Medical Word Embedding

NLP Group Project

K. B. Ravi Kiran Reddy (IMT2017034)
S. Purvaj (IMT2017039)
G. Sai Sriram (IMT2017018)

Abstract

In this paper we will give a picture about training and obtaining medical word embeddings. We will discuss about the different medical data that is available. We will have a comparision the embeddings trained from different methods like *CBOW*, skipgram etc. and try to give some explanations why each of them might perform in such a way.

1 Introduction

1.1 Problem Statement

The current problem at hand is to develop an embedding scheme satisfying the following properties .

- Embeddings for a disease must be close to its corresponding symptoms, medications(drugs), part of the human anatomy etc.
- Given an embedding we must be able to categorize it into few categories human anatomy, disease, symptom , etc. However we have fixed the classes here into disease,symptom and cure.

2 Dataset / Corpus

2.1 Different Medical Corpora

With the advancement in the medical field it has become less difficult to obtain medical records , medical texts for developing machine learning models. However most of them do not seem to be available for free. We list some of the well known medical data in textual form.

2.1.1 PubMed Dataset

This data comprises more than 26 million citations for biomedical literature from MEDLINE, life science journals, and online books. This is a publicly available dataset for academic usage. Link to the homepage.

2.1.2 MIMIC Dataset

MIMIC-III (Medical Information Mart for Intensive Care III) is a large, freely-available database comprising deidentified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. This is data though available freely requires one to clear a course to obtain access. Link to the homepage.

2.1.3 EBM NLP

Collection of pubmed abstracts from randomized control trials (RCTs). Annotation of Population, Intervention, and Outcomes (PICO elements) are available. Link to the Dataset.
In this paper we have used the **PubMed** dataset as it was more close to the problem statement. The Drive link to the dataset. This data is a result for RCT. In clinical research, randomized controlled trials (RCTs) are the best way to study the safety and efficacy of new treatments. RCTs are used to answer patient-related question.

2.1.4 Scraping Data

Some of the major sources of medical data are online medical websites, articles etc. Data from these can be scraped using proper web scarping scripts and then used for data analysis.

2.2 Data Preprocessing

One of the main task for NLP is to pre process data that is obtained from the various sources. In current scenario we have 2 sets of PUb Med data 20K, 200K based on the documents covered.

- Lowercasing the words.
- Removing stop words. After some experiments it was found that removing the stop words from he data hindered the training of the word embeddings.
- lemmetization and Stemming were a possible task to be done but there might be change in context A treats B $\longrightarrow$ B is treated by A. Therefore we have not gone with lemmetization.
- Removing the numbers of present any.

3 Word embeddings

Word embeddings are a type of word representation that allows words with similar meaning to have a similar representation.

3.1 Medical word Embeddings

Medical word embeddings will have huge impact on the many of the medical tasks that can be automated. Some models can help doctors in improving their efficiency. Recently with the advancement in chatbots there is a pressing need to develop Language models and embeddings to assist the patients without waiting for doctors. This also plays a major role in tasks such as NER.

3.1.1 Desired Properties

- f($disease_1$,cure) = cure for $disease_1$
- g($disease_1$,bodypart) = body part that this disease effects
- h(w_1) = whether it is disease, cure, medicine etc.

The above are some of the properties where f,g,h are function ranging from simple linear functions to complex ML models. Ideally we must get the embeddings in such a way that

$$w_1 \text{ - } disease + cure = u_1$$

where u_1 must be the cure for disease w_1. *cure*, *disease* are the embeddigns of the respective words. Another example can be,

$$opthamology \text{ - } eye + brain = neurology$$

4 Obtaining Word Embeddings

There are many methods to obtain word embeddings ranging from $CBOW$ to the latest sub word embeddings. We will discuss some of the possible methods and the possible hyper-parameters that can be tweaked to get better results. Out of all the methods we have implemented some of them and regarding others, we discuss how useful are they in terms of the said task.

4.1 Word2Vec - *CBOW*

4.1.1 Theory

In this method, the embedding for each word is learned by considering the context of the word, which is a fixed window of words to the left and right of the considered word.

