Word Embeddings for Clinical Systems

Hathaitorn Rojnirun
hr346@cornell.edu

Oluseye Bankole
ob97@cornell.edu

Abstract

In this paper, we evaluate a baseline word embedding model for a set of clinical notes derived from patient records. For our baseline, we extract features for this embedding using the **Word2Vec** module from the gensim package. We also build two models, a word2vec skipgram model with negative sampling and a positive pointwise mutual information (PPMI) model by training on the processed clinical notes. Our evaluation shows that both the PPMI and the skipgram models show improved results for medically-related terms when compared with the baseline model. PPMI shows the best result out of all three models.

1 Introduction

Word embeddings are incredibly powerful modeling and feature engineering techniques that allow us to perform very advanced NLP tasks. More traditional NLP techniques like Bag-Of-Words (BOW) and n-gram models, even though simple and powerful, have their limitations for NLP tasks like sentiment classification and part-of-speech tagging. Because word embeddings map words from a single-word dimension space to a real-value multi-dimension space, this new representation allows one to perform more complex NLP tasks that involve words and distributed contexts. Mikolov et al [1] implemented an efficient model for **Word2Vec**, a word embedding model that was trained on a billion words and their embeddings have now become a widely popular model for several NLP tasks.

There have been many efforts to employ word embedding engineering in other domains. One of such applications is in the delivery of clinical medicine. With the growing penetration of EHR systems which are deeply rich in textual data, there is an opportunity to develop smarter EHR systems. Specifically, by learning clinical word embeddings for use in EHR systems, we might be able to develop better tools for clinician decision support, patient text summarization, bill coding, information retrieval, all of which can improve quality outcomes for the patient. Given the sensitive nature of health-care data, there is a dearth of publicly-available datasets to encourage open research in this area.

1.1 Related Work

Despite this limitation, multiple authors have conducted promising research in this area. Choi et al [2] compare results of embeddings generated from three different datasets for medical relatedness and concept similarity. Their datasets were from medical journal abstracts, private medical claims and one publicly-available de-identified EHR data. Their evaluation shows that medical journal embeddings generally show strongly-correlated concepts than clinical narratives from EHR systems. As a result, they opine that the data source would greatly impact the results of the end NLP task.

Similarly, Asgari and Mofrad [3] also develop similar embeddings for proteins **(ProtVec)** and genes **(GeneVec)**. Given an unstructured dataset of protein sequences, they were able to successfully generate the right sequences using their pre-trained sequence embeddings.

On another note, Dubois and Romano [4] also describe a number of effective techniques for learning clinical embeddings. In their paper, they create multi-level embeddings, first at the word level, then at the note level and finally at the patient level. They evaluated the embeddings generated using multiple learning techniques at each level and found mixed results. Notwithstanding,

Query Concepts	Top-5 Similar Concepts	Similarity Scores
schizophrenia	personal history of schizophrenia	0.523
	antipsychotic agents	0.491
	selenium	0.491
	clozapine	0.484
	panic attacks	0.479
movement disorders	tetrabenazine	0.539
	muscle hypertonia	0.477
	orthostasis	0.467
	dystonia 1, torsion, autosomal dominant	0.462
	red man syndrome	0.458
headache	menstrual headache	0.508
	syringomyelia	0.429
	angioma, cavernous	0.427
	potassium chloride ...	0.417
	pupils equal round ...	0.416

Table 1: Sample queries and their top-5 similar concepts along with the corresponding similarity scores. The cosine distance was used as the similarity measure.

their results showed that word embeddings generated from clinical notes can be used to predict certain medical events such as death, admission or E.R visits.

2 Model Descriptions

2.1 Baseline Model

In order to generate continuous vector representations of words, we utilized the **Word2Vec** model provided by the Gensim library. The model consists of a single hidden layer, fully connected neural network. We varied the embedding size, which can be specified via the constructor of the **Word2Vec** model.

Similar to the probabilistic feed-forward neural network language model (NNLM), the Continuous Bag-of-Words (CBOW) model consists of input, projection, hidden and output layers. However, for CBOW, the projection layers is shared by all the input words [6], resulting in a desirable property for our task as past word orderings do not influence the projection.

As a preprocessing step, we employed a simple strategy of forcing logical groupings among words of each input medical phrase by replacing blank spaces with underscores. This approach has shown a promising as explained in Section 3.2.

2.2 Skigram Model with negative sampling

For the alternative models, we developed a skip-gram model using negative sampling. This method has been described in great detail in the paper by Mikolov et al [1] as it is a more computationally efficient method, allowing us to only update very specific weights. We use the Dynet Framework for model implementation.

2.3 Positive Point-wise Mutual Information (PPMI)

Unlike the other two methods described earlier, this method is less computationally expensive, which was especially beneficial since the data has been preprocessed. It relies on counting and estimating the joint and individual probabilities of word pairs and words respectively. A few authors (Keerthi et al [7] and Levy and Goldberg [8]) describe this method in their paper and establish the similarity between another variant of this method Shifted Positive Point-wise Mutual Information and Negative Sampling with Stochastic Gradient Optimization.

