Processing Patient Information Leaflets with Embeddings

Sven Stahlmann
University of Cologne
stahlmann@wim.uni-koeln.de

Stefan Hirschmeier
University of Cologne
hirschmeier@wim.uni-koeln.de

Detlef Schoder
University of Cologne
schoder@wim.uni-koeln.de

Abstract

As of 2021, more than 100,000 drugs are approved in Germany, 35,000 of which are non-prescriptive over-the-counter drugs. While proven information from medical studies is given in patient information leaflets, patients are often lost when trying to determine which drugs are compatible with their needs or which alternatives are suitable. We show that representing patient information leaflets as dense vectors allows us to extract more valuable medical information than is explicitly stated in the leaflets. Without any explicit insertion of medical knowledge, our embeddings capture concepts of generics, even with respect to the dosage form. Furthermore, the embeddings allow patients to identify drug clusters based on their treatment area and offer suitable alternatives based on analogical reasoning. The carved-out information may not only help patients to explore alternative drugs but also serve pharmacists and patients as a new way to search for drugs tailored to dietary, allergic, or medical needs.

1. Introduction

Although technology has revolutionized many aspects of daily life, the field of medical service is still largely in the hands of medical and healthcare professionals. Even for minor issues, patients request advice from medical professionals, increasing waiting times for patients and workloads for healthcare professionals. Among these minor issues are legitimate requests for alternative drugs, either because of drug intolerance, the intention to find a cheaper alternative, or problems with the dosage form. The sheer volume of available drugs makes it difficult for patients to keep track of which drugs are available for their specific condition. As an example, Germany has a total of more than 100,000 approved drugs, 35,000 of which are non-prescriptive, i.e., sold over the counter [1].

By law, drugs are supplied with patient information leaflets. While these leaflets contain valuable medical information for the end consumer, such as detailed information about the dosage, side effects, and interactions with other drugs, they usually do not provide information about alternatives. In addition to scarce information about alternatives in leaflets, studies have also shown that only half of the patients read the leaflets because reading them increases their anxiety level [2]. Searching for alternatives on the internet may result in single hits but does not provide a comprehensive overview.

Recent research has shown that dense vector representations of text or words can capture semantic meaning and relationships of words and documents within a corpus [3–5]. Approaches such as word2Vec [4, 5] and paragraph2vec [3], also known as doc2vec, have been used to transform text into numerical representations to gain knowledge about semantic structures of text without depending on any external knowledge [6]. With the resulting vector representations, we can calculate the cosine similarities based on the dot products of two vectors.

Texts represented in vectors in an algebraic space enable us to retrieve documents similar to a given document, in our case, retrieving drugs similar to a drug a patient already takes. Given that these retrieved drugs treat the same medical condition, the patient can choose the drug that best fits his or her preferences. Additionally, arithmetic operations on these vectors can help to query alternatives in a more target-oriented way, e.g., by excluding an ingredient from a drug that the patient does not tolerate—in terms of vector arithmetics, subtracting the ingredient vector from the drug vector, and obtaining a vector that is close to drugs that do not contain the undesirable ingredient.

Given the possibilities of semantic embeddings, we ask the following question: *How can medical knowledge such as alternative drugs (in terms of active ingredients and dosage forms) be extracted from patient information leaflets*?

2. Theoretical Background

One approach to natural language processing (NLP) that has gained enormous traction in the past several years is capturing the context of words and

URI: https://hdl.handle.net/10125/79789
978-0-9981331-5-7

representing them in a distributed way as vectors (also called word embeddings)—that is, each word is represented by a series of coordinates that position it in a high-dimensional space. Different approaches to generating these vectors include word2vec [4, 5], Glove [7], FastText [8], ELMo [9], and Sentence-BERT [10]. The word2vec model by [4] has been shown to generate high-quality embeddings in different scenarios [11–13].

The word2vec model causes words that appear in a similar context to have similar vectors with respect to cosine similarity. This results in semantically close words also being close in the vector space, e.g., the vector "Paris" is closer to "France" than it is to the word "car." The embeddings not only capture semantic distance but were also shown to mirror the semantic relationships of words in a text corpus. This allows semantic computations to be performed on the results [11, 13, 14]. One famous example is to solve analogy questions such as "King relates to Man as Woman relates to...?" by a simple vector calculation, i.e., the vector of the word "King" minus the vector of the word "Man" plus the vector of the word "Woman" results in a word vector that is close to the vector of the word "Queen" in terms of distance [11].

Word2vec is proposed for learning word embeddings through two neural network architectures: Continuous Bag of Words (CBOW) and the Skip-gram model [4]. Both are neural network architectures that efficiently learn vector representations from very large datasets and preserve the semantics of the processed text. Both algorithms use a flat, fully connected neural network with a single hidden layer to generate vector representations of each word in a corpus. The actual task of the neural network is to maximize the average log-likelihood of each context word for a given target (mean) word, where the prediction of the likelihood is computed using a hierarchical softmax function [3]. The Skip-gram model uses the mean word as input to the neural network and attempts to predict the context words. The CBOW model works in the opposite way, using the context words as input to predict the mean word. The context is derived from a sliding window over the document and contains a fixed number of words before and after the mean word [4].

