# Learning Patient Similarity Using Joint Distributed Embeddings of Treatment and Diagnoses

**Christopher Ormandy**
Department of Informatics
King's College London
christopher.ormandy@kcl.ac.uk

**Zina M. Ibrahim**
Department of Biostatistics
& Health Informatics
King's College London
zina.ibrahim@kcl.ac.uk

**Richard JB Dobson**
Department of Biostatistics
& Health Informatics
King's College London
richard.j.dobson@kcl.ac.uk

## Abstract

We propose the use of vector-based word embedding models to learn a cross-conceptual representation of medical vocabulary. The learned model is dense and encodes useful knowledge from the training concepts. Applying the embedding to the concepts of diagnoses and medications, we then show that they can then be used to measure similarities among patient prescriptions, leading to the discovery of in- formative and intuitive relationships between patients.

## 1 Introduction

In simple word representation techniques such as the Ngram model [Brants *et al.*, 2007], words are regarded as single atomic units, and no notion of similarity between words exists. Conversely, distributed word representations in vector space provide an explicit grouping of similar words to achieve high performance in Natural Language Processing tasks [Rumelhart *et al.*, 1988]. Such *embeddings* rely on vector operations to represent *learned* word proximities or similarities [Mikolov *et al.*, 2013] and have have been used to efficiently learn high-quality word vectors from very large datasets (containing billions of words) using a vocabulary containing millions of words [Collobert and Weston, 2008; Bengio and Usunier, 2011; Socher *et al.*, 2011; Glorot *et al.*, 2011; Turney and Pantel, 2010; Turney, 2013].

Recently, [Mikolov *et al.*, 2013] has introduced a neural network design using distributed word representations to capture interesting features such as linguistic regularities and patterns. The architecture, named the Skip-Gram model, is trained to find word representations of a given (input) word that are useful in predicting its surrounding words in a sentence or a document. The vector representation used in the Skip-Gram model highly increases the network's training efficiency, with the ability to train 100 billion words in single optimized machine [Mikolov *et al.*, 2013].

Our idea lies in using a Skip-Gram model to learn a compact representation of *patient* features. Using an initial model with medications and diagnoses as features, we propose a scheme to embed top-level ICD 9 codes of patient prescriptions and diagnoses within the same continuous representational space. We then build a skip-gram representation using the chosen system to create a compact and continuous representation of patients enabling: 1) efficient feature processing and 2) some degree of generalization in finding similarity between patients given their features. Using our model, we would be able to reach the natural conclusion of a patient diagnosed with Diabetes being similar to a patient receiving insulin treatment. This is a non-trivial exercise for a machine learning algorithm, as we understand that the two cases are to some degree the same abstract concept expressed across two different domains (diagnoses vs. treatment).

The paper is structured as follows. After a brief illustration of the required background in Section 2, we discuss our architecture in Section 3. In Section 4, we show the results of training the resulting neural network model on a large database of intensive care unit medical records. We conclude with ongoing work and future directions in Section 5.

## 2 Background

### 2.1 Vector-based Word Representation

A well-established approach for representing concepts to facilitate learning is the use of a fixed dimension, real valued vector representing words. Each entry of this vector corresponds to some feature in a hypothetical latent space, rendering the size of the vector to be the dimensions of the feature space used to represent a single word.

For example, creating a 5-dimensional representation of prescriptions such as Aspirin, Ibuprofen, and Insulin, we could decide on features such as "Heart problems," "Pain killer," "Kidney Problems," "Critical Importance medication" and "Preventative treatment." In this example, Aspirin would rank moderately for "Heart," quite highly "Pain killer," relatively lowly for "Kidney Problems," perhaps low to moderately for "Critical Importance" and moderately to high on "Preventative." Normalizing the values of an arbitrary patient (by assuming a vector length of unity) gives the vector shown in Table 1.