3 Data Description

The dataset was derived from the Stanford Translational Research Integrated Database Environment (STRIDE) dataset - a publicly-available electronic health record including doctors clinical notes of patients over a nineteen-year period. We use the preprocessed form of the dataset released by Finalyson et al [5]. Their dataset consists of co-occurrence frequencies of pair-wise terms and concepts in 1, 7, 30, 90, 180, 365 and infinity

day bins. The terms were culled from the notes and mapped to medical concepts from the Unified Medical Language System (UMLS). For all our experiments, we only use the 1-day bin occurrence dataset. Since the pre-processed data only shows the medical codes for the concepts and terms, the data had to first be decoded using an external resource (SNOMED, UMLS) before it could be passed through word embedding packages like Gensim (Baseline model).

For all the alternative word embedding models that we developed, we employ a word vector representation following the skipgram methodology. To do this, we use the concept unique identifiers (CUI) code representations in the co-occurring matrix as representations of words and contexts for our skipgram model.

4 Experimental Setup

In this section, we describe our experimental setup for constructing the baseline model, provide the implementation details and discuss some of our findings.

4.1 Baseline Implementation

In order to implement the baseline model, we utilized the Gensim library in Python. The current application is capable of producing the following results:

- Given two medical concepts present in the input medical corpus, return their similarity value.
- Given a medical concept, retrieve the top-5 most similar concepts from the input medical corpus.

We employed the Continuous Bag of Words (CBOW) model, treating the two medical concepts as a context for one another. The size of the embeddings was set to 300 to facilitate the comparison between our baseline model and the ones mentioned in [2], while the window size, which dictates the distance between the current word and the predicted word, was set to 2.

We trained the CBOW model on the entire medical dataset, which consisted of 8,555,210 records. We utilized an iterator to feed medical records to the **Word2Vec** model, iteratively updating it. Without the iterator, the CPU ran out of memory after approximately five million records. The number of worker threads was set to 8 to match the number of CPU cores.

In order to improve on the baseline model, we also trained Word2Vec with different combinations of window sizes (2 and 5) and embedding dimensions (100, 200, and 300).

4.2 Alternative Model Implementation

We attempted a number of alternative models that attempt to learn directly from the co-occurrence matrix. From the pre-processed dataset, there are about 8.5 million co-occuring terms. If the frequencies of these co-occurring terms are considered, there are about 13.3 billion terms/concepts across all co-occurring pairs. Since this would be almost too expensive to compute, we set out to compare try different data management strategies:

- In the first case, we assume that all the co-occurring terms occur exactly once.
- In the next case, we assume that the frequencies of the co-occurring terms can be approximated by the natural logarithm of their frequency counts. This assumption while modest, still minimizes the amount of training data points in this model. For example, for concept/term pairs that occur a million times, they will be approximated by the floor value of log(1000000)), which equals 13 counts.
- In our final implementation, we introduce Positive Point-wise Mutual Information. This methods uses the probabilities of the co-occurring pairs of words to determine word similarity across the vocabulary space. We used SVD to truncate this large dimensionality into different word embedding sizes.

4.3 Evaluation and Testing

We performed two forms of evaluation on the experimental results. The first was a cursory, crude search to check if the word embedding representations made sense. The second approach was more objective, where we utilize UMLS, an external medical resource to evaluate the embeddings for their effectiveness in predicting medical relatedness for different medical terms or concepts. The medical terms are derived from a publicly-available document about drugs known to either prevent **(May Prevent)** or cure **(May Treat)** certain diseases. See Table 3 and Table 4 for exam-

	Context	Embedding Size	Window/NS SIze	May Treat (%)	May Prevent (%)
Baseline	1 day bin, 1 count	300	2	4.98	4.41
	1 day bin, 1 count	100	2	5.95	**4.78**
	1 day bin, 1 count	200	2	6.93	4.04
	1 day bin, 1 count	100	5	**8.87**	4.04
	1 day bin, 1 count	200	5	2.49	1.84
	1 day bin, 1 count	300	5	1.19	1.47
	1 day bin, 1 count	200	3	4.44	3.31
SGNS	1 day bin, 1 count	100	2	16.56	**8.82**
	1 day bin, 1 count	100	5	13.96	5.51
	1 day bin, log count	100	2	**17.75**	7.72
	1 day bin, log count	300	5	13.42	7.72
PPMI	1 day bin, all counts	100	-	15.80	7.35
	1 day bin, all counts	200	-	16.77	7.72
	1 day bin, all counts	300	-	**17.97**	**8.46**

Table 2: Medical Relatedness score for different experimental strategies.