Le and Mikolov extended the word2vec model to the paragraph2vec model, also known as doc2vec, by adding a paragraph (document) vector that is shared across all contexts generated from the same documents, but not across all documents [3]. This allows vectors to be generated for sentences and paragraphs. Analogously to word2vec, there are two different methods for learning distributed representations of documents: the Distributed Bag of Words version of the Paragraph Vector (PV-DBOW) and the Distributed Memory version of the Paragraph Vector (PV-DM). The PV-DBOW model is very similar to the Skip-gram model of word2vec. The only difference is that instead of the middle word, a unique document vector is fed into the neural network [3]. The PV-DM is a modification of the CBOW model. In this model, the unique paragraph vector is added to the context words in the input layer. Thus, the unique paragraph vector contributes to the task of predicting the mean word [3]. In this model, the paragraph and word vectors are then trained in parallel.

Recent context-aware embeddings such as ELMo and Sentence-BERT have had breakthrough performances in classification and sentence pair tasks [9, 10]. Instead of having a fixed vector for a word, the vector of the word changes based on the sentence it is used in. While this improves classification performance, it does lose the ability to do vector arithmetics with single words; therefore we did not use context-aware embeddings for our approach.

Traditional approaches exist to represent medical knowledge (e.g., through databases using SNOMED CT [15] or ATC [16]). These enable professionals to query drugs in a structured way. However, in comparison to our proposed NLP approach they don't include the context that these drugs are used in.

3. Approach

We structure our research following the Cross Industry Standard Process for Data Mining (CRISP-DM) approach [17]. The CRISP-DM framework comprises six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. In the following subsections, we describe our main actions during the first five phases of CRISP-DM.

3.1. Business understanding

Patients rarely have an overview of all the available medication. On the internet, search results for, e.g., "ibuprofen alternatives" bring up several million hits, and forums are full of questions for drug alternatives. Patients are often aware of some over-the-counter drugs that they take to treat common conditions such as colds or headaches, but they might not know of alternatives that have the same medical profile and effect. Even if they have the knowledge of the existence of some alternatives, finding alternatives suitable to their dietary or allergy needs is not an easy task without expert knowledge. Offering a method that extracts similar drugs or alternative dosage forms offers patients a significant benefit. It opens the possibilities to efficiently avoid intolerance of specific ingredients, find cheaper alternatives, or identify more appropriate dosage forms.

3.2. Data understanding

The data includes a total of 13,644 patient information leaflets for over-the-counter drugs that were approved in Germany and were provided by ABDATA [18]. These text documents are written in the German language. In general, the patient information leaflet of a drug is a text document containing essential characteristics of a medicinal product in a highly standardized structure, as they must comply with a predefined layout. Their main purpose is to ensure the correct use of the drug, as well as to inform users about the frequency and nature of side effects. Among other details, they include the name of the drug, the dosage form and amount, the medical application field, interactions with other medicines, side effects, and the full pharmaceutical composition. Drugs are often available in different dosage amounts. However, the dosage amount is not relevant to our research question; therefore we opted to merge all drugs with the same name and dosage form irrespective of the dosage amount.

3.3. Data preparation

A document in the corpus corresponds to a patient information leaflet document of a single drug. We applied the following preprocessing steps: summarizing drugs with different dosages and thus removing numerical values, lowercasing all words in the leaflets, and filtering German stop words (using the NLTK Python library).[1] Then, we tokenized the leaflet and assigned a unique document tag as an identifier. The unique document tags are made up of the drug name and the dosage form so that drugs with the same name can be distinguished at an additional level in the evaluation process. These preprocessing steps resulted in a total of 6,488 leaflets out of a total of 13,644 raw documents with a total vocabulary size of 21,024 words. We have focused on a set of common data preprocessing steps that are typical when dealing with text. The goal was to remove as much noise as possible while retaining as much medical knowledge in these documents as possible.

3.4. Modeling approach

In the modeling phase, the doc2vec model learns the document and word embeddings by transforming the input text into numerical representations. For this, we used the Python library Gensim[2]. This library is a technical implementation of the conceptual models introduced by [3–5].

To train the model, a configuration of several parameters is required. The configuration of the so-called hyperparameters and their optimization is a field of research that is the subject of comprehensive discussions and intense research [17, 18]. Nevertheless, most of the research in this field concentrates on text documents in the English language, and thus the scope of hyperparameters and their optimization for German texts are less extensively studied [21].

In our research, we do not intend to find the optimal combination of parameters, achieved by tuning the parameters with the help of a grid-search process. Nevertheless, the combination of hyperparameters is a prerequisite for calculating high-quality document and word embeddings, and thus our configuration is guided by the work of Brito et al., who tested hyperparameter configurations for doc2vec models on German texts [21].

Following the research mentioned above, the combination of the hyperparameters we used in our work is summarized in Table 1, which gives a brief description and the corresponding value of each.