In practice, we do not suggest the nature of each feature, but merely supply the number of them - a neural network or another approach then learns these features so as to serve its needs best. However, the basic premise is the same - each feature has some meaning in the hypothetical latent space learned by the network, and so similar values in the same position indicate two samples both share some aspect of this

| Drug | Heart | Pain killer | Kidney Problem | Critical Importance | Preventative |
|---|---|---|---|---|---|
| Aspirin | 0.57 | 0.74 | 0.04 | 0.12 | 0.33 |
| Insulin | 0.07 | 0.07 | 0.64 | 0.54 | 0.54 |
| Ibuprofen | 0.1 | 0.99 | 0.05 | 0.05 | 0.05 |

Table 1: Example of normalized manual encoding

feature. Examples which share a large number of features are therefore closer than those which share only a few, as a consequence of this encoding, which is the mechanism by which similarity is explicitly encoded as Table 2 shows.

| Drug Pair | Similarity |
|---|---|
| Aspirin - Insulin | 0.36 |
| Aspirin - Ibuprofen | 0.82 |

Table 2: Example of embedding similarity

## 2.2 Learning via the Skip-Gram Model

The Skip-Gram model is based on the goal of finding word representations that would enable the prediction of surrounding words of a given word in a sentence. The idea is for any 'candidate' word found in the training vocabulary; we can associate the most likely 'context' word such that the two words show the maximum association.

Formally, given a sequence of words $w_1, w_2, ..., w_n, ..., w_N$, the Skip-Gram model will train a multi-class logistic regression so that for each candidate word $w_n$, we can find a 'context word' $w_n + j$ falling within the window of $c$ words before or after $w_n$ such that the probability of $P(w_{n+j}|w_n)$ is maximum [Mikolov *et al.*, 2013]. In other words, the Skip-Gram model aims to maximum the average log probability:

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{-c\leq j\leq c, j\neq 0} logP(w_{n+j}|w_n)$$

$c$ is the size of the training context and is used to adjust the model. Larger $c$ values associate a wider context with a given candidate word, implying more training examples, slower training but better classification.

# 3 Our Work: A Patient-Focused Skip-Gram Model

The work performed here is based on the idea of generalising vector-based embeddings to any number of medical concepts, regardless of whether or not they come from the same underlying distribution. The main implication of this is that the features potentially become more general or invisible to us. However, with related domains such as diseases and drugs, we could imagine a normalized encoding as given in figure 1 for features spanning the two concepts of disease and medication.

## 3.1 The Skip-Gram Model

The details of our implementation are largely based on the skip-gram model [Mikolov *et al.*, 2013; Rumelhart *et al.*, 1988] and is shown in Figure 1. The implemented logistic regression classifier receives as input an ID corresponding to an item in our vocabulary (in this case a list of all the ICD 9 codes for diagnoses and Medications). This ID corresponds to the Drug Embedding, which is a row within our Embedding matrix. Using the $embedding\_lookup(...)$ functionality in tensorflow, we retrieve the 100-dimensional Embedding for the input, multiply it by a weight vector and pass through a softmax function. The input-output pairs are created to be all permutations of pairs that appear together in the same set. For example, if a patient was prescribed medications A, B, and diagnosis D, we create input-output pairs as: (A, B), (A, D), (B, A), (B, D), (D, A). We aggregate all these input-output pairs across all patients in the training set and use them to perform mini-batch back-propagation on the embedding matrix and logistic regression parameters simultaneously. As proposed by Mikolov et al. [Mikolov *et al.*, 2013], we use Noise Contrastive Estimation to approximate the loss at each step of training, to improve the efficiency of computation, and built our model in tensorflow.

## 3.2 Patient Similarity Using Unsupervised Embeddings

Using the unsupervised joint embeddings, we show that meaningful patient similarities can be discovered within the data. To do this, we train the prescription and diagnosis joint embeddings in the manner described in the previous section, on a subsection of the data (100,000) prescriptions, and then draw patients randomly from the remaining portion of the data. We then aggregate all the prescriptions given on a daily basis to the patient during their stay and replace each one with the relevant embedding trained previously.

