Explainable Word-Embeddings for Medical Digital Libraries – a Context-Aware Approach

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

Said Ahmad Ratib Hussaini
TU-Braunschweig
Braunschweig, Germany
s.hussaini@tu-braunschweig.de

Oliver Wiehr
University of Houston
Houston, TX, USA
owiehr@uh.edu

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

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

ABSTRACT

State of the Art Neural Language Models (NLMs) such as Word2Vec are becoming increasingly successful for important biomedical tasks such as the literature-based prediction of complex chemical properties or for finding novel drug-disease associations (DDAs). However, NLMs have the disadvantage of being hard to interpret. Therefore, it is notoriously difficult to explain *why* an artificial neural network learned or predicted some specific association.

Considering that digital libraries offer well-curated contexts, the challenge is to automatically create a reasonable explanation that is intuitively understandable for a user. For a pharmaceutical use case, we present a new method that generates pharmaceutical explanations for predicted DDAs in intuitively understandable sentences. In other words, our approach enables a context-aware access to embedded entities. We test the accuracy of our approach with a comprehensive retrospective analysis considering real DDA predictions. Our explanations can automatically determine the association type (Drug *treats* or *induces* a disease) of a predicted DDA with up to 83%. For existing DDAs, we even achieve accuracies up to 87%. We show that we perform better than deep-learning approaches in this classification task by up to 9%.

CCS CONCEPTS

• **Information systems~Digital libraries and archives**
• Information systems~Specialized information retrieval

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

ACM ISBN 978-1-4503-7585-6/20/06$15.00
https://doi.org/10.1145/3383583.3398522

KEYWORDS

Word embeddings, Interpretability, Context-aware access paths, Pharmaceutical entities, Medical digital libraries, Neural language models, Metadata generation

ACM Reference format:

Janus Wawrzinek, Said A. R. Hussaini, Oliver Wiehr, José M. G. Pinto, and Wolf-Tilo Balke. 2020. Explainable Word-Embeddings for Medical Digital Libraries – a Context-Aware Approach. In *Proceedings of ACM/IEEE-CS* Joint Conference on Digital Libraries *(JCDL'20). ACM, New York, NY, USA, 10 pages.*

1 INTRODUCTION

In times when term-based searches lead to thousands of results, the exponential growth of scientific publications [22] poses medical digital libraries with significant challenges. Innovative services, beyond term-based searches, that facilitate exploration of pharmaceutical entities embedded in literature can help to address these problems. If we consider the bio-medical field, drugs and diseases and their complex relationships among each other, as well as their understanding, play a central role in essential tasks like drug-repurposing [20, 21, 23]. In this context, state of the art neural language models (NLMs) such as Word2Vec [4, 5] can learn efficiently relationships embedded in publications. Thus, NLMs can be seen as possible candidates for the development of such innovative access paths to biomedical literature. The general idea behind NLMs like Word2Vec is based on the contextualization of entities: Entities that appear in similar word contexts (this can be thousands) are positioned close to each other in a high-dimensional space. Novel research shows that this contextualization property can be used to predict complex chemical [11] and new drug relationships [7, 8, 19], as well as can help support the process of medical hypothesis generation [3, 6], to name just a few. In a word-embedding space, a distance between embedded entities may express a particular semantic. On the other side, and in general, the output of an artificial neural network is difficult to explain and a scalar like a mere dis-

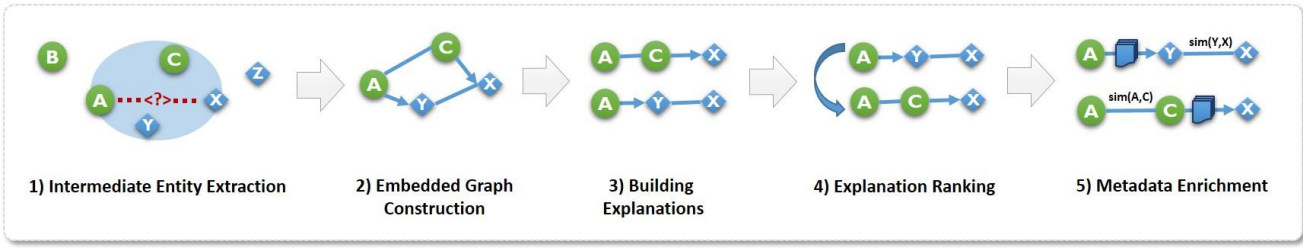

Figure 1. Method overview: we start with a pre-trained model, and then we apply our approach. First, for a given DDA we identify intermediate entities followed by an embedded-graph construction step. After that, we generate explanation-instances, rank them and in the last step we enrich the explanations with semantic metadata.

tance/similarity value is difficult to interpret [1, 2, 8, 9, 13]. In the case of embedded drug-disease associations (DDAs), it hardly contains any semantic information about the relation between the two embedded entities, i.e., *does a drug treat or induce a disease? What can be the reason why a drug probably may treat or cause a disease? What is the probable common context in the embedding space that has been learned by the NLM?*

Regarding NLMs, the general problem of interpretability is known and works like Rothe et al. [2], and Jha et al. [9] present approaches that transform the original embedding space into a semantically optimized subspace. The focus of these methods is mainly on the entire (transformed) space. Therefore, unfortunately, these approaches can hardly give information about what exactly two entities have in common.

Since DDAs plays a central role in pharmaceutical research [20, 21] and NLMs can be used to predict them [7, 19] our goal in this paper is to develop an approach that is able to provide explanations for predicted DDAs in form of short sentences. In addition, we want is to make these explanations intuitively understandable for the users of a medical digital library.