Table 1: Hyperparameters selected based on the research conducted by Brito et al. [21]

Hyperparameter	Meaning	Value
vector_size	Dimension of the feature vectors.	300
min_count	Ignore words with a total occurrence lower than this threshold.	5
sample	Threshold for which higher-frequency words are randomly downsampled.	1e-5
negative	Number of negative samples.	5
dm	Specifies the training algorithm; 0 equals PV-DBOW.	0
dbow_words	If set to 1, trains word-vectors (in Skip-gram fashion) simultaneously.	1
Window	Determines the width of the sliding window.	10

We adapted two hyperparameters from Brito et al. [21] to better fit the model to our specific data source. First, we reduced the value *min_count* to 5 to keep

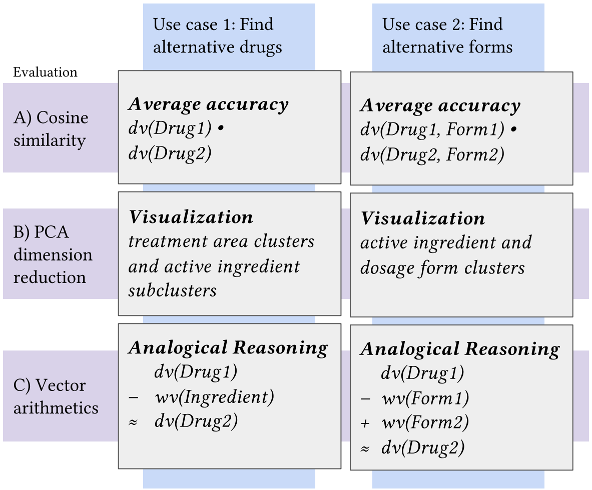
[1] https://www.nltk.org

[2] https://radimrehurek.com/gensim

words that occur rarely in the corpus. Since the patient leaflets contain specific terms and the total size of the vocabulary is small, we did not want to risk important semantic insights getting lost. Second, we set the value *dbow_words* to keep the trained word embeddings.

3.5. Evaluation approach

According to Baroni et al., there are various absolute intrinsic evaluation categories, such as semantic relatedness, analogy, categorization, and selectional preference for evaluating an unsupervised embedding model [22]. We decided to evaluate our model in three facets (Figure 1), which reflect the first three evaluation categories mentioned by [22]. First, we refer to semantic relatedness (method A in Figure 1). For semantic relatedness, Schnable et al. state that the cosine similarity of the embeddings for a pair of words should have a high correlation with the relatedness score rated by humans [23]. In this research, we follow this concept but modify it to the extent that the relatedness score of documents rather than words will be taken into account. Second, we refer to a mix of semantic relatedness and categorization, as we reduce dimensions by principal component analysis (PCA) to be able to visualize clusters (method B in Figure 1). Third, we refer to an analogy by performing vector arithmetics (method C in Figure 1). While method A usually refers to the application field of quantitative evaluation approaches, methods B and C represent qualitative approaches.



Evaluation	Use case 1: Find alternative drugs	Use case 2: Find alternative forms
A) Cosine similarity	***Average accuracy*** *dv(Drug1) • dv(Drug2)*	***Average accuracy*** *dv(Drug1, Form1) • dv(Drug2, Form2)*
B) PCA dimension reduction	***Visualization*** *treatment area clusters and active ingredient subclusters*	***Visualization*** *active ingredient and dosage form clusters*
C) Vector arithmetics	***Analogical Reasoning*** *dv(Drug1)* – *wv(Ingredient)* ≈ *dv(Drug2)*	***Analogical Reasoning*** *dv(Drug1)* – *wv(Form1)* + *wv(Form2)* ≈ *dv(Drug2)*

Figure 1: Evaluation approach with three methods and two use cases

Furthermore, we evaluated two different use cases that substantiate our research question 1) finding alternative drugs and 2) finding alternative dosage forms. The resulting evaluation approach consists of 3x2 evaluation facets (Figure 1). In the following subsections, the evaluation methods are described in further detail.

Evaluation method A. Evaluation A is a quantitative evaluation in which we evaluate the accuracy of alternative drugs or dosage forms. We describe the evaluation process for each use case separately. The process is depicted in Figure 2.

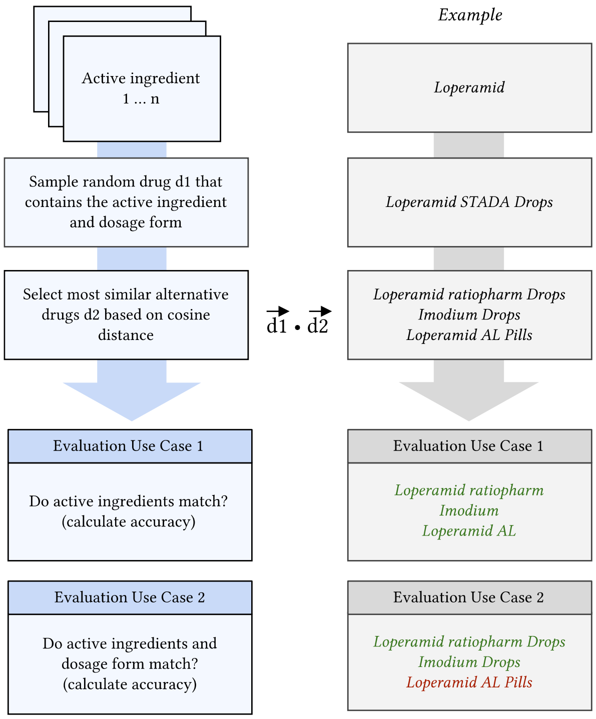

Active ingredient 1 … n
Sample random drug d1 that contains the active ingredient and dosage form
Select most similar alternative drugs d2 based on cosine distance
$\vec{d1} \bullet \vec{d2}$
Evaluation Use Case 1
Do active ingredients match? (calculate accuracy)
Evaluation Use Case 2
Do active ingredients and dosage form match? (calculate accuracy)
Example
Loperamid
Loperamid STADA Drops
Loperamid ratiopharm Drops
Imodium Drops
Loperamid AL Pills
Evaluation Use Case 1
Loperamid ratiopharm
Imodium
Loperamid AL
Evaluation Use Case 2
Loperamid ratiopharm Drops
Imodium Drops
Loperamid AL Pills

