% of some other entity e_m. Given the definition of how to find an intruder, we evaluate subspace quality using the distance ratio (DR) metric [2, 26-27].

The distance ratio is the average of the ratio between D^e_{inter} and D^e_{intra} over n entities, calculated as follows:

$$DR = \frac{1}{n}\frac{\sum_{e=1}^{n} D^e_{inter}}{\sum_{e=1}^{n} D^e_{intra}}$$

Where D^e_{intra} is the average distance between each pair among the top k neighbors of an entity e

$$D^e_{intra} = \frac{\sum_{e_i}\sum_{e_j} dist(e_i, e_j)}{k(k-1)}$$

Moreover, D^e_{inter} is the average distance between the intruder and each of the top k neighbors of entity e

$$D^e_{inter} = \frac{\sum_{e_i} dist(e_i, e_{intruder})}{k}$$

The higher the distance ratio, the better because it shows to which degree a multidimensional space is semantically coherent. In other words, an average distance between similar entities with an intruder entity should be higher than the average distance between similar entities. In our experiments, we use the cosine distance and different values of k as follows: we compute distance ratio using the k neighbors from five to 20 for a range of randomly sampled entities (five to 50) five times. To summarize the results, we average distance ratios among the sampled entities. Finally, to assess the statistical significance between the differences among methods, we used Welch's t-test (when we observed a p-value less than 0.05, we rejected the null hypothesis of equal means).

Experimental Implementation Details. We compare our approach SemSubspace to the original space (Original) and to eight (state-of-the-art) other approaches from the paper [2]. For comparison, we used the three subspaces (anatomic, therapeutic, and pharmacologic) from our first experiment (Section 5.1). In [2], researchers show that the use of rotational methods improves the interpretability of word embeddings. We use the implementation of the authors and follow the experimental settings to obtain the different rotated versions that we show in our results.

Results and Result Interpretation. In Table 7, we observe that overall our approach SemSubspace performs better than the original space and all the rotated-based approaches. The observed differences are statistically significant, based on Welch's t-test. We also observe that methods based on orthogonal rotation consistently outperform oblique rotation methods. To our surprise, all the orthogonal methods deliver a semantic coherence space with no differences between them. We can conclude

Table 7. Distance ratio for the different similarities

Approach	Th.	An.	Ph.
Original	5.802	6.109	6.667
Quartimax(orth.)	5.802	6.109	6.667
Varimax(orth.)	5.802	6.109	6.667
Parsimax(orth.)	5.802	6.109	6.667
FacParsim(orth.)	5.802	6.109	6.667
Quartimax(obl.)	2.832	3.589	3.830
Varimax(obl.)	2.832	3.590	3.611
Parsimax(obl.)	3.038	3.445	3.611
FacParsim(obl.)	3.006	3.445	3.865
SemSubspace	**6.241**	**6.445**	**7.419**

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

that our approach SemSubspace consistently extracts from the original space a semantically interpretable subspace.

In summary, our experiments prove that our approach SemSubspace delivers a specific semantic space that outperforms the original space and delivers a space that does not lose the representative power of the original space. In other words, SemSubspace finds a subset of the original space that is semantically coherent and delivers better results than the other tested approaches.

5.4 Subspace Construction Performance

In our last experiment, we investigate how much time the entire method takes to construct a semantic subspace for a user-defined query, considering all chemical substances (~75.000) found in all 29 million PubMed abstracts. We assume that the original space W already exists and a semantic subspace for a user has to be extracted. We assume that the user has a pharmaceutical interest and is mainly interested in active substances.

In the following, we will determine the performance of a user query. The machine on which we measure the time has the following characteristics: 16 Xeon E5/Core i7 Processors with 377 GB of RAM

Results: First, the user submits a sample of entities that have a semantic similarity (e.g., therapeutic) and the next step is to find the dimension that has the lowest *MSE* for the user sample (see 2.2). For this step, we select a 10% sample of active ingredients (~112) from our active ingredient entities, which should ideally overlap largely with the sample defined by the user. We then calculate the MSE in multithread mode (10 threads) and arrive at a runtime of ~0.852 milliseconds. Afterward, the semantic vector y_s is determined by our graph-pruning approach (see Section 3). For this step, we use the therapeutic entities as an example and prune so far that we get half of all entities (562) for training. This step has a runtime of 3.58 seconds. As next step, LAR determines the coefficients. For this step, we need another ~2.6 seconds to train the 562 entities with LAR in Lasso Mode. The choice of dimensions with the SD approach in the next step takes a total of ~0.03 seconds. In the last step, the subspace for all chemical substances (~75.000) is constructed and the user can query the k-Nearest-Neighbors. This step and querying a certain entity and its 10-NNs takes a total of ~1.80 seconds. Thus, in summary, we achieve a total execution time of fewer than 9 seconds. Compared to the rotational approach [2] that takes for the same number of entities and the same query a total of 900 seconds, which is 100 times slower.