How to explain relationships between drugs and diseases? In pharmaceutical research, the similarity between pharmaceutical entities is the basis for explanations, i.e., if two drugs are chemically similar, we can assume that they also have similar therapeutic or toxic properties [23, 28]. This explanation, based on a similarity of substances, is a drug-centric approach. The counterpart is a disease-centric approach, where a similarity of diseases is determined [20]. The hypothesis here is that a drug is likely to treat or to induce similar diseases. Our explanation approach is based on this kind of triangulation. Therefore, for a predicted DDA an additional (intermediate) entity is always needed to help explain a predicted DDA. In such cases, explanations can be defined as a restricted graph-pattern [24]. Based on this graph-pattern, we can generate drug- and disease-centric explanations for the user, which are intuitively understandable, i.e., like the following drug-centric explanation: Since drug A and B are chemically similar, and drug B induces disease X, there is a high probability that A also induces X. In the next step of our approach, we rank the explanations according to their *information value*, i.e., according to how good they explain a predicted DDA. Important for the user is the degree and type of chemical similarity and which publications describe the individual associations. Therefore, we enrich our explanations with additional meta-information to support users in their exploration. We evaluate our approach and show that we can determine the treats/induces association type of a DDA prediction with an accuracy of up to 83%. We determine these values based on a comprehensive retrospective analysis for real DDA predictions. As baselines, we use state-of-the-art approaches for literature-based classification tasks and show that we even outperform deep learning approaches (up to 5%). We repeat our experiments also for existing DDAs, where the accuracy increases by up to 87%. Here we outperform deep-learning approaches with up to 9%.

We believe that our approach provides innovative access paths to bio-medical literature and can help researchers to explore literature in a new context-aware way: Instead of retrieving entire documents as possible explanations, our approach generates small chains of facts *preserving the contexts* of embedded knowledge. The latter is particularly useful for challenging tasks such as drug repurposing or, in general, for literature-based hypothesis generation.

We organize our paper as follows: Section 2 revisits related work accompanied by our methodology in Section 3. In Section 4, we evaluate our approach with an extensive investigation of embedded drug-disease associations. Finally, we present our main findings in Section 5.

2 RELATED WORK

The prediction and the explanation of drug-disease associations (DDAs) are of great interest in important bio-medical tasks like drug repurposing [20, 21, 23]. As one result of this interest, numerous computer-aided methods have been developed to automatically predict DDAs, whereby the similarity of entities plays a central role for most of them [20, 28]. If we consider active ingredients, then their properties can be inferred based on their chemical/molecular similarity [28]. The main hypothesis is that similar active substances can have similar (therapeutic) properties. Here, in a first step, a bit-fingerprint representation of an active ingredient is generated. This fingerprint can encode information about, e.g., ring compositions, atom sequences, and other molecular features. Next, and using a similarity measure (e.g. Cosine-Similarity), different active ingredients can be compared with each other. Since in this case only active ingredients are compared with each other, this approach is called a drug-centric approach. On the other side, also disease-centric approaches were developed that make predictions based on the similarity of diseases. The hypothesis here is that if a drug is effective against a certain disease, it is likely to be effective against

other similar diseases [20]. For our approach, we use the drug- and disease-centric methods as templates for the design of explanation-patterns.

Specialized databases with well-structured and manually curated information about drugs and diseases are usually the source for the approaches mentioned above. In this context, the Comparative Toxicogenomics Database [27] (CTD[1]) is one of the best databases for curated DDAs, which is why we use it as a ground truth in our work. Furthermore, other mining approaches focus on the extraction of DDAs from unstructured data, such as from biomedical publications. One of the most popular approaches is the co-occurrence/mentioning approach [30], where the hypothesis is that if entities co-occur in documents together, then a relationship/association between these entities can be assumed. In our work, we will use the co-occurrence approach in combination with a retrospective analysis to identify DDA predictions.

Newer approaches use state-of-the-art neural language models (NLMs) such as Word2Vec [4] for the investigation and prediction of semantic relations and word-analogies [5, 10, 11]. Here, the main idea of these models is to embed words in a high dimensional space based on their word-contexts. Afterward, a similarity or relationship between words can be inferred based on the distance in the embedding space. On the other side, the output of a neural network is generally difficult to interpret, so that the meaning of distance in the embedding space often remains unclear [1]. This problem is addressed by recent works [2, 9], that try to create a semantically changed (sub-) space from the original word-embedding space to increase the interpretability. The main idea is to use expert knowledge to learn a transformation matrix. This transformation matrix is then applied to transform the original embedding space to an optimized (sub-) space. One of the drawbacks is that semantic information may be lost [2, 9], and besides, these approaches focus on the entire word-embedding space and therefore cannot explain the relationships between two entities in an understandable fashion. Compared to our approach, we focus on the explanation of entity relationships in the original space.

In this context, Derrac et al. [29] describe in their work that intermediate entities can help explain relationships between a pair of embedded entities. In our work, we will use their approaches to identify intermediate entities.

Fang et al. [24] have introduced a new model called REX, which receives a related entity pair and generates a ranked list of explanations that describe their relationships. For example, their model explains how the actors "Angelina Jolie" and "Brad Pitt" are related to each other. The explanations are based on an existing knowledge source, which contains structured information about all the entities. They are retrieving explanations from a knowledge graph using some predefined explanation graph patterns. This work inspires our approach. Compared to REX, we are not relying on a knowledge graph, but instead, we use the embedding space, and besides, no information exist for predicted DDAs.

[1] http://ctdbase.org/

3 BUILDING EXPLANATIONS

In this section, we formalize the problem we aim to solve in this paper, introduce a core concept: pharmaceutical patterns. Then we describe the five steps of our approach (Sections 3.3-3.7, Figure 1).