Figure 2: Evaluation process for method A (accuracy via cosine similarities)

Evaluation A for use case 1. Find alternative drugs. In the first use case, we evaluated the performance of the model with respect to identifying matching suitable alternative drugs based on their active ingredient. For this purpose, we defined a sample of 20 drugs selected based on frequently prescribed active ingredients in Germany in 2015 (see Table 2). For each active ingredient, we randomly selected a corresponding drug. Next, we submitted each of these sampled drugs to our model, which listed the most similar drugs as an output. We then calculated the accuracy of the selected drugs based on matching active ingredients to the submitted drug. This process was repeated ten times in total and an average accuracy was calculated based on the results of all active ingredients. A high accuracy would lead to the conclusion that the model can identify alternative drugs.

Evaluation A for use case 2. Find alternative forms. In the use case, we measured the performance on a more detailed level by taking the dosage form of a drug into account in addition to its active ingredient. Therefore, we combined some dosage forms into main categories. For example, “film-coated tablets,” “coated tablets,” or “extended-release tablets” are grouped as “tablets.” Since not every active ingredient listed in Table 2 is available in different dosage forms, a subset of the list needed to be created. The ingredients that are part of this subset are checked in the column Dosage Form Evaluation below.

Table 2: Top 20 active ingredients frequently prescribed in Germany [24]

Active Ingredients	Altern. Drugs Evaluation	Dosage Form Evaluation
Acetylsalicylsäure	✓	
Allopurinol	✓	
Azithromycin	✓	✓
Bisoprolol	✓	
Diclofenac	✓	✓
Doxycyclin	✓	✓
Ibuprofen	✓	✓
Levothyroxin	✓	
Metamizol	✓	✓
Metformin	✓	
Metoprolol	✓	
Omeprazol	✓	✓
Pantoprazol	✓	
Paracetamol	✓	✓
Ramipril	✓	
Tilidin	✓	✓
Torasemid	✓	
Tramadol	✓	✓
Venlafaxin	✓	✓
Xylometazolin	✓	

We faced the problem that the number of drugs sharing the same active ingredient and the same dosage form is not evenly distributed in our dataset. To get unbiased results, we set the number of the tested results equal to 20 or the total number of potential matches, whatever is lower.

Evaluation method B. As a qualitative evaluation facet, we decided to plot the drug vectors in a two-dimensional chart to see if clusters can be identified visually. To break the multidimensional vectors from the doc2vec neural net down to two dimensions, we used Principal Component Analysis (PCA),[3] chose a random sample, and plotted the resulting two-dimensional vectors. The idea of PCA is to reduce the dimensions of the vector space while preserving the similarities and dissimilarities as well as possible. Next to the quantitative evaluation of method A, method B supports the evaluation visually and qualitatively. For use case 1, we plot the drugs only; for use case 2, we plot drugs and dosage form.

Evaluation method C. In addition to evaluation with methods A and B, we wanted to evaluate whether our model allows for vector arithmetics to query alternatives in a more target-oriented way, either by excluding ingredients, e.g., because of intolerance (use case 1), or by exploring alternative dosage forms (use case 2). In terms of vector arithmetics, for use case 1 we would query

dv(Drug1) – wv(Ingredient) ~= dv(Drug2)

to receive a vector that is expected to be close to a drug that contains the same active ingredient but does not contain the undesired ingredient, with *dv* denoting a document vector and *wv* a word vector. Analogously, for use case 2, we would query

dv(Drug1) – wv(Form1) + wv(Form2) ~=dv(Drug2)

to receive a vector that is close to drug vectors with the same active ingredient but a different dosage form. For the result we query the drug vector closest to the vector of the arithmetic operation.

While the idea of vector arithmetics is inspired by the famous *wv(king) - wv(man) + wv(woman) ~= wv(queen)* example [4], whether word vectors can be added to document vectors is still an open question. Technically, the arithmetic operation is feasible, as both word vectors and document vectors share the same dimensionality and structure. However, it is not clear if word vectors and document vectors can be mixed in a meaningful way, as a word vector represents the context of a word in the whole text corpus, whereas a document vector represents the context of a document, which is less easy to imagine.

In a way, the training procedure for word vectors and document vectors is similar, but the training input is different, so one could argue that there is a fundamental difference between word vectors and document vectors, but one could also argue for similarities. Lau and Baldwin [20] state that the qualitative difference between word vectors and document vectors remains unclear. They try to give an impression of the differences with an example document. Apart from that, the comparability of word vectors and document vectors has not been thoroughly discussed in the literature so far. Practitioners who have been experimenting with similarities across words and documents find that—at least on a Wikipedia corpus—the closest similar vectors for words are mostly other words, and for documents mostly other

[3] PCA was done using the scikit-learn python library https://scikit-learn.org

documents.[4] Furthermore, they state that it depends on the training method and data whether it is meaningful to compare word vectors and document vectors.