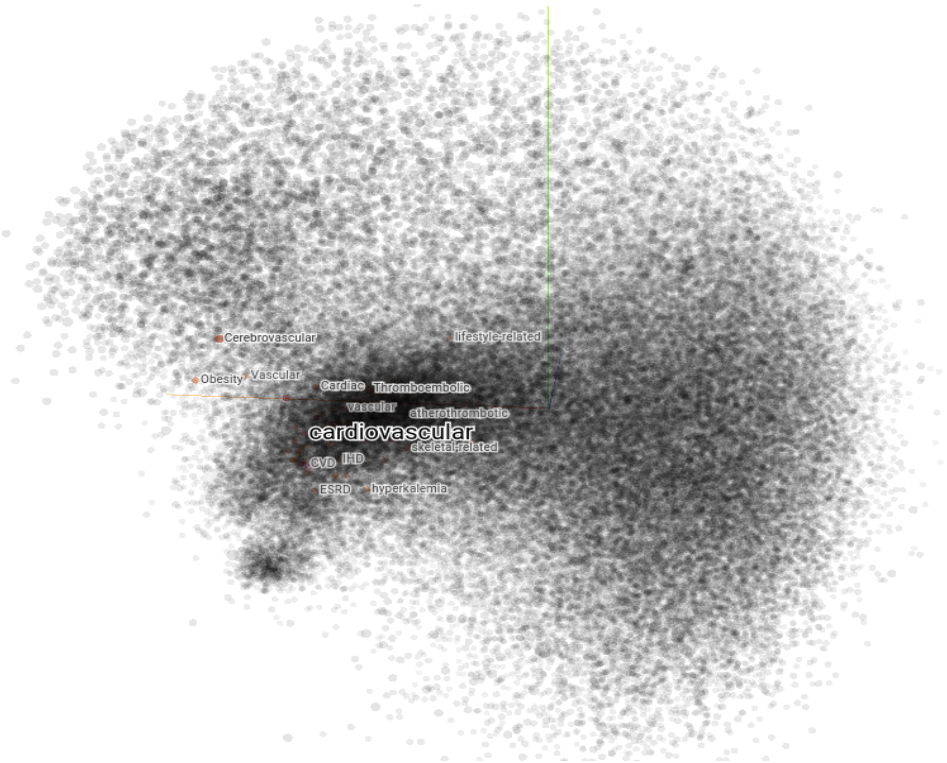
4.1.2 Experiments

For our experiments, we have used the Word2Vec model from the Gensim library which has the tools for training and some inference techniques for both CBOW and Skipgram models. We trained the CBOW model using the PubMed dataset and here are the observations we made corresponding to the model.

We tried with window sizes of 3, 5 and 7 and a context of size 5 seems to work best. We tried to obtain some words embeddings which close to a given word. Here is an example, the word embeddings closest to the word "cardiovascular" were 'CVD', 'atherothrombotic', 'cerebrovascular', 'cardiometabolic', 'macrovascular', 'CHD', 'vascular', 'cardiac', 'thrombotic' and 'thromboembolic. Similarly, words with embeddings closest to "brain" were neuronal, cerebral, hippocampal, neural, cortical, brainstem, limbic, axonal, cerebellar, myocardial, myocardium. We used tensorflow embeding projector to have a visual at the 100 dimensinoal embeddings in 3D.

```
[('CVD', 0.8057840466499329),
 ('atherothrombotic', 0.7619547247886658),
 ('cerebrovascular', 0.7502428293228149),
 ('macrovascular', 0.7423928380012512),
 ('cardiometabolic', 0.7204415202140808),
 ('CHD', 0.7027085423469543),
 ('vascular', 0.6984412670135498),
 ('cardiac', 0.6910381317138672),
 ('thrombotic', 0.6840763092041016),
 ('thromboembolic', 0.6602818965911865),
 ('Cardiovascular', 0.6329425573348999),
 ('cardio-metabolic', 0.6171876192092896),
 ('coronary', 0.6088035106658936),
 ('thrombo-embolic', 0.5994089841842651),
 ('arrhythmic', 0.5947771668434143)]
```

```
[('cutaneous', 0.7127124071121216),
 ('Skin', 0.7032967209815979),
 ('facial', 0.6588599681854248),
 ('dermal', 0.6499264240264893),
 ('thermal', 0.635189414024353),
 ('eyelid', 0.6264931559562683),
 ('dentinal', 0.6177009344100952),
 ('erythema', 0.6150423288345337),
 ('wound', 0.6118344664573669),
 ('dentine', 0.60770583152771),
 ('pigmentation', 0.6013128757476807),
 ('conjunctival', 0.6009274125099182),
 ('tongue', 0.6000021696090698),
 ('conjunctiva', 0.5998902320861816),
 ('tooth', 0.5936832427978516)]
```