3.1 Fundamentals

Our goal in this work is to create a user-understandable explanation for embedded associations. According to Fang et al. [24], we can define an explanation as a restricted pattern, a graph structure that links two entities with each other:

"Given a knowledge base that can be represented as a graph $G = (V, E, \lambda)$*, where* V *is the set of nodes,* E *is the set of edges, and* $\lambda = E \rightarrow \Sigma$ *is the edge labeling function, a relationship explanation pattern can be represented as a 5-tuple,* $p = (V, E, \lambda, vstart, vend)$*, where* V *is the set of node variables, with two special variables* $vstart$ *and* $vend$ *,* E *is a multiset of edges, and* $\lambda = E \rightarrow \Sigma$ *is the edge labeling function." [24]*

As already described in related work, so-called explanation instances are extracted from a knowledge-graph using an explanation-pattern [24]. In addition, the authors used intermediate entities for the explanation-patterns. For example, the intermediate entity of the type film can help explain a relationship between two actors, i.e., they played together in the same film. These intermediate entities are chosen to have a meaningful relationship to $vstart$ and $vend$. In our case, instead of a knowledge graph we have an embedding space and thus our problem can be defined as follows.

Problem definition. Given a predicted DDA, where the drug embedding is the start-entity and the disease embedding is the end-entity, it is necessary to identify *meaningful* intermediate entities, or more specific their embeddings, that can help explain the relationship of the DDA. Afterwards, based on these embeddings an embedded knowledge-graph has to be constructed. Then, this knowledge graph serves as a source to generate explanations using predefined and restricted pharmaceutical patterns.

Transferred to our pharmaceutical use case, this raises several questions and problems: *Which patterns are useful? How to select intermediate entities? How to extract a knowledge graph from the embedding space? If several explanations exist, how can they be ranked in a meaningful way?* In the following sections, we answer all these questions.

3.2 Pharmaceutical Patterns

In this section, we will first describe the pharmaceutical patterns and identify the type of the intermediate entities that are important for us.

As already described in the related work, so-called drug-centric [28] or disease-centric approaches [20] are used in drug

repurposing to predict probable, yet currently unknown drug properties. Drug similarity and disease similarity form the basis for these approaches [23]. Figure 2 shows an example of the relationships between the different entities. In the case of a drug-centric approach we can infer: Drug A is similar (i.e., chemical) to drug B, drug B is effective against disease X, so there is a chance that drug A also helps against disease X.

For the disease-centric approach, we can infer the following: Drug A is used to treat disease Y, where disease Y shows similarities to disease X. In this case, drug A could be also helpful in the treatment of X. Figure 2 shows the two examples for the case that the predicted DDA consists of drug A and disease X.

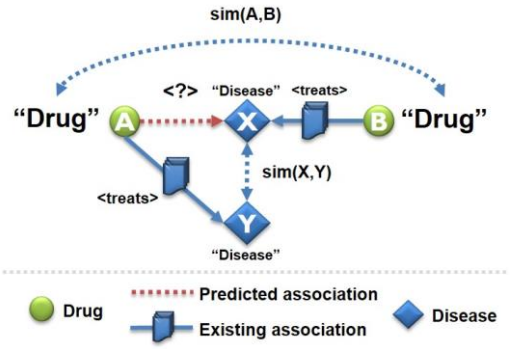

Figure 2. Drug- and disease-centric graph example.

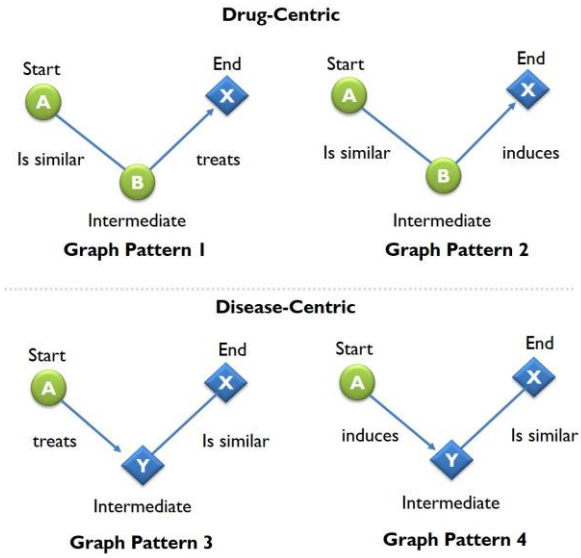

Figure 3. Drug-centric and disease-centric patterns.

We are aware that many other ways to explain a relationship between a drug and a disease exist, i.e., by including genetic information [21]. However, in this paper, we will focus exclusively on the drug/disease centric approaches, as they are easy to understand and to extract. If we consider the examples described above, the intermediate entities are of the type drug or disease. Furthermore, curated information about the relationship between the intermediate- and the start- (i.e., A and Y) or end- (i.e., B and X) entity should be already available. Furthermore, we must be able to calculate a similarity between the entities. With these restrictions, we can define the four explanation patterns for the drug and the disease-centric approaches (Figure 3).

In the next section and in the first step of our approach, we will identify the (embedded) intermediate entities.

3.3 Intermediate Entity Identification

In this section, we describe how to identify embedded intermediate entities using the two embedded start- (drug) and end- (disease) entities. We first discuss what defines meaningful intermediate entities.

What are the characteristics of meaningful intermediate entities? In this context, we have not yet considered the importance of the spatial position of intermediate entities. However, recent work shows [29] that entities located in a particular region, between a pair of entities, can explain the semantic relationship of this pair. We hypothesize that the closer an intermediate entity is to a DDA or its location in a particular region, the higher the probability that all three entities appear together in a (future) publication. Whereby co-occurrence of pharmaceutical entities in documents indicates a direct relationship between them [30]. We will prove this hypothesis in our evaluation by means of a retrospective analysis (see section 4.3). Therefore, we expect that the closer an intermediate entity is to the embedded DDA, the higher the common lexical context, the more meaningful this intermediate entity is for an explanation. To determine the region, we choose an approach proposed by Derrac et al. [29] to calculate a score that represents the (regional) distance from intermediate- i to the start- s and end-entity e embeddings. First, we prove if i is in-between s and e with:

$$\cos(\vec{se}, \vec{si}) \geq 0 \; and \; \cos(\vec{es}, \vec{ei}) \geq 0$$

If this condition is fulfilled, we measure a betweenness-score, which is based on the triangle inequality [29]:

$$BTW_{score}(s, i, e) = \|\vec{si}\| + \|\vec{ie}\| \; - \; \|\vec{se}\|$$

We rescale and normalize the scores, so that a value of 1.0 means that an entity is perfect in-between the embedded DDA and with a decreasing score the distance to the DDA increases. In the next step, we describe how we generate an embedded-graph based on the entities identified by this measure.