In this paper, we assume that adding word vectors to document vectors is possible. This way, our way of vector arithmetics goes beyond the proposed vector calculus proposed by Mikolov et al. [4]. Therefore, if we are able to show that these vector arithmetics with word vectors and document vectors do make sense, we not only evaluate the usefulness of our model but also show—at least in the example of patient information leaflets—that vectors of different types can be mixed in calculus.

4. Results

In this section, we present the results of the evaluation for the two use cases and the three different evaluation methods outlined in the previous chapter.

4.1. Use case 1: Find alternative drugs

In the first use case, we aimed to identify alternative drugs with the same active ingredient, or, in a wider scope, with the same effect but different active ingredients. We present the results from evaluation methods A, B, and C below.

Table 3: Results from evaluation A for alternative drugs

Alternative Drugs Evaluation

Iteration	Accuracy
1	0.98
2	0.97
3	0.97
4	0.97
5	0.97
6	0.96
7	0.97
8	0.90
9	0.97
10	0.95
Ø	**0.96**

Results from evaluation A (accuracy of alternative drugs via cosine similarities). We conducted 10 iterations of the alternative drugs evaluation approach described in Section 3.5 on 20 example drugs from the list of active ingredients shown in Table 2. The results for each iteration and the average are presented in Table 3.

From these results we can see that the accuracy of the model at selecting correct alternative drugs based on an active ingredient lies between 90% and 98%. Looking at the worst performing run, from the drugs selected by the model, nine out of ten drugs are valid alternatives to the supplied drug.

Figure 3 presents the average accuracy grouped by active ingredients. Most of the ingredients achieve strong results over 95%, although the results vary across some ingredients.

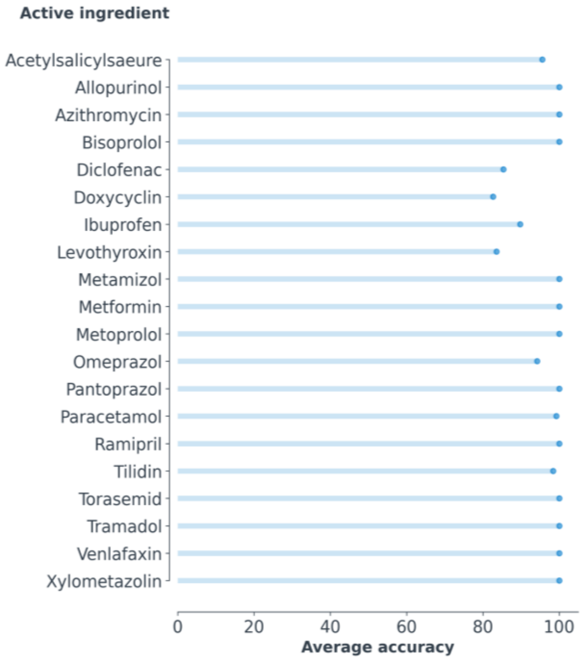


Figure 3: Average accuracy of each active ingredient for alternative drugs evaluation

Results from evaluation B (visual plot of PCA dimension reduction). We selected a random sample of drugs, reduced the vectors by PCA, and scatter plotted the resulting two-dimensional vectors. The points are color-coded based on the active ingredient of the drugs. In the scatter plot, clear clusters are visible. First, one cluster groups together different painkillers based on active ingredients, such as Acetylsalicyl acid, Diclofenac, Ibuprofen, and Paracetamol but also mostly separates them into sub-clusters according to their ingredient. Second, there is a cluster of drugs against diabetes based on “Metformin” that is also clearly separated from other clusters like painkillers. Third, another cluster groups drugs treating thyroid hormone deficiency and is also characterized by a low inter-cluster similarity and a high intra-cluster similarity. Last, an interesting cluster in the bottom

[4] https://groups.google.com/forum/#!topic/gensim/Fujja7aOH6E

right corner groups various oil-based supplements used for the treatment of colds.

Therefore, we can both identify clusters of the same treatment area (red circles in Figure 4) and clusters of the same active ingredient (sub-clusters of the same color).

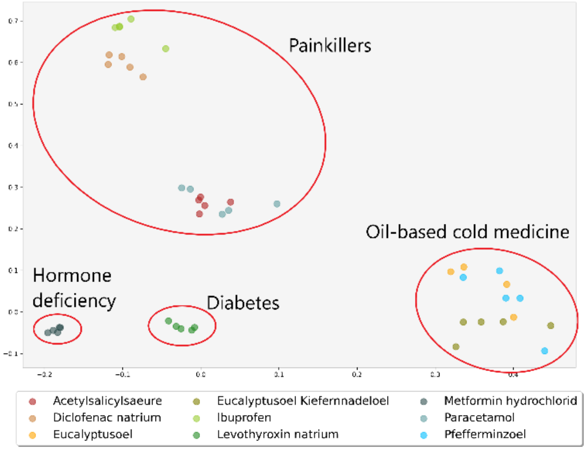


Figure 4: Average accuracy of each active ingredient for alternative drugs evaluation

Results from evaluation C (vector arithmetics for analogical reasoning). To evaluate whether vector arithmetics provide meaningful results, we present a handful of examples. We start by selecting a suitable starting drug from which we subtract typical allergenic ingredients such as lactose, fructose, and gelatin that is present in the drug. The evaluation is considered positive if the closest drug to the result of the arithmetic operation has the same medical effect as the original drug but does not include the undesired ingredient. In the following, we present some examples that perfectly allow vector arithmetics.