4.2 Word2Vec - *Skipgram*

4.2.1 Theory

In Skipgram, the embedding is learned in the reverse manner as that of CBOW. The target word is used to predict the probabilities of the words that could appear in the context of the word in consideration in order to learn the embeddings.

4.2.2 Experiments

Similar to CBOW, we tried with context windows of 3, 5 and 7 in the Skipgram model and window size 5 was the best among them. The words closest to the term "cardiovascular" with this model were 'CVD', 'CHD', 'macrovascular', 'cerebrovascular', 'lifestyle-related', 'CVDs', 'nonatherosclerotic','cardio-metabolic', 'MVEs', 'CVRFs'.
Belowo are the words closest to cardiovascular and skin obtained by this embedding.

```
[('CVD', 0.8593591451644897),
 ('CHD', 0.7907732129096985),
 ('macrovascular', 0.780956506729126),
 ('cerebrovascular', 0.779969334602356),
 ('cardio-metabolic', 0.7702720165252686),
 ('CVEs', 0.7678258419036865),
 ('noncardiovascular', 0.7629328966140747),
 ('CVDs', 0.7549433708190918),
 ('atherothrombotic', 0.7514378428459167),
 ('nonatherosclerotic', 0.7478843331336975),
 ('cardiometabolic', 0.7432365417480469),
 ('CVRFs', 0.7429026365280151),
 ('non-cardiovascular', 0.7402695417404175),
 ('CVD-related', 0.7304717302322388),
 ('lifestyle-related', 0.7303916811943054)]
```

```
[('cutaneous', 0.8287128210067749),
 ('Skin', 0.7847676277160645),
 ('sunburn', 0.7182192802429199),
 ('dermal', 0.7150313258171082),
 ('eczematous', 0.6894139051437378),
 ('pruritic', 0.6872032284736633),
 ('fingertip', 0.684495747089386),
 ('photoageing', 0.6837857961654663),
 ('immediate-type', 0.6751985549926758),
 ('hyperkeratotic', 0.6669816970825195),
 ('sun-damaged', 0.6669158935546875),
 ('non-melanoma', 0.6645284295082092),
 ('Mantoux', 0.6639062762260437),
 ('periwound', 0.663240909576416),
 ('corneum', 0.6620181798934937)]
```

To summarize, in both the variants of Word2Vec, we see that the words corresponding closest embeddings are more or less related to the word although some other common english terms seem to creep in sometimes.