3.4 Embedded Graph Extraction

Next, we create one embedded graph per DDA. For this purpose, we first identify all k-nearest intermediate entities of the DDA with the described BTW_{score}. If the intermediate entity is of the type drug, we first check whether curated information to the end entity is available. In other words, whether it is known if the intermediate drug *treats* or *induces* the disease. If information is available, we create an edge from intermediate entity to end entity and add a treats/induces label. If there is no curated information available, we remove the intermediate entity, because in

this case this intermediate entity is of no benefit to the user in an explanation. Analogously we proceed if the intermediate entity is of the type disease. In this case, we check if curated information for the intermediate entity and the starting entity (drug) exists. Therefore, we always create only one *curated* edge from an intermediate entity to either a start or an end entity. The other edge is labelled with a similarity-value (e.g., cosine-, chemical-, therapeutic-similarity, etc.).

In the next step, we describe how we use the explanation patterns to create explanation instances from this graph.

3.5 Explanation Instance Generation

We use the defined explanation patterns to extract explanation instances after generating the embedded graph for a given DDA. We want to describe this step using an example DDA. For this, we choose the DDA between the drug *simvastatin* and the disease *rhabdomyolysis*. Simvastatin is in this context a frequently applied drug for the treatment of hypertension. A severe side effect that can occur under certain circumstances is rhabdomyolysis. During this side effect, the patient's muscles decompose, which can lead to death [31]. For this DDA, we first determine the type of an intermediate entity (drug or disease). If the intermediate entity is of the type drug, we select a drug-centric pattern first. Afterwards, we determine the association type (treat/induces) for the edge between intermediate- and end-entity. If this is an induced association, we select the specific drug-centric pattern (Figure 2, Graph-pattern 2). In the case of simvastatin, the drug lovastatin is an intermediate entity. Figure 4 shows an explanation instance for this case. We perform this process for all intermediate entities and each defined pattern.

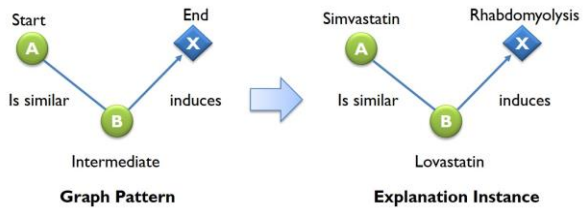

Figure 4. Explanation instance example.

After the explanation-instance generation step, we want to rank the instances according to their *information value*.

3.6 Explanation Ranking

In the next step of our approach, we rank the explanation instances, as well as the explanation patterns, by their information value. How can we determine an "information value" of an explanation instance or a pattern?

Since the probability of a pharmaceutical association increases with increasing cosine-similarity between two entities [19], we assume that an intermediate entity that is closer to a start- or end-entity is of higher interest. Therefore, we rank the explanation instances in a first step combining the BTW_{score} with the cosine-similarity. We calculate the cosine-similarity only between entities of the same type. Thus, if the intermediate entity is of the type drug, we calculate the cosine-similarity between the start- and the intermediate-entity embeddings. If the entity is of the type disease, we calculate the cosine-similarity between the intermediate- and the end-entity embeddings. We calculate a combination of the two measures as follows:

$$BTW_{comb} = \lambda(BTW_{score} + sim_{cos})$$

where λ is a normalization factor. We show in our experiments that this combination is advantageous and leads to improved explanation-instance ranking results (Section 4.4).

In a further step and similar to [24], we assume that the more explanation instances of a certain pattern exists, the more information can be offered to the user, and the more important a pattern and its instances are. Thus, for each pattern we count the number of all generated explanation instances. We rank for presentation the explanation patterns higher with the most explanations instances.

The explanations show what other substances and diseases are roughly related, and how, to a DDA. However, we do not yet know what kind of semantic relationship exists between two drugs, such as between simvastatin and lovastatin, i.e., are the drugs chemical or pharmacological similar? In the next step, we will use metadata that can help to explain the semantic relationship further.

3.7 Explanation-Instance Enrichment

In order to increase the interpretability for the user, we enrich each explanation instance with pharmaceutical metadata. So far, we can only inform the user that there is a *context-based* similarity between the drugs simvastatin and lovastatin, i.e., using cosine similarity or BTW score. However, this information is rather abstract and only conditionally interpretable. It would be better if the user could additionally assess whether two substances are chemically, therapeutically, or pharmacologically similar. To calculate a pharmaceutical-similarity, we use pharmaceutical classification systems. In this context, the *Anatomical Therapeutic Chemical (ATC) Classification System* is one of the most used drug-classification systems and serves as an important source for bio-medical tasks like, e.g., drug repurposing and drug therapy composition [15]. Besides, we use the *American Hospital Formulary Service (AHFS)*. Compared to ATC, in AHFS, drugs are grouped according to their pharmacologic and therapeutic effect and, thus additional information of a different drug-semantic can be presented to the user. Furthermore, we use the *Medical Subject Headings (MeSH) Trees* to generate metadata for drugs as well as diseases. MeSH is a controlled vocabulary where pharmaceutical entities are organized in 16 main categories, e.g. category C for diseases and D for drugs, further divided into finer levels (subgroups), leading to a hierarchical structure.

Using ATC as an example, we want to demonstrate how we calculate a taxonomic similarity between two drugs. In this context, individual drugs such as simvastatin may have several labels. One of the labels is "C10AA01". ATC labels have a hierarchical structure and can be divided into five levels. The first level consists of one letter "C", and describes the anatomical main

group. The following second level "10" consists of two digits and describes the therapeutic subgroup. The third level "A" consists of one letter and describes pharmacological/therapeutic features. The fourth level "A" consist also of one letter and describes additional chemical properties. The last two digits of the fifth level identify the active substance.