- **Example 1: Subtracting Lactose.**
  dv(Metamizol HEXAL Film-coated tablets) – wv(Lactose) ~= dv(Novalgin for kids suppositories)
  While the original drug was provided in pills and contains lactose, the substitute drug contains the same active ingredient but is provided in the form of suppositories that do not contain lactose. Overall, we found that the subtraction of the word vector lactose works very well, resulting in possible alternatives without lactose.
- **Example 2: Subtracting benzylalcohol.**
  dv(Ibuprofen AbZ Sirup) - wv(benzylalcohol) ~= dv(IBULYSINratiopharm coated tablet)
  The original drug contains the ingredient benzylalcohol. The substitute drug has the same active ingredient as the original but contains no benzylalcohol.
- **Example 3: Adding pine needle oil.**
  dv(Exeu Capsule) + wv(pine needle oil) ~= dv(Transpulmin Cold Balsam for kids)
  Interestingly, not only subtraction but also addition provides meaningful results. The original drug and the substitute drug both have the same active ingredient except that the substitute has the added active ingredient of pine needle oil.

However, we also found areas where vector operations do not seem to be successful. Subtracting fructose did not yield successful results in our evaluation. We also tried to subtract side effects from drugs but were not able to achieve positive results.

4.2. Use case 2: Find alternative forms

In the second use case, we aimed to identify alternative dosage forms for the same ingredient. In the following, we present the results from evaluation methods A, B, and C.

Results from evaluation A (accuracy of alternative drugs via cosine similarities). We conducted the evaluation based on the ten active ingredients shown in Table 2. The evaluation details are described in Section 3.5. The results are presented in Table 4 and display the averages of the respective runs.

Table 4: Results of method A for alternative dosage forms

Alternative Drugs Evaluation

Iteration	Accuracy
1	0.68
2	0.79
3	0.89
4	0.75
5	0.80
6	0.73
7	0.79
8	0.84
9	0.84
10	0.79
Ø	**0.79**

Unsurprisingly, this evaluation leads to lower accuracy results than the evaluation of the first use case, since it must consider dosage form and active ingredient to be considered a correct selected drug. Nevertheless, the results range between 68% and 89% and are conservatively formulated; in the worst performing run, almost seven out of ten drugs are correctly identified.

Figure 5 below displays the average accuracy of the evaluation grouped by active ingredient. We see the same effects as in use case 1 but with overall lower accuracy values and higher variance among the different ingredients.

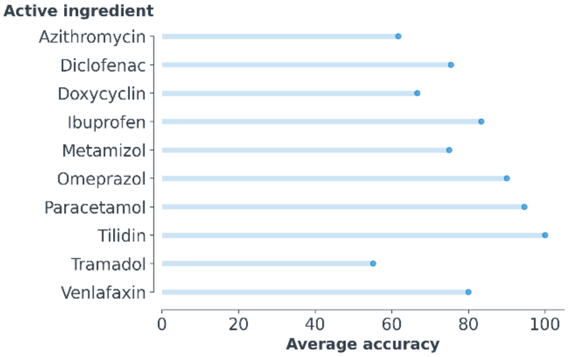


Figure 5: Average accuracy of each active ingredient for dosage form evaluation

Results from evaluation B (visual plot of PCA dimension reduction). We randomly selected drugs containing the active ingredient Ibuprofen and Paracetamol with different dosage forms. The resulting document vectors were reduced using the PCA method and plotted. The result is illustrated in Figure 6. The symbols are color-coded based on the active ingredient; different symbols represent different dosage forms.

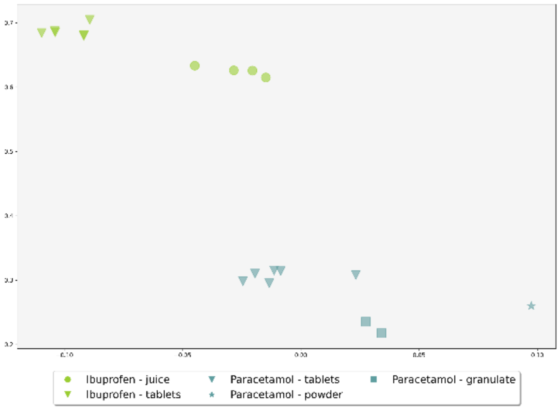


Figure 6: PCA plot of selected drugs containing Ibuprofen or Paracetamol

The plot nicely illustrates that subclusters based on the dosage form exist. At the top, we see the Ibuprofen cluster, which is separated into two subclusters, one for the dosage form juice and one for tablets. At the bottom of the plot, the Paracetamol cluster is not as clearly separated into subclusters. While most of the Paracetamol tablets are grouped, one outlier is closer to the Paracetamol granulate subcluster. The Paracetamol drug with the dosage form powder is clearly separated from the other subclusters.