To generate a treatment vector, we average all the individual drug embeddings for each day. By then taking a single days treatment vector, and computing the cosine similarity between that and other daily treatment vectors, we can find the similarity between patient treatments.

This is a well-established trick in NLP and is often a primary benchmark to compare other methods against, and while it may not seem like the most sophisticated solution, it can be surprisingly effective.

# 4 Experiments & Results

## 4.1 Data Source & Preprocessing

The model was trained using the MIMIC dataset [Johnson *et al.*, 2016]. This is a large Intensive Care Unit (ICU) dataset containing the records of over 40,000 patients in the ICU

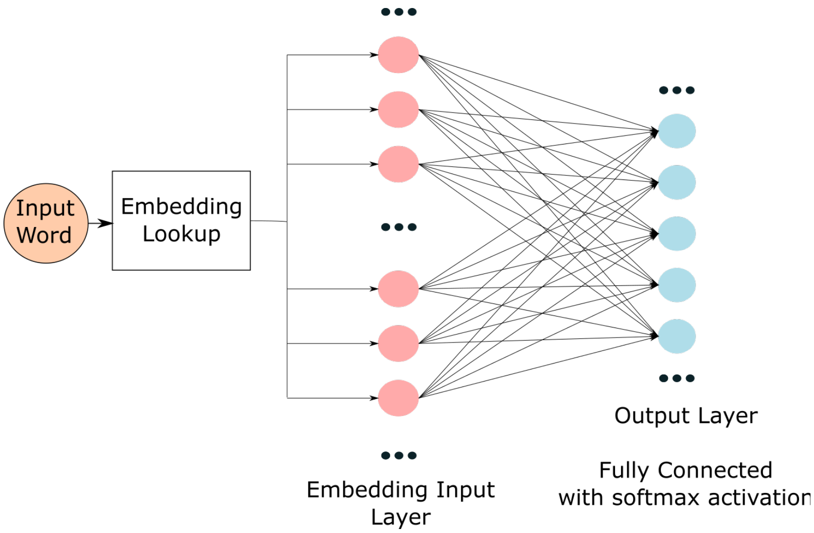


Figure 1: The Skip-Gram Architecture Used

of the Beth Israel Deaconess Medical Center, Boston, Massachusetts, U.S.A. between 2005 and 2012. Prescriptions are registered alongside a unique and anonymized patient identifier, with a date range indicating the period this was to be administered over.

To train a neural network on patient prescriptions, one must first extract the data and reshape it, which is a non-trivial task for the way the data is presented in MIMIC iii. As shown in Figure 2, prescriptions are primarily indicated by a combination of hospital admission id, start date, end date and drug.

### 4.2 Tensorflow Implementation

The first step was to aggregate all the drugs by day and hospital admission id, to compile a list of concepts to be used per day, as shown in Figure 3. Each day defines a context window for that patient, so if a patient received drugs A, B and C on a given day, the input output pairs for the network are (A, B), (A, C), (B, A), (B, C), (C, A) and (C, B).

Next, we assign each concept an arbitrary ID, with 0 reserved for an 'unknown' entry. This allows unseen concepts to be included after training time. Each ID maps to a row in a randomly initialized embedding matrix, which has dimensions (number of drugs x embedding size). This embedding matrix is then used as inputs to logistic regression classifier, which performs a one-hot prediction for the output concept, with size (number of drugs,). This is displayed mathematically in 2.

$$E = embedding\_lookup(X) \tag{1}$$

$$\hat{y} = softmax(E \cdot W + b) \tag{2}$$

This system is trained via back propagation, and it simultaneously learns both the W and b parameters and the values of the embedding matrix. Once training is complete, the embedding matrix acts as a lookup dictionary - to get the representation for a particular drug, simply find the ID it maps to and extracts this row from the embedding matrix. I used the standard Adam as the optimization method and negative log likelihood for the loss function.