Given two class labels of two different drugs, we can now calculate a hierarchical class label overlap to determine a similarity of the drugs. The more levels match, the more similar the drugs are, i.e., if there is a level overlap in the hierarchy up to the fourth level, we can speak of a chemical similarity and if only the first two levels match, we can conclude that there is a therapeutic similarity. If the drugs have more than one label, we always select the label with the maximum overlap. Given two drugs $d1$ and $d2$ and their class labels $label_{d1}$ and $label_{d2}$ we calculate the similarity as follows:

$$Sim_{ATC}\ (d1, d2) = \frac{\max[match(label_{d1}, label_{d2})]}{maximum\ number\ of\ levels}$$

Since the fifth level clearly identifies a drug, the maximum number of levels is four for ATC. The next example shows the calculation for the labels of the two substances $d1 = \{\boldsymbol{C\ 10\ A}\ A\ 01\}$ and $d2 = \{\boldsymbol{C\ 10\ A}\ B\ 01\}$:

$$Sim_{ATC}(d1, d2) = \frac{3}{4} = 0.75$$

In this example, there is a pharmacological/therapeutic similarity between the two active substances. Since AHFS and MeSH also have a hierarchical structure, we use the same procedure to calculate similarity. We use ATC and AHFS to calculate drug-similarities and MeSH for disease similarity.

In the next sections, we evaluate our hypotheses regarding the intermediate entities and investigate the quality of our explanations.

4 EVALUATION

Herein, we describe the general experimental setup and implementation details. Then we explain how we can identify real DDA predictions with retrospective analysis. The DDA predictions form our test set for the entire evaluation.

We perform two experiments. First, we investigate whether the choice of intermediate entities is meaningful with our similarity measures (Section 4.3). Then, we show that our explanations and their ranking can help to determine the type of a predicted association (treats/induces). We compare our approach to different (Deep-learning) baselines. Finally, we present and discuss a single use case. This case refers to the example DDA "Simvastatin-Rhabdomyolysis", which has already been mentioned several times in this paper.

4.1 Experimental Setup and Implementation

In this section, we first describe our document corpus and the query entities. Afterwards we describe our document pre-processing and implementation details.

Document Corpus. For our evaluation corpus, we collected all abstracts from the biomedical digital library PubMed[2] for the period between 01-01-1900 and 06-01-2019 (~29 million abstracts). Furthermore, word-embedding algorithms usually train on single words, resulting in one embedding per word and not per entity. This representation becomes a problem if we consider that drugs and diseases can consist of several words (e.g., coronavirus infections). To solve this problem, we first identify the entities in documents with PubTator[3] and then place a unique identifier (MeSH-Id) at the entity's position in the text.

Query-Entities. DrugBank[4] is one of the most comprehensive databases for curated drug-information. In a first step, we selected all drugs as query entities for the evaluation from the database. Next, we filtered out all drugs that we could not find (using a MeSH-Id) in the CTD Database as well as in the pre-processed documents. Thus, our final entity set for evaluation consists of ~ 1700 drugs. As ground truth, we selected for each drug all manually curated drug-disease associations from CTD. This data set consists of 33541 inducing and 18664 therapeutic drug-disease associations.

Text Pre-processing. We removed stop-words and performed stemming using Lucene's[5] *Porter Stemmer* implementation to increase word-embedding quality and training performance. Here we made sure that the drug and disease identifiers were not affected by ignoring all words with the prefix "mesh:" during the pre-processing.

Word Embeddings. We learned word-embeddings with DeepLearning4J's *Word2Vec*[6] (Skip-gram) implementation. In this context, a larger window-size can lead to improved results in learning (pharmaceutical) associations [8, 13, 19]. Therefore, to train Word2Vec, we set the word window size in our investigations to 50. Further, as proposed by Chiu et al. [13] we set the layer size to 200 features per word and the minimum word frequency of 5 occurrences.

Similarity-Measures. As the similarity measure for intermediate entities, we choose the approach defined in section 3.3 in the experiments (BTW_{score}). As mentioned in section 3.6 (Ranking) we will use BTW_{comb} to rank the explanation-instances. Here, a value of 1 means the highest (best) score, and the value 0 means the lowest (maximal dissimilar) score.

4.2 Retrospective Analysis

In this section, we describe how we identify *real* DDA predictions in historical corpora using a retrospective analysis [3, 11, 19] in combination with a co-occurrence approach [30].

[2] https://www.ncbi.nlm.nih.gov/pubmed/
[3] https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/PubTator/
[4] https://www.drugbank.ca/
[5] https://lucene.apache.org/
[6] https://deeplearning4j.org/word2vec

Retrospective Analysis Evaluation Corpora. In order to detect predicted DDAs using a retrospective analysis, we divide our evaluation corpus into two corpora: 1900.01.01-12.31.1988 (1989 corpus) and 1900.01.01-06.01.2019 (2019 corpus). Each corpus contains only the documents for the respective time period. As next, we proceed as follows.

First, we train our model with the historical corpus t that contains all publications till the year 1989. Next, for each drug-entity we extract the k-Nearest-Disease-Neighbours (leads to k DDAs per drug). Afterwards, we first check if a DDA does not exists, i.e., does not appear in at least three publications [30], in time period t. Then we check if a non-existing DDAs extracted in time period t will appear in time period t+E (Corpus 2019) in at least three documents or/and can be found in CTD. With this approach, we identify DDA predictions within the k-NDNs sets of each drug (where k = 5, 10, 20). This yields to the data set shown in Table 1.