Drug Name	May Treat
zidovudin	AIDS
	HIV infekce
alemtuzumab	chronick lymfatick leukemie
melatonin	jet lag syndrom
sitagliptin	diabetes mellitus
valdecoxib	revmatoidn artritida
	bolest
	osteoartrza
	dysmenorea

Table 3: May treat examples: paired associations between drugs and diseases they may treat

Drug Name	May Prevent
dolasetron	zvracen
	nauzea
fiber	zcpa
hydroxokobalamin	nedostatek vitaminu B12
chlorid draseln	hypokalemie
eucerin	dermatitida
	pruritus

Table 4: May prevent examples: paired associations between drugs and diseases they may prevent

ples. This document shows a one to many relationship between the drugs and the diseases. We map these drugs against the drug occurrences in our word embedding to extract a subset of the embeddings. Next, we do a cosine similarity to match between this subset embedding and the entire embedding matrix and find the top 40 neighbors for each drug. The number of neighbors is a hyperparameter that we have not tuned in this case. Choosing a larger number may be too leaky and a smaller number will be uninformative for the medical terms. We have chosen 40 neighbors, as used in the paper by Choi et al [2]. We take the results of the top 40 neighbors for each drug and evaluate the number of disease hits, that is how many of these top results contain information about the diseases in the **May Prevent** and **May Treat** documents.

4.4 Experimental Results and Future Work

In this section, we discuss the results obtained from our baseline and alternative model experimental runs. As a sanity check, we evaluate the results of the baseline model to see how well it is able to predict co-similar terms. We present sample top-5 retrieval results in Table 1. It seems that these results make sense intuitively. For instance, for the query word *schizophrenia*, the medical concept with the second highest similarity score is *selenium*. According to the Journal of Orthomolecular Medicine, selenium deficiency is potentially a risk factor in schizophrenia. The model works well not only on single-word inputs but also medical phrases. With the query *movement disorders*, the model returns *muscle hypertonia* (upper motor neuron lesions) with the second highest similarity score.

Additionally, we checked the sensitivity of the model to dissimilar terms. To do this, we picked a fixed medical concept, *tobacco* and then four medical concepts from the input dataset (some related, others unrelated). We see from Table 5 that the results again make intuitive sense. The concepts *tobacco* and *smoker* are highly related while *tobacco* and *sugar* are not.

For our subsequent analysis, we evaluated the models for medical-relatedness using the method described in the Evaluation and Testing section. We see from Table 2 that our ability to predict relationships between drugs and diseases in the **May Treat** case improved incredibly when we moved from the baseline model to both PPMI and skipgram alternative models. There are observable differences in the results of the skipgram models, summarized in Table 2 as Skipgram Negative Sampling (SGNS). We noticed that the model behaves better when the embedding size and context is smaller. This might be because of the domain specificity of medicine so the association between words are often limited to a smaller scope than might be the case with general NLP tasks. The PPMI experiments show the best model results, with a dimension size of 300 showing the top results for both the **May Treat** and **May Prevent** cases. This result is in contrast to the SGNS results. We are not sure if the difference in optimization techniques between both methods might be responsible for this differing behavior.

Overall, we think we were able to get the best outcome for PPMI since we were able to use the whole data set unlike the SGNS method where we could only use a subset of the data due to computational limits. At just log count frequency of the data for SGNS, we were able to get results within decimal points difference from the best PPMI result, which suggests that we can expect to get even better results if the whole dataset is used with the

Medical Concepts	Similarity Scores
(tobacco, smoker)	0.481
(tobacco, burning sensation)	0.361
(tobacco, tooth diseases)	0.297
(tobacco, sugar)	0.054

Table 5: Sample pairwise similarity scores between *tobacco* and randomly selected concepts. The medical concepts range from highly related: *smoker*. Concepts that are unrelated to the term *tobacco* logically have lower similarity scores, like *sugar*.

SGNS method.

In conclusion, we learned to use multiple techniques to develop word embeddings for a specific domain, in this case medicine. We also learned how to leverage alternative data summarization techniques (probabilities with PPMI) and dimension reduction techniques (SVD) to develop word embeddings in a computationally-efficient way. Moving forward, it will be interesting to look to other Big Data methods, perhaps parallel or batching operations, that could be used to train a dataset of this volume using the SGNS stochastic gradient technique.

5 Git Repository

The source code along with the extracted embeddings can be found in the following repository: https://github.com/obanko01/embeddings.

References

1. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed Representations of Words and Phrases and their Compositionality. In: Advances in Neural Information Processing Systems 26; 2013. p. 31113119.

2. Choi, Youngduck, Chill Yi-I. Chiu, and David Sontag. "Learning low-dimensional representations of medical concepts." AMIA Summits on Translational Science Proceedings 2016 (2016): 41.

3. Asgari, Ehsaneddin, and Mohammad RK Mofrad. "Continuous distributed representation of biological sequences for deep proteomics and genomics." PloS one 10.11 (2015): e0141287.

4. Dubois, Sebastien, and Nathanael Romano. "Learning Effective Embeddings from Medical Notes."

5. Finlayson, S. G., LePendu, P. and Shah, N. H. Dryad http://dx.doi.org/10.5061/dryad.jp917 (2014).

6. Mikolov T, Chen K, Corrado G, Dean J. "Efficient Estimation of Word Representations in Vector Space." https://arxiv.org/pdf/1301.3781.pdf (2013).

7. Keerthi, S. Sathiya, Tobias Schnabel, and Rajiv Khanna. "Towards a better understanding of predict and count models." arXiv preprint arXiv:1511.02024 (2015).

8. O. Levy and Y. Goldberg. Neural word embedding as implicit factorization. NIPS 2014.