In tensorflow, we initialized the weights randomly, with a truncated normal distribution for weights and a random uniform for embeddings and biases. This is based upon conventional methodologies found to be most useful in a wide range of settings, as described in [LeCun *et al.*, 2012].

Following from [Mikolov *et al.*, 2013], I use Noise contrastive estimation to improve the efficiency of the model. As the model has many outputs (one for each entry in the 'vocabulary'), computing the softmax at each stage is computationally expensive. As most of the entries are in fact not relevant (we have many classes, but most should be 0, and we want only a single entry that is substantially non-zero), we can improve the computation efficiency by sampling the loss function rather than computing it exhaustively. There are two ways to achieve this in practice, one is with a sampled softmax, which essentially computes a Monte Carlo estimate, and Noise contrastive estimation which picks examples of the positive and negative classes so as to get an estimate that way.

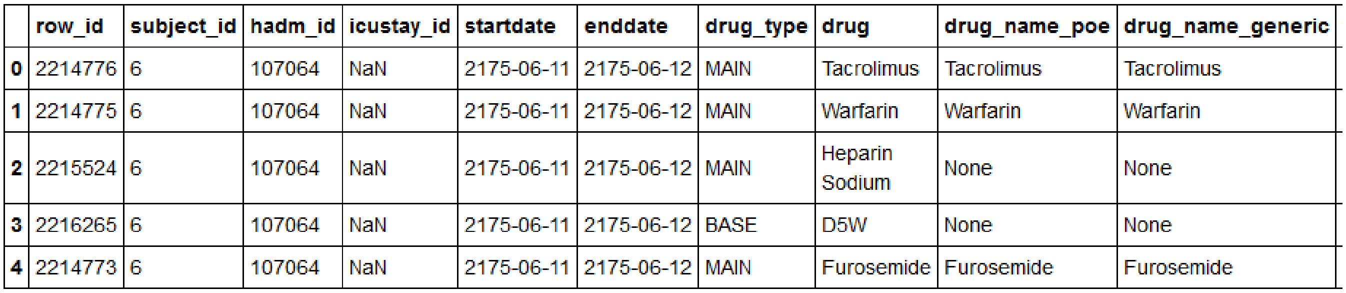
### 4.3 Results

#### Evaluating Prescription Embeddings

As this is an unsupervised approach, quantitative evaluation of the results is difficult. To assess if the neighbourhoods are correct, most previous work either appeals to experts to evaluate the quality or avoids this altogether and leaves the reader to judge for themselves [Mikolov *et al.*, 2013].

To provide a qualitative evaluation of the results, we took the top occurring drugs and found the nearest neighbours to them using cosine similarity as a measure.

These nearest neighbour relationships show some useful similarity between drugs. For example, we see salts and electrolytes naturally grouping together (e.g. Potassium Chloride and Magnesium Sulfate). Aspirin is close to two statins - drugs which try to treat blood pressure and alleviate the risks of heart attack or similar problems. Metoclopramide is used