Table 1. Number of real predicted as well as existing DDAs extracted using a k-Nearest-Disease-Neighbors (k-NDN) approach.

	k-NDNs/ DDA-Type	5	10	20
Test (predicted)	inducing	88	168	402
	therapeutic	91	195	474
Train (existing)	inducing	687	1161	2311
	therapeutic	1408	2182	3535

4.3 Evaluation of Intermediate Entities

In our first experiment, we investigate how meaningful our intermediate entities' choice is and how the quality is related to the distance in space. For this, we define the following quality criterion:

A co-occurrence of entities in documents indicates a relationship between these entities [30]. The probability of a co-occurrence should therefore decrease with decreasing similarity (increasing distance) between an intermediate entity and the DDA entities. Thus, with decreasing similarity the probability that start-, end-, and intermediate entity (date < 1989) appear together in a future publication (date >=1989) should decrease.

From our retrospective dataset (Table 1) we select all DDA predictions for k=20. With k=20 we get the highest amount of predictions. For each of the predicted DDAs we extract all intermediate entities using the measure defined in Section 3.3. Next, we check if all three entities co-occur together in future publications (Title/Abstract). With this approach, we determine the precision. To determine the recall we first identify all documents in which the predicted DDA occurs. Then we list all other drug/disease entities that occur in these future documents. It should be noted that new drugs or diseases may be mentioned after >=1989 and therefore the recall in 1989 can never reach a 1. For the existing DDAs, we select the same (predicted) DDAs but from the Word2Vec model trained on the 2019 corpus. For the period until 2019, the predicted DDAs already exist. Our results for predicted DDAs are presented in Figure 5 and in Figure 6 for existing DDAs.

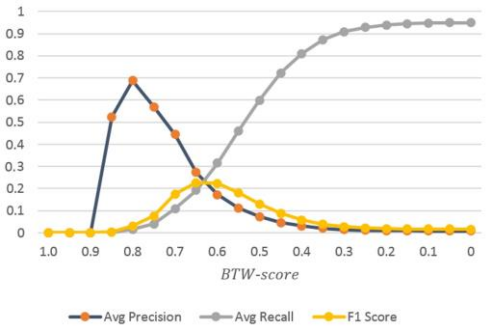

Figure 5. Results for intermediate entities and for predicted DDAs.

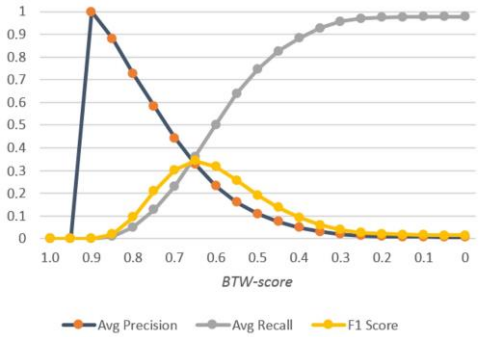

Figure 6. Results for intermediate entities and for existing DDAs.

Results and Result Interpretation. As presented in Figures 5 and 6, with a high BTW_{score} the region is too small, and thus no entities are found initially. For the predicted DDAs (Figure 5) and at a BTW_{score} value of 0.8, the precision increases to 0.69 and decreases slowly as BTW_{score} values decrease. We see a similar result for the existing DDAs (Figure 6), but the precision increases to 1.0. Our experiment confirms our hypothesis. It shows that the probability of co-occurrence decreases with increasing distance. Therefore, explanations whose intermediate entities show a small distance should be preferred and ranked higher. Our approach identifies intermediate entities that probably will appear together (will be set in context) with the DDA entities within a publication. In our next section, we will evaluate the quality of our explanations.

4.4 Evaluation of the Explanations

In this section, we will investigate if explanations can provide valuable information about a predicted DDA. Indeed, it is not easy to assess what information a user would consider as valuable. Therefore, we narrow this down. Thus, "valuable" in this context means that the user can at least decide based on few explanations whether a (predicted) DDA is of the type *treats* or *in-*

duces. From this point of view, this is a classical classification task using word embeddings. To allow a comparison with our approach, we generate three different baselines: A Support Vector Classifier and two deep learning models (Multilayer Perceptron and a Convolutional Neural Network).

In this chapter, we describe how we build and train our baselines (4.4.1). Then we describe how we can perform a simple classification based on the explanations (4.4.2). We do not neglect the users' perspective, so the classification should be possible with a few explanations (i.e., three). In section 4.4.3, we discuss the results.

4.4.1 Baseline Construction

How to use Word-embeddings for classification tasks? In this setting, the work of Lev et al. [17] reveals that one can apply Word2Vec using pooling techniques. These pooling techniques can outperform other algorithms in several NLP classification tasks. Thus, in our work, we consider pooling techniques to build strong baselines. For example, one pooling technique widely used is to calculate a mean vector for different vectors.

Moreover, as investigated by Lev et al. [17], "*concat*" pooling is also a possibility since it helps to capture more semantics that "mean" pooling can lose. In our case, we use a drug vector and a disease vector for the pooling approach. We tried both approaches and found that using the "*concat"* approach led to better results. Thus, we will use "*concat"* pooling in our work to build the following baselines. Note that "*concat*" in our case study means that we concatenate the drug and disease vectors resulting in a 400-dimensional vector.

Using our new DDA feature vector representations, we train the following algorithms to classify a DDA representation into a *treats* or an *induces* association.

Support-Vector-Classifier (SVC) Baseline: We use the Scikit-learns package to train a Support Vector Classifier (SVC). After some experimentation, we chose the following hyperparameters for SVC: a degree of three and a radial kernel.

Multilayer Perceptron (MLP) Baseline: We use the Keras open-source library to train a multilayer perceptron. Our model's architecture consists of three densely connected layers of decreasing size. In the first two layers, we use a rectified linear unit (ReLU) as the activation function. Then, in the final layer, we use a sigmoid function to produce the binary classification output. Finally, we use a binary cross-entropy loss function. To speed up learning, we use Adam optimizer [12], with a learning rate of 1e-4 and a batch size of eight.