6 RELATED WORK

Our work builds on the current growing interest in interpretable semantics of word embeddings. One line of research, for instance, is the work of Rothe et al. [1] that given the word embedding space, performs an orthogonal transformation of the original space using a task-specific training set. Thus, Rothe's approach called Densifier, finds a new space that is several orders of magnitude lower than the original space where one can find some specific semantics given some binary classification task, such as positive or negative sentiment, frequency of words, e.g., frequent and not frequent, and concreteness of words, e.g., concrete and no concrete. Similar to their approach, we do have as an input to our model a supervised task. However, we push further the idea by letting the user define a continuous semantic subspace instead of binary assignment of classes. Thus, we complement previous efforts that allow for more flexibility empowering the user for query-time semantics using a handful of examples. Park et al. [2] use exploratory factor analysis to word embedding spaces using basis rotation algorithms. The algorithms used in their work empirically proved to ease understanding of the dimensions of the original embedding space while retaining the semantics in several NLP tasks. After applying the rotation algorithms, one needs expert manual effort to interpret each dimension. We compare our approach to what Park et al. contributed to the community because we considered Park's approach a good baseline since it improved the original space in a word similarity task. In our case, our goal is arguable more challenging since entity similarity in the biomedical field involves specialized vocabulary and acronyms to consider.

Some representative examples of another line of work that modifies the original word embeddings algorithms aiming at a better interpretation of the dimensions are the works of Luo et al. [3] and Murphy [4]. Murphy applies non-negative constraints to the word vectors learned from a matrix-factorization approach. Similarly, Luo et al. applied non-negative constraints to the Skip-gram model [5-6].

Moreover, the work of [7-8] developed methods to transform dense word vectors into sparse, word vectors with the potential of making ease the final vector's dimensions. These approaches focused on designing new algorithms that enforce through the non-negative constraints and sparseness the interpretability of the learned embeddings. Here, we instead focus on designing an algorithm that works on top of a conventional model such as Skip-gram driven by user input and aiming at finding a semantic continuous similarity space.

7 CONCLUSIONS AND FUTURE WORK

Considering the exponential growth of scientific publications in digital libraries [44], new and innovative ways of extracting knowledge from literature and exploring it are urgently needed. In this context, neural language models can be a tool to automatically and efficiently learn and predict entity relations embedded in literature.

In this paper, we introduced the novel idea of finding semantic subspaces for medical entities. Our approach SemSubspace contributes to the current body of knowledge regarding the *interpretability* of word embeddings. SemSubspace has two core advantages over the original space and competitive state-of-the-art baselines: firstly, it allows finding a user-defined semantic subspace in query-time, and secondly, it shows substantial improvements in two meaningful tasks: clustering of pharmaceutical entities and entity intrusion detection. In both tasks, SemSubspace results show that it can extract a subspace that is semantically coherent. Moreover, our in-depth analysis showed the sta-

bility of our approach regarding the variety of entities selected as input.

To further advance our current efforts in the biomedical field, we would like to address the task of semantic relatedness between two different types of entities, such as those that exist between drugs and diseases. In this context, a drug can be used to *treat* a disease or on the other side can *induce* it. We believe that the extraction of a semantic subspace that represents "treats" or "induces" associations for example, could help to advance literature-based biomedical research analytics such as needed for complex tasks like drug-repurposing.

ACKNOWLEDGMENTS

We thank the German Research Foundation (DFG) for enabling this research: *PubPharm – the Specialized Information Service for Pharmacy* (Gepris 267140244) – www.pubpharm.de

REFERENCES

[1] Rothe, S. et al. 2016. Ultradense Word Embeddings by Orthogonal Transformation. Proceedings of the 2016 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies (San Diego, California, Jun. 2016), 767–777.

[2] Park, S. et al. 2017. Rotated Word Vector Representations and their Interpretability. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Copenhagen, Denmark, Sep. 2017), 401–411.

[3] Luo, H. et al. 2015. Online Learning of Interpretable Word Embeddings. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal, Sep. 2015), 1687–1692.

[4] Murphy, B. et al. 2012. Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding. *Proceedings of {COLING} 2012* (Mumbai, India, Dec. 2012), 1933–1950.

[5] Mikolov, T. et al. 2013. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)* (Scottsdale, Arizona USA, 2013), 1–12.

[6] Mikolov, T. et al. 2013. Distributed Representations of Words and Phrases and Their Compositionality. *Proceedings of the 26th International Conference on Neural Information Processing Systems* (Lake Tahoe, Nevada, 2013), 3111--3119.

[7] Faruqui, M. et al. 2015. Sparse Overcomplete Word Vector Representations. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (Beijing, China, Jul. 2015), 1491–1500.

[8] Subramanian, A. et al. SPINE: SParse Interpretable Neural Embeddings. Proceedings of the Thirty Second AAAI Conference on Artificial Intelligence (AAAI).