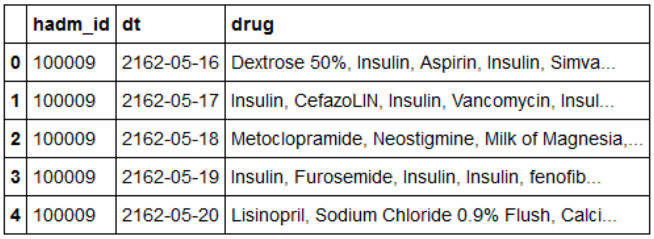


| | row_id | subject_id | hadm_id | icustay_id | startdate | enddate | drug_type | drug | drug_name_poe | drug_name_generic |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2214776 | 6 | 107064 | NaN | 2175-06-11 | 2175-06-12 | MAIN | Tacrolimus | Tacrolimus | Tacrolimus |
| 1 | 2214775 | 6 | 107064 | NaN | 2175-06-11 | 2175-06-12 | MAIN | Warfarin | Warfarin | Warfarin |
| 2 | 2215524 | 6 | 107064 | NaN | 2175-06-11 | 2175-06-12 | MAIN | Heparin Sodium | None | None |
| 3 | 2216265 | 6 | 107064 | NaN | 2175-06-11 | 2175-06-12 | BASE | D5W | None | None |
| 4 | 2214773 | 6 | 107064 | NaN | 2175-06-11 | 2175-06-12 | MAIN | Furosemide | Furosemide | Furosemide |

Figure 2: Example format of prescriptions

| Drug | Nearest Neighbour | 2nd Nearest Neighbour |
|---|---|---|
| Potassium Chloride | Magnesium Sulfate | Calcium Gluconate |
| Morphine Sulfate | Acetaminophen | Oxycodone-Acetaminophen |
| Docusate Sodium | Sodium Chloride 0.9% Flush | Acetaminophen |
| Calcium Gluconate | Potassium Chloride | Magnesium Sulfate |
| Aspirin | Simvastatin | Atorvastatin |
| Metoclopramide | Ranitidine | Nitroglycerin |
| Amiodarone HCl | D5W (EXCEL BAG) | Phenylephrine HCl |
| Heparin Sodium | Warfarin | Ibuprofen |

Table 3: Nearest Neighbours for Drug Embeddings



| | hadm_id | dt | drug |
|---|---|---|---|
| 0 | 100009 | 2162-05-16 | Dextrose 50%, Insulin, Aspirin, Insulin, Simva... |
| 1 | 100009 | 2162-05-17 | Insulin, CefazoLIN, Insulin, Vancomycin, Insul... |
| 2 | 100009 | 2162-05-18 | Metoclopramide, Neostigmine, Milk of Magnesia,... |
| 3 | 100009 | 2162-05-19 | Insulin, Furosemide, Insulin, Insulin, fenofib... |
| 4 | 100009 | 2162-05-20 | Lisinopril, Sodium Chloride 0.9% Flush, Calci... |

Figure 3: Aggregated daily prescriptions

to treat acid reflux, a stomach complaint, and Ranitidine is used to reduce the amount of stomach acid produced.

We also see relationships between items that often appear together even if they are not direct replacements. For example, Amiodarone HCL is an antiarrhythmic drug, used to treat issues with irregular heartbeats, and its nearest neighbour is D5W. D5W is a code for Dextrose 5% and water, which is essentially just a carrier for IV lines and similar methods of delivery. These two are near as it is common within the data to administer Amiodarone HCL as a solution with D5W.

### Joint Embeddings

As with the prescription only embeddings, proving these encode useful information in a quantitative way is somewhat complicated. We follow the same approach as the previous section and provide some of the nearest neighbours for common entries in the data, and also, in the next section, show that these embeddings are useful for the task of finding patients with similar treatments, as a way to demonstrate that they encode relevant information.

As can be seen in table 4, the approach of using joint embeddings encodes the same relevant information seen in the results for single embeddings, while also providing links between diagnoses codes and drugs. For more broad ranging drugs, such as painkillers, we see a clustering that is not particularly associated with a single ICD9 code, for example, Bisacodyl is close to Docusate Sodium and Morphine Sulphate. This also shows another interesting artifact of this method - docusate sodium is not a painkiller, but is 'close' to bisacodyl because they often appear together. Acetaminophen, Meperidine, and Morphine Sulfate are another cluster of pain relief medications which do not appear 'close' to a particular ICD9 diagnosis code.

We see interesting clustering of ICD9 codes - 427, 428 and 414 all representing heart problems for example. We also see cross group clusters, which put Diabetes and Insulin close together, as well as Aspirin and heart disease.