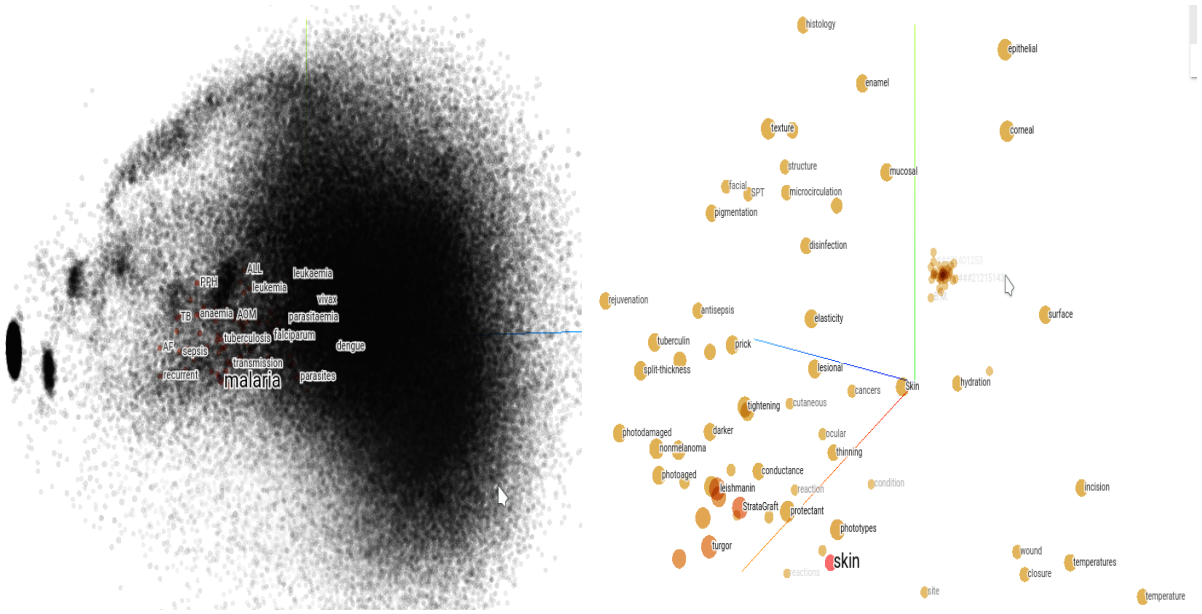
4.3 GloVe

4.3.1 Theory

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

4.3.2 Experiments

We trained the Glove model using the PubMed dataset using the glove library in python. Below are a couple of examples of how the word embeddings turned out in 100 dimensions. We think the glove embeddings are not as good the Word2Vev embeddings.



4.4 Fast Text(Sub word models)

Fastext is an extension of the Word2vec model. It was developed by Facebook AI and research (FAIR).

4.4.1 Theory and Maths

Instead of considering whole word as an embeddign fast text considers the parts of words as n grams. For example

$$\text{apple , n} = 2 \longrightarrow ¡ap,\ pp,\ pl\ ,\ le¿$$

This helps the model to understand the prefixes and suffixes of the words. the embedding for a word will be an average / sum of all its n grams .Once the word has been represented using character n-grams, a skip-gram model is trained to learn the embeddings. This model is considered to be a bag of words model with a sliding window over a word because no internal structure of the word is taken into account. As long as the characters are within this window, the order of the n-grams doesn't matter.
This also gives an advantage over the traditional models by trying to get embeddings for the out of vocabulary words by combining embeddings for the ngrams of the new word. Theoretically we can observe that this is helpful in the field of medicine.
For example,

cardiology -: cardio + ology
opthamologist -: opthamology + ologist

This model has an additional hyper parameter besides the word2vec model i.e. n grams.

4.4.2 Experiments

Here are some of the results that we got based on each of the words. @ is the token that replaces all the numbers.

```
model.get_nearest_neighbors('nose',20)
```

```
[(0.8252351880073547, 'nose/nasal'),
 (0.806947648525238, 'runny'),
 (0.7981841564178467, 'sneezing'),
 (0.7970660924911499, 'stuffiness'),
 (0.7968831658363342, 'nose/throat'),
 (0.7911538481712341, 'stuffy'),
 (0.782988429069519, 'rhinorrhea'),
 (0.7792298197746277, 'nasal'),
 (0.7725257277488708, 'rhinorrea'),
 (0.7685657143592834, 'bunny'),
 (0.7679197788238525, 'nonnasal'),
 (0.7637228965759277, 'nostrils'),
 (0.7570949196815491, 'noses'),
 (0.7515284419059753, 'postnasal'),
 (0.7488957643508911, 'blows'),
 (0.7484537959098816, 'rhinorrhoea'),
 (0.7475621700286865, 'sneeze'),
 (0.7384189963340759, 'nostril'),
 (0.7373910546302795, 'itchy'),
 (0.731083333492279, 'sneezes')]
```

```
model.get_nearest_neighbors('heart',20)
```

```
[(0.7720064520835876, 'heart-failure'),
 (0.7541844844818115, 'congestive'),
 (0.739101231098175, 'Heart'),
 (0.7043959498405457, 'arterial-cardiac'),
 (0.6951054930686951, 'heart/lung'),
 (0.6903347373008728, 'asystolic'),
 (0.6862385869026184, 'systolic'),
 (0.6793137788772583, 'CHF'),
 (0.6762699484825134, '@bpm'),
 (0.6746391654014587, 'cardiac'),
 (0.6738244891166687, 'non-heart'),
 (0.6718320250511169, 'beating-heart'),
 (0.6714013814926147, 'rate-systolic'),
 (0.6692805290222168, 'SIHD'),
 (0.6637172102928162, 'failure'),
 (0.6624863147735596, 'left-ventricular'),
 (0.6611158847808838, 'cardio-respiratory'),
 (0.6608555912971497, 'rate-pressure'),
 (0.653653085231781, 'HF'),
 (0.653373122215271, 'cardiocirculatory')]
```

```
model.get_nearest_neighbors('throat',20)
```

```
(0.9268910884857178, 'sore'),
(0.921187162399292, 'nose/throat'),
(0.9205098748207092, 'throats'),
(0.8450037240982056, 'hoarseness'),
(0.8071814775466919, 'Hoarseness'),
(0.7631379961967468, 'pharyngolaryngeal'),
(0.7446455359458923, 'Throat'),
(0.7406995296478271, 'pharyngotonsillitis'),
(0.7251002788543701, 'laryngo-pharyngeal'),
(0.7204555869102478, 'stuffiness'),
(0.7202932834625244, 'expectorating'),
(0.715045690536499, 'cough'),
(0.7102416753768921, 'nose'),
(0.710084080696106, 'tonsillopharyngitis'),
(0.7055957317352295, 'earache'),
(0.700631320476532, 'ear-nose-throat'),
(0.7000788450241089, 'rhinorrhea'),
(0.6964247226715088, 'expectoration'),
(0.6922354102134705, 'sneezing'),
(0.6902430653572083, 'laryngopharyngeal')]
```