[9] Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician*, *27*(1), 17-21.

[10] Osten, D. W. (1988). Selection of optimal regression models via cross-validation. *Journal of Chemometrics*, *2*(1), 39-48.

[11] Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of statistics*, *32*(2), 407-499.

[12] Houle, M. E., Ma, X., Oria, V., & Sun, J. (2016). Efficient similarity search within user-specified projective subspaces. *Information Systems*, *59*, 2-14.

[13] Kibriya, A. M., & Frank, E. (2007, September). An empirical comparison of exact nearest neighbour algorithms. In *European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 140-151). Springer, Berlin, Heidelberg.

[14] Wold, S., Sjöström, M., & Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, *58*(2), 109-130.

[15] Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49-67.

[16] Xun, G., Jha, K., Gopalakrishnan, V., Li, Y., & Zhang, A. (2017, November). Generating medical hypotheses based on evolutionary medical concepts. In *2017 IEEE International Conference on Data Mining (ICDM)* (pp. 535-544). IEEE.

[17] 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.

[18] Gefen, D., Miller, J., Armstrong, J. K., Cornelius, F. H., Robertson, N., Smith-McLallen, A., & Taylor, J. A. (2018). Identifying patterns in medical records through latent semantic analysis. *Communications of the ACM*, *61*(6), 72-77.

[19] Elekes, Á., Schäler, M., & Böhm, K. (2017, June). On the Various Semantics of Similarity in Word Embedding Models. In *Digital Libraries (JCDL), 2017 ACM/IEEE Joint Conference on* (pp. 1-10). IEEE.

[20] Hill, F., Reichart, R., & Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, *41*(4), 665-695.

[21] Jha, K., Wang, Y., Xun, G., & Zhang, A. (2018, November). Interpretable Word Embeddings for Medical Domain. In 2018 IEEE International Conference on Data Mining (ICDM) (pp. 1061-1066). IEEE.

[22] 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).

[23] Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., & Jain, A. (2019). Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, *571*(7763), 95.

[24] Epasto, A. and Perozzi, B. 2019. Is a Single Embedding Enough? Learning Node Representations That Capture Multiple Social Contexts. *The World Wide Web Conference* (New York, NY, USA, 2019), 394–404.

[25] Grover, A. and Leskovec, J. 2016. Node2Vec: Scalable Feature Learning for Networks. *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2016), 855–864.

[26] Sun, F. et al. 2016. Sparse word embeddings using l1 regularized online learning. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (2016), 2915–2921.

[27] Murphy, B. et al. 2012. Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding. *Proceedings of {COLING} 2012* (Mumbai, India, Dec. 2012), 1933–1950.

[28] Manning, C., Raghavan, P., & Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering*, *16*(1), 100-103.

[29] Pinto, J. M. G., Wawrzinek, J., & Balke, W. T. (2019, June). What Drives Research Efforts? Find Scientific Claims that Count! In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)* (pp. 217-226). IEEE.

[30] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*(Oct), 2825-2830.

[31] Pennington, J. et al. 2014. Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), 1532–1543.

[32] Bojanowski, P. et al. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*. 5, (2017), 135–146.

[33] Bengio, Y. et al. 2003. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*. 3, (2003), 1137–1155.

[34] Chiu, B. et al. 2016. How to Train good Word Embeddings for Biomedical NLP. *Proceedings of the 15th Workshop on Biomedical Natural Language Processing* (Berlin, Germany, Aug. 2016), 166–174.

[35] Wang, Y. et al. 2018. A comparison of word embeddings for the biomedical natural language processing. *Journal of Biomedical Informatics*. 87, (2018), 12–20.

[36] Zhang, Y. et al. 2019. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific Data*. 6, 1 (2019), 52.

[37] Jiang, Z. et al. 2015. Training word embeddings for deep learning in biomedical text mining tasks. *Proceedings - 2015 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2015* (2015).

[38] Agrawal, R. et al. 1998. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. *SIGMOD Rec.* 27, 2 (Jun. 1998), 94–105.

[39] Soltanolkotabi, M. et al. 2014. Robust subspace clustering. *The Annals of Statistics*. 42, 2 (2014), 669–699.

[40] Elhamifar, E. and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*. 35, 11 (2013), 2765–2781.

[41] Patrick, M. T., Raja, K., Miller, K., Sotzen, J., Gudjonsson, J. E., Elder, J. T., & Tsoi, L. C. (2019). Drug Repurposing Prediction for Immune-Mediated Cutaneous Diseases using a Word-Embedding–Based Machine Learning Approach. *Journal of Investigative Dermatology*, *139*(3), 683-691.

[42] Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83-85.

[43] Willett, P., Barnard, J. M., & Downs, G. M. (1998). Chemical similarity searching. *Journal of chemical information and computer sciences*, *38*(6), 983-996.

[44] Larsen, P. O., & Von Ins, M. (2010). The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index. *Scientometrics*, *84*(3), 575-603.