Results from evaluation C (vector arithmetics for analogical reasoning). To show the viability of analogical reasoning for finding alternative dosage forms of drugs using vector arithmetics, we present a handful of examples. We start by selecting a suitable drug for which other drugs with the same active ingredient but different dosage forms exist. We then subtract the dosage form from the original drug and add the desired dosage form. The drug closest to this vector operation should ideally match the original drug in its active ingredient, but should not have the subtracted dosage form, which is present in the original drug, but the dosage form supplied by the addition. In the following, we present some examples that achieve the stated goal.

- **Example 1: From tablets to drops.**
  dv(Metamizol HEXAL Tablets) - wv(tablets) + wv(drops) ~= dv(Metamizol HEXAL Drops)
  The original drug has the dosage form tablets, while the drug nearest to the resulting vector has the dosage form drops. Both the original drug and the resulting drug have the same active ingredient and are painkillers.
- **Example 2: From drops to juice.**
  dv(Ambroxol acis Drops) - wv(drops) + wv(juice) ~= dv(AmbroHEXAL S Cough Syrup Juice)
  The suggested drug resulting from this vector operation has the same active ingredient (Ambroxol hydrochlorid) as the original drug. It fits the added dosage form of juice instead of the original dosage form of drops.
- **Example 3: Just no tablets, please.**
  dv(Paracetamol OPT Tablets) - wv(tablets) ~= dv(RubieMol Juice)
  Just subtracting without adding the desired dosage form also leads to suitable results. *Rubimol Juice* and *Paracetamol OPT Tablets* have both the active ingredient Paracetamol. As the names suggest, *Paracetamol OPT Tablets* has the dosage form tablets while *Rubimol Juice* is a juice. Interestingly, the second nearest drug to the resulting vector is *Enelfa Dr Henk Suppositories*, which also shares the same active ingredient but is a suppository.

However, we also had some vector operations where the resulting drug had the appropriate alternative dosage forms but did not match in the same medical treatment effect as the supplied dosage form.

5. Discussion

Results from evaluation A indicate that the trained model can provide alternative drugs with high accuracy of over 90%. Also, the combination of drugs and dosage form led to high accuracy values between 68% and 89%. High accuracy values, denoting a meaningful allocation of drugs in an algebraic space, could be confirmed in exemplary visual plots in evaluation B. Therefore, cosine similarities, i.e., dot products, work very well as vector operations on the dataset. Patient information leaflets turn out to be a suitable data source for document processing with unsupervised methods.

In addition to dot products, we tested other vector operations in the form of vector arithmetics, combining word and document vectors. Interestingly, we found evidence that such vector arithmetics are feasible, both for subtracting ingredients and dosage forms. We were even able to add desired ingredients and dosage forms. This gives interesting insights into the compatibility of word vectors and document vectors, a topic that has been previously unexplored.

Regarding the two use cases, we can formulate the following two contributions.

Contribution 1. We were able to show that we could identify alternative drugs not only by A) cosine similarities and B) the visualization of a PCA reduced vector space, but, much more intriguing, also by C) vector operations with two different kinds of vectors, namely adding word vectors to document vectors.

Contribution 2. Similar to contribution 1, we could show that we were able to identify alternative dosage forms not only by A) cosine similarities and B) the visualization of a PCA reduced vector space, but also by C) vector arithmetics with document and word vectors.

However, vector arithmetics did not work on all attempts to include or exclude properties, like subtracting side effects, and also not for every ingredient. We were not able to find general patterns of operations that reliably work well. Whereas vector operations with dot products can be seen as a very solid way to find alternative drugs, the use of vector arithmetics is rather only suitable for exploration purposes. Still, the vector operations we described allow for completely new ways of navigating in the space of drugs, both for patients and pharmacists.

6. Limitations

One of the limitations is that by removing numeric characters and then merging different dosage amounts of the same drug, this information was removed from our data. On the positive side, this removes noise from our data, but on the negative side, gives our model fewer data for embedding, with the effect that our model cannot differentiate on the dosage amount.

Another limitation is that extensive hyperparameter tuning was not done. Instead, the parameters found by Brito et al. [21] with minor adaptations were used. While these parameters represent a good starting point, we think performance can be improved by further optimizing the parameters for this specific task.

Concerning the size of our training set, resulting from the preprocessing steps, the 13,644 patient information leaflets were condensed into 6,488 documents. This is a small dataset compared to the usual data word embeddings are trained on (see [4, 5]). Bigger datasets can help make the embeddings more robust.

Our evaluation of vector arithmetics could only be performed on examples, which is a further limitation of our work. Still, compared to the King-Man-Woman-Queen example, we found some quite intriguing examples of valid vector operations.

Certainly, the model we created does not exhaustively reflect all suitable alternative drugs. The evaluation of suitable alternative drugs is based on the comparison of the active ingredients of the drugs and no expert knowledge is available during the evaluation. There might be drugs that have different active ingredients but still can be considered alternatives by experts between drugs; we cannot assess these because of our lack of expert pharmaceutical knowledge.

7. Conclusion and Outlook

In this research paper, we successfully showed that document and word embeddings from patient information leaflets can be used to find suitable alternative drugs and dosage forms. The model can extract alternative drugs given a specified drug. The training was performed in an unsupervised way, i.e., without a labeled dataset or any insertion of explicit medical knowledge. The resulting model considers the dosage form of the drug when selecting possible alternatives and selects drugs with the same dosage form with a higher probability.