Convolutional Neural Networks (CNN): Following the success of CNNs in Computer Vision in challenging tasks such as object detection and image recognition, researchers have also applied CNNs to natural language tasks, including document classification. The interested reader can see Goodfellow, I. et al. [32] to learn more about the key benefits of applying CNNs in such challenging tasks. In our work, we implement a CNN model using Keras with the following architecture and hyperparameter decisions. We use dropout to avoid overfitting with a rate of 0.1. The kernel size is set to three for the concatenated vectors. Like the MLP, the loss function is a binary cross-entropy loss function, and the optimizer is Adam [12] with a learning rate of 1e-4. The batch size is also eight.

Cross-Validation: We applied ten-fold cross-validation in our evaluation for all baseline approaches. First, and for each training and test iteration, we randomly selected a balanced data set consisting of 50% inducing and 50% therapeutic associations. Next, we randomly selected 90% of the associations for training and 10% for testing. Here, all test and training sets contained also 50% inducing associations as well as 50% therapeutic associations and in all experiments we measured the average test accuracy.

4.4.2 Explanation Implementation

We generate the explanations as described in our five steps and for each predicted DDA from the respective evaluation dataset (k=5, 10, 20). To generate the knowledge graph, we select *k*-intermediate entities using the Btw_{score} (Section 3.3). We rank the explanation-instance using our combined measure BTW_{comb} (Section 3.6). To allow a fair comparison to our baselines, we only use an explanation if an intermediate entity and a start or end entity appear together in a publication (publication date < 1989). Only when they co-occur there is a chance that this association could be curated at that time. We ensure that we receive at least three explanation-instances.

Classification with Explanations. We choose the top 3 ranked explanations and by a majority vote we determine the treats/induces type of the predicted DDA. We test our approach on the same balanced test-datasets that we used for the baselines. We show the results of our approach and the baselines in table 2 and 3.

Table 2. Accuracies achieved with the different approaches on the different k-NDN datasets and for predicted DDAs (time < 1989)

Datasets/ Methods	5	10	20
SVC	0.65	0.67	0.71
MLP	0.77	0.75	0.76
CNN	0.78	0.76	0.75
Explanations (k=3)	**0.83**	**0.80**	**0.78**

Table 3. Accuracies achieved with the different approaches on the different k-NDN datasets and for existing DDAs (time = 2019)

Datasets/ Methods	5	10	20
SVC	0.65	0.66	0.69
MLP	0.79	0.75	0.78
CNN	0.78	0.78	0.77
Explanations (k=3)	**0.87**	**0.85**	**0.81**

4.4.3 Results and Result Interpretation

As shown in Table 2, we achieve the lowest accuracy for the SVC. However, the accuracy increases slightly with the number of training data (increasing k), which is not surprising. Overall, we can see that the deep learning approaches always lead to better results compared to the SVC. For the deep-learning approaches, however, despite the increasing number of training data, the average accuracy tends to decrease. We explain this effect by the fact that the deep learning approaches learn a latent DDA representation and prefer certain features. At the same time, we know that with increasing k, therefore with increasing distance between drug and disease, the probability of a DDA generally decreases [19]. Therefore, we assume that the latent information (treats/induces) is weaker expressed in the embeddings when distance increases. Surprisingly, we can see that our simple approach leads to the best results. We can see a similar result for the existing DDAs (Table 3). Here, we reach accuracies of up to 87%. Compared to the deep-learning approaches, we achieve an up to 9% better accuracy. We assume that the increase is mainly related to two facts. First, there are already papers explaining the associations in 2019 and second, the 2019 corpus is much larger. We can assume that with more Word2Vec training data the contextualization quality will also increase. Overall, we can say that our explanations can help to distinguish between treats/induces associations. Compared to the other approaches, the user can better assess a DDA prediction, because the user is not left alone with a single value or label (treats/induces), as would be the case with the baselines. In this context, explanations can be used by the user to explore further and evaluate the result. Furthermore, there is no contradiction if the explanations contain treats as well as induced associations for a specific DDA. Typically, drugs that induce a disease are also often mentioned together in a document, with drugs that can be used to treat the same disease. With our approach, these different relations are *fanned out* for the user.

So far, we have completely ignored the step of metadata enrichment in our experiments. In the next section, we will use an example DDA to show how the *complete* explanations are presented and what additional semantic information they can provide to the user.

4.5 Example Use-Case

In this section, we will illustrate and discuss the result of our explanation approach using a single-use case. In our example, we will use the previously mentioned DDA "Simvastatin-Rhabdomyolysis". It should be noted that no papers exist containing the two entities together before the year 1989. After the year 1989, 422 publications were published about this DDA. Thus, there was (and is) a strong interest explaining this association. We have generated the explanations according to our five steps using the historical corpus (date < 1989) and the Word2Vec model for this time period. Similar to our previous experiments, we only select the explanations if papers are available for the intermediate entities and their curated edges. Hence our assumption again is: If no papers exist, curation of the association at that time would not be possible. This restriction allows us to remove information from the future as much as this is possible. In this example, we also assume that the similarity calculation using ATC, MeSH, or AHFS is possible at that time. This is of course to be considered critically, but we want to simulate the current situation where all this information is available and can be used (nowadays) for predicted DDAs.

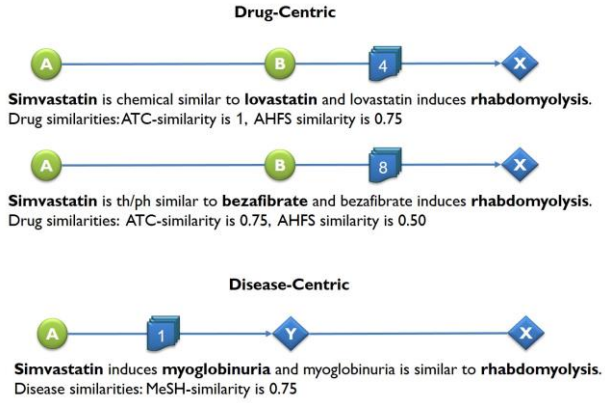

Figure 7. Explanation instances for simvastatin and rhabdomyolysis before the year 1989.