We can observe that the neighbours for each of the words are related to them.

4.5 Enhancing Embeddings using MeSH

Recently, some studies have suggested that integrating domain knowledge with the text corpora can be beneficial to improve the quality of word embeddings(more emphasis on sub-word embeddings). We construct a MeSH(medical subject headings) term graph based on the MeSH RDF data, followed by a random sampling strategy to generate a number of MeSH term sequences

Sampling can be done using BFS(breadth first search) or DFS(Depth first search). The pre-existing embeddings are trained on these sequences using transfer learning. Sampling can be done using BFS(breadth first search) or DFS(Depth first search). The pre-existing embeddings are trained on these sequences using transfer learning.

4.6 NNLM

Neural-Net Language Models (NNLM) is a very early idea based on a neural probabilistic language model proposed by Bengio et al. in their paper, 'A Neural Probabilistic Language Model' in 2003. It discusses about learning a distributed representation for words which allows each training sentence to inform the model about an exponential number of semantically neighboring sentences. The model learns simultaneously a distributed representation for each word along with the probability function for word sequences, expressed in terms of these representations. Generalization is obtained because a sequence of words that has never been seen before gets high probability if it is made of words that are similar (in the sense of having a nearby representation) to words forming an already seen sentence.

5 Categorizing Words using Embeddings

There are several ways to tackle this specific problem. Some of them being :

5.1 Developing an ML classifier

Mathematically ,

$$\text{Model(embedding)} = \text{class}$$

The class is a probability with which it belongs to other classes. However some drawbacks of this method are that we will need a dataset with labelled classes and also the number of classes if fixed. If we want to add newer classes then we need to retrain the model. Also the model will be specific to this embedding scheme itself. With this method we can also achieve the Named entity recognition problem as well.

5.2 Classifying based on Cosine Similarity

Here, we consider the cosine similarity of a word with a set of predefined words categorized into three classes (disease, symptom and drug) for current analysis. Cosine similarity of a word embedding is measured with each of the word embeddings in a particular class and this is done with all the three classes. The average of the cosine similarities within each class is considered and the word is put into the class with which it achieves the highest average cosine similarity.
Note that this method only helps in comparing embeddings obtained from different models for the task but is not an absolute measure of the property of how well the embeddings are indicative of their corresponding class.

5.3 Generating Data using Pretrained NER

NER [6] seeks to locate and classify named entities mentioned in unstructured text into predefined categories We could use a Named Entity Recognition on the model to identify the words belonging to specific class, thus generating a supervised data. We can use different

NER models for each of the classes to determine the labels of words for each class and then use some classification model.