### Evaluating Patient Similarity

Finding patients who shared a similar daily treatment vectors worked well to find patients of similar types. Due to the nature of the ICU, many patients received a large number of drugs, and using embeddings rather than a one hot style approach allows for meaningful entries to be more discriminative. We selected patients at random, and then picked a random day for that patient, and computed the cosine similarity between that daily treatment vector and all other daily treatment vectors for all patients. As expected, other days from that patients stay in the ICU rank very highly in many cases. However, even if we look only at other patients, we see meaningful groupings occurring. Some examples are included in Table 5. Similarities are Cosine similarities of normalised vectors, and so they vary between 100% and -100%. A similarity of 100% means the same, while -100% indicates

| Entry | Nearest Neighbour | 2nd Nearest Neighbour |
|---|---|---|
| Bisacodyl | Docusate Sodium | Morphine Sulfate |
| Calcium Gluconate | SW | D5W |
| Acetaminophen | Meperidine | Morphine Sulfate |
| Insulin | 250 (Diabetes mellitus) | Tamsulosin HCl |
| 427 (Cardiac dysrhythmias) | 428 (Congestive heart failure) | 414 (Other forms of chronic ischemic heart disease) |
| 276 (Disorders of fluid electrolyte) | 530 (Diseases of esophagus) | 790 (Nonspecific findings on examination of blood) |
| 401 (Essential hypertension) | 746 (Other congenital anomalies of heart) | 272 (Disorders of lipoid metabolism) |
| Aspirin | Clopidogrel Bisulfate | 414 (Other forms of chronic ischemic heart disease) |
| Pantoprazole Sodium | Iso-Osmotic Sodium Chloride | Magnesium Sulfate |
| Morphine Sulfate | Acetaminophen | Oxycodone-Acetaminophen |
| Heparin | Guaifenesin | Senna |
| Lorazepam | Chlorhexidine Gluconate | Diphenhydramine HCl |
| Metoprolol | Cefazolin | Warfarin |
| 250 (Diabetes mellitus) | Insulin | 414 (Other forms of chronic ischemic heart disease) |

Table 4: Nearest Neighbours for Drug & Diagnosis Embeddings

complete opposites. As such, we can show example pairs which are similar, and those which are also not similar.

As shown in Figure 4, if we conduct PCA on the 100-dimensional embeddings to reduce the dimensionality down to two principle components, we can visualise the 'closeness' of a small number of entries. The figure shows that the selected entries fall into 3 or 4 clusters. Diabetes is close to Insulin, which two neonatal drugs (denoted by NEO*) are close to the diagnosis ICD 9 code for electrolyte imbalance - a condition that strongly associates with newborns in the dataset. The other cluster corresponds to pain medication (and docusate sodium which is used to alleviate constipation, a common side effect of many pain medications). This cluster seems to split into two subclusters, potentially corresponding to differing use of the different drugs, but likely an artifact of the dimensionality reduction.

## 5 Conclusions and Ongoing Work

The idea and initial results presented here are part of our ongoing work of finding compact representations of patients using their electronic hospital records, to make use of the massive deluge of longitudinal data available to understand the densest set of features representing a given patient. Our long-term goal is to complement endeavors in personalized medicine by devising patient similarity measures that can be used to supplement scientific inquiry and which can translate into actionable knowledge in the bedside. For instance, we would like to be able to answer questions such as Given that patient *X* is similar to patient *Y*, will *X* respond to treatment *Z* similarly to patient *Y*? Or why has patient *A* responded differently than patient *B* to the same treatment provided?

## 6 Acknowledgements

The authors would like to acknowledge the supports of the NIHR Biomedical Research Centre for Mental Health, the Biomedical Research Unit for Dementia at the South London, the Maudsley NHS Foundation Trust and Kings College London, and a joint infrastructure grant from Guys and St Thomas Charity and the Maudsley Charity, London, United Kingdom.