For the DDA simvastatin-rhabdomyolysis the pattern with the most explanation-instances is the drug-centric (induces) pattern as shown in figure 7. The first explanation can be read as follows: Lovastatin is chemically similar to simvastatin (ATC Similarity=1.0) and lovastatin induces rhabdomyolysis. As already mentioned above, chemical similarity indicates similar properties [28], which would allow the hypothesis that simvastatin probably also causes rhabdomyolysis. At this time (<1989), four publications exist in which the entities lovastatin and rhabdomyolysis co-occur and these publications could help to give hints on the relationship between simvastatin and rhabdomyolysis. It is also interesting to know how many publications will be published in the future (>= 1989), that contain all three entities. Executing a query in PubMed with all three entities, the total number of publications is 347. Therefore, there is/was a strong interest to find out and describe the relationship between all three entities after 1989. We get a similar result with the second drug bezafibrate. Here we have eight publications, which mention bezafibrate and rhabdomyolysis together (<1989). According to ATC-similarity (0.75), there is a therapeutic/pharmacological relationship between simvastatin and bezafibrate. Here, after 1989 seven papers were published which contain all three entities of the explanation. A further explanation is generated for the Disease-Centric Pattern (induces). Disease *myoglobinuria* has a high MeSH-similarity (0.75) with rhabdomyolysis. In total 938 publications exist (< 1989) containing myoglobinuria and rhabdomyolysis. It is known that myoglobinuria is a symptom for rhabdomyolysis or muscle destruction [31]. Here, nine papers (>=1989) were published containing all the three entities.

This single example should illustrate the possibilities of our approach. Our approach can, to a certain extent, automatically generate relevant entity relations and point the users to publications where they can find possible important information early.

5 CONCLUSIONS

NLMs offer great opportunities for creating innovative access paths to scientific literature. With the increasing use of AI in intelligent systems, the interest in explaining the results of AI algorithms is growing as well [33, 34]. In this paper, we introduced new context-aware access paths to bio-medical literature, which can help users to explain predicted entity relations.

Our extensive evaluation shows that our approach can predict the type (treats/induces) of a drug-disease association with an accuracy of up to 87% and achieve up to 9% better results compared to deep learning approaches. Therefore, we believe that our method has the potential to support researchers in complex scientific tasks such as hypothesis generation.

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] 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.

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

[3] 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.

[4] 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.

[5] 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).

[6] 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).

[7] 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.

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

[9] 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.

[10] 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).

[11] 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.

[12] Kingma, D.P. and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *CoRR*. abs/1412.6980, (2014).

[13] 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.

[14] 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.

[15] Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., & Woolsey, J. (2006). DrugBank: a comprehensive resource for in silico drug discovery and exploration. Nucleic acids research, 34(suppl_1), D668-D672.

[16] Agarwal, P., & Searls, D. B. (2009). Can literature analysis identify innovation drivers in drug discovery? Nature Reviews Drug Discovery, 8(11), 865.

[17] Lev, G., Klein, B., & Wolf, L. (2015, June). In defense of word embedding for generic text representation. In International Conference on Applications of Natural Language to Information Systems (pp. 35-50). Springer, Cham

[18] 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

[19] Wawrzinek, J., & Balke, W. T. (2018, November). Measuring the Semantic World–How to Map Meaning to High-Dimensional Entity Clusters in PubMed? In *International Conference on Asian Digital Libraries* (pp. 15-27). Springer, Cham.

[20] Chiang, A. P., & Butte, A. J. (2009). Systematic evaluation of drug–disease relationships to identify leads for novel drug uses. *Clinical Pharmacology & Therapeutics*, *86*(5), 507-510.

[21] Hu, G., & Agarwal, P. (2009). Human disease-drug network based on genomic expression profiles. *PloS one*, *4*(8), e6536.

[22] 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.

[23] Gottlieb, A., Stein, G. Y., Ruppin, E., & Sharan, R. (2011). PREDICT: a method for inferring novel drug indications with application to personalized medicine. *Molecular systems biology*, *7*(1), 496.

[24] Fang, L., Sarma, A. D., Yu, C., & Bohannon, P. (2011). Rex: explaining relationships between entity pairs. *Proceedings of the VLDB Endowment*, *5*(3), 241-252.

[25] 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.

[26] Hinton, G.E. et al. 2012. Improving neural networks by preventing co-adaptation of feature detectors.

[27] Rinaldi, F., Clematide, S., & Hafner, S. (2012, April). Ranking of CTD articles and interactions using the OntoGene pipeline. In *Proceedings of the 2012 BioCreative Workshop*.

[28] Keiser, M. J., Setola, V., Irwin, J. J., Laggner, C., Abbas, A. I., Hufeisen, S. J., & Whaley, R. (2009). Predicting new molecular targets for known drugs. *Nature*, *462*(7270), 175.

[29] Derrac, J., & Schockaert, S. (2015). Inducing semantic relations from conceptual spaces: a data-driven approach to plausible reasoning. *Artificial Intelligence*, *228*, 66-94.

[30] Jensen, L. J., Saric, J., & Bork, P. (2006). Literature mining for the biologist: from information retrieval to biological discovery. *Nature reviews genetics*, *7*(2), 119

[31] Omar, M. A., & Wilson, J. P. (2002). FDA adverse event reports on statin-associated rhabdomyolysis. *Annals of Pharmacotherapy*, *36*(2), 288-295.

[32] Goodfellow, I. et al. 2016. Deep Learning--book. *MIT Press*. 521, 7553 (2016), 800.

[33] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). Why should i trust you? Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144). ACM.

[34] Samek, W. (2019). Explainable AI: interpreting, explaining and visualizing deep learning (Vol. 11700). Springer Nature.