Visualizing the document vectors as a two-dimensional scatter plot revealed that the model clusters drugs with the same effects, for example, painkillers or diabetes medications, together. This is a strong indication that the model works on deeper semantic layers than a purely syntactic extraction can. We also evaluated analogical reasoning and were able to find drugs in a target-oriented way by excluding ingredients through vector arithmetics, demonstrating a completely new way to navigate in the drug realm.

Considering over the counter drugs, pharmacies and health insurance companies could offer a web service for their clients to explore other dosage forms and alternative ingredients, combined with advice to consult a pharmacist as vector arithmetics give indications, but no guarantee of suitable alternatives. Here, service providers would need to check regulatory requirements of their countries to avoid liability issues.

The most intriguing area for further research, from a methodological perspective, is more investigation on the compatibility of word vectors and document vectors. More data sets are needed to verify our initial indication that word vectors can be added or subtracted from document vectors.

With respect to future research in the medical field, there is still room to test further use cases that vector representations of drugs offer. Analogies especially have the potential to create new ways of querying drugs. Also, the data that the model is trained on can be extended, for example, user-generated content from medical forums.

8. Acknowledgements

We gratefully acknowledge the support of the German Research Foundation (DFG), project FOR 1452 / TP1 SCHO 1321/3-2. We also thank the ABDATA Pharma Data Service for providing the data used in this study.

9. References

[1] ABDA, “Statistik”, https://www.abda.de/aktuelles-und-presse/zdf.

[2] Vinker, S., V. Eliyahu, and J. Yaphe, “The effect of drug information leaflets on patient behavior”, *IMAJ-RAMAT GAN- 9*(5), 2007, pp. 383.

[3] Le, Q., and T. Mikolov, “Distributed representations of sentences and documents”, *Int. Conf. on Machine Learning*, PMLR (2014), 1188–1196.

[4] Mikolov, T., K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space”, *ICLR*, (2013).

[5] Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality”, *Adv. in neural inf. processing systems*, (2013), 3111–3119.

[6] Tshitoyan, V., J. Dagdelen, L. Weston, et al., “Unsupervised word embeddings capture latent knowledge from materials science literature”, *Nature 571*(7763), 2019, pp. 95–98.

[7] Pennington, J., R. Socher, and C.D. Manning, “Glove: Global vectors for word representation”, *Conf. on empirical methods in natural language processing (EMNLP)*, (2014), 1532–1543.

[8] Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information”, *Transactions of the Association for Computational Linguistics 5*, 2017, pp. 135–146.

[9] Peters, M.E., M. Neumann, M. Iyyer, et al., “Deep contextualized word representations”, *Proc. of NAACL*, (2018).

[10] Reimers, N., and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”, *Conference on Empirical Methods in NLP and the Int. Joint Confe. on NLP (EMNLP-IJCNLP)*, Association for Computational Linguistics (2019), 3982–3992.

[11] Chen, D., J.C. Peterson, and T.L. Griffiths, “Evaluating vector-space models of analogy”, *arXiv:1705.04416*, 2017.

[12] Goldberg, Y., and O. Levy, “word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method”, *arXiv:1402.3722*, 2014.

[13] Rong, X., “word2vec parameter learning explained”, *arXiv:1411.2738*, 2014.

[14] Risch, J., and R. Krestel, “Domain-specific word embeddings for patent classification”, *Data Technologies and Applications*, 2019.

[15] Lee, D., R. Cornet, F. Lau, and N. De Keizer, “A survey of SNOMED CT implementations”, *J. of biomedical informatics 46*(1), 2013, pp. 87–96.

[16] Mahoney, A., and J. Evans, “Comparing drug classification systems”, *AMIA*, 2008, pp. 1039.

[17] Azevedo, A.I.R.L., and M.F. Santos, “KDD, SEMMA and CRISP-DM: a parallel overview”, *IADS-DM*, 2008.

[18] ABDATA, “ABDATA Pharma-Daten-Service”, *ABDATA*. https://abdata.de/

[19] Bergstra, J., and Y. Bengio, “Random search for hyper-parameter optimization.”, *Journal of machine learning research 13*(2), 2012.

[20] Lau, J.H., and T. Baldwin, “An empirical evaluation of doc2vec with practical insights into document embedding generation”, *arXiv:1607.05368*, 2016.

[21] Brito, E., R. Sifa, K. Cvejoski, C. Ojeda, and C. Bauckhage, “Towards German word embeddings: A use case with predictive sentiment analysis”, In *Data Science–Analytics and Applications*. Springer, 2017, 59–62.

[22] Baroni, M., G. Dinu, and G. Kruszewski, “Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors”, *Annual Meeting of the Association for Computational Linguistics*, (2014), 238–247.

[23] Schnabel, T., I. Labutov, D. Mimno, and T. Joachims, “Evaluation methods for unsupervised word embeddings”, *Conf. on empirical methods in natural language processing*, (2015), 298–307.

[24] Apotheke Adhoc, “Diese Wirkstoffe werden am häufigsten verordnet”, *APOTHEKE ADHOC*. https://www.apotheke-adhoc.de/nc/mediathek/detail/wirkstoff-ranking-das-sind-die-am-haeufigsten-verordneten-arzneistoffe/