This research was also supported by researchers at the National Institute for Health Research University College London Hospitals Biomedical Research Centre, and by awards establishing the Farr Institute of Health Informatics Research at UCLPartners, from the Medical Research Council, Arthritis Research UK, British Heart Foundation, Cancer Research UK, Chief Scientist Office, Economic and Social Research Council, Engineering and Physical Sciences Research Council, National Institute for Health Research, National Institute for Social Care and Health Research, and Wellcome Trust.

## References

[Bengio and Usunier, 2011] Jason Weston Samy Bengio and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *International joint conference on Artificial Intelligence (IJCAI)*, page 27642770, 2011.

[Brants *et al.*, 2007] T Brants, C Popat, F Xu, J Och, and J Dean. Large language models in machine translation. In *the Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning*, 2007.

[Collobert and Weston, 2008] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning.

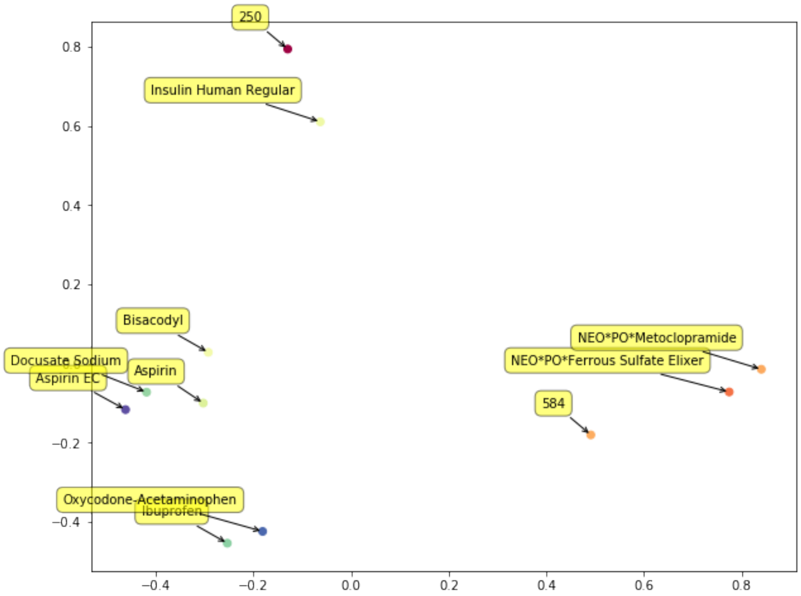


Figure 4: PCA on selected embeddings

| **Patient 1 Diagnoses** | **Patient 2 Diagnoses** | **Treatment Similarity** |
|---|---|---|
| Newborn | Newborn | 99% |
| Stab Wounds | Motor Vehicle Accident | 89% |
| Pneumonia | Hypoxia | 85% |
| Diaherria | Fever | 85% |
| Urinary Tract Infection | Renal Failure | 78% |
| Encephalopathy | Congestive Heart Failure | -41% |
| Incarcerated Hernia | Sepsis | -40% |
| Newborn | Arterial Injury | -35% |

Table 5: Similarity between selected daily treatment vectors

In *International conference on Machine learning*, page 160167, 2008.

[Glorot *et al.*, 2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *International Conference on Machine Learning (ICML)*, page 513520, 2011.

[Johnson *et al.*, 2016] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3, 2016.

[LeCun *et al.*, 2012] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[Rumelhart *et al.*, 1988] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[Socher *et al.*, 2011] Richard Socher, Cliff Lin, Andrew Ng, and Christopher Manning. Parsing natural scenes and natural language with recursive neural networks. In *International Conference on Machine Learning (ICML)*, 2011.

[Turney and Pantel, 2010] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

[Turney, 2013] Peter Turney. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *Transactions of the Association for Computational Linguistics (TACL)*, page 353366, 2013.