5.4 Classifying based on sentence embedding

Here, in this approach we are tyring to use unsupervised learning approach to index the data as disease,symptom or a cure. We cannot use any unsupervised learning approaches to the word embedding as the disease contains its cure and symptoms in its vicinity. So we tried to generate some sentences in the following way:
For every word 'w' in corpus:

$$\text{"w + is a disease"} \longrightarrow \text{add to DiseaseList}$$
$$\text{"w is a cure"} \longrightarrow \text{add to CureList}$$
$$\text{"w is a symptom"} \longrightarrow \text{add to SymptomList}$$

Now we generate sentence embeddings to sentences in the DiseaseList and apply k-means clustering on these sentence embedding with number of clusters set to 2. So now we from the test set(words labelled as diseases), we look these word's sentences(w + " is a disease") in these clusters and assign the probability to the words in those clusters. For example if x+y are the number of words in the testset and x number of words' are located in $cluster_1$ and y in $cluster_2$ then the words in $cluster_1$ have a probabilty of x/(x+y) to be a disease and words in $cluster_2$ one have a probabilty of y/(x+y) to be a disease. In this way words are assigned the probability of being a cure or a symptom using the CureList and the DiseaseList. Based on the maximum probability value the label a particular word is assigned.

6 Measuring the closeness of a medical term embedding with other related terms

This is a method we are proposing for the first property mentioned in the introduction. It is also based on the cosine similarities of a word to other words. We consider a fixed set of diseases, their corresponding symptoms and drugs. Let us refer to a trio of (disease, symptom, drug) as a row. For each row, the pair wise cosine similarities are calculated and summed up. If this sum is greater than any other sum of cosine similarities for the same disease but different symptom and/or drug or vice versa, then the embeddings can be said to have this property to at least a considerable extent.

7 Results and Inferences

Here are the results for the experiments we did with the different embedding models and how they turned out on our metrics for the mentioned properties.

Embeddings	Classification method	Property 1	Property 2
CBOW	Cosine similarity	0.5747	—-
SkipGram	Cosine similarity	0.5810	0.3555
Glove	Cosine similarity	0.5049	—
Fastext	Cosine similarity	0.6274	0.4647
CBOW	NER	0.5747	0.6361
SkipGram	NER	0.5810	0.6732
Glove	NER	0.5049	0.6777
Fastext	NER	0.6274	0.7132

8 Metrics

There is a huge doubt about measuring the quality of embeddings developed as one cannot just check the whole list of corpus. Here are some metric that can be used. However some of them are tedious to implement so we weren't able to do them in practice.

Cosine Similarity Between two words :
Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. It is defined to equal the cosine of the angle between them, which is also the same as

$$cos(\theta) = A \ . \ B \ / \ ||A|| \ . \ ||B||$$

Coherence metric :
This metric which was measures whether or not a group of words adjacent in the embedding space are mutually related. Commonly, this relatedness task has been evaluated in a subjective manner (i.e. using a human judge).
Semantic Relatedness metric
This is for knowledge graphs by exchanging words for concepts. Typically, this metric represents the relatedness score of two given words. In the context of a knowledge graph, we give a pair of concepts to human judges (usually domain experts) to rate the relatedness score on a predefined scale, then, the correlation of the cosine similarity of the embeddings for concepts is measured with human judge scores using Spearman or Pearson.

9 Further Improvements

Some of the improvements that we could have done are as below

- In case we acquire some labelled data or we come up with some unsupervised classification models, then we can develop a classification model that takes the embedding as an input and gives the classification score.
- Another extension of the above solution is to somehow incorporate the loss function of the label of the embedding into the loss function of the word embedding model if feasible.
- Using random walks to sample sequences from the MeSH graph
- Current Fasttext model uses word2vec algorithm, but one could always extend it some other algorithms

References

[1] A Comparison of Word Embeddings for the Biomedical Natural Language Processing
`https://arxiv.org/abs/1802.00400`

[2] Evaluating Word Embedding Models: Methods and Experimental Results
`https://arxiv.org/pdf/1901.09785.pdf`

[3] How to Train Good Word Embeddings for Biomedical NLP
`https://www.aclweb.org/anthology/W16-2922.pdf`

[4] Adapting Pre-trained Word Embeddings For Use In Medical Coding
`https://www.cse.iitb.ac.in/ pb/papers/bionlp-acl17-medical-coding.pdf`

[5] Fast Text
`fasttext.cc`

[6] NER Paper `A Survey on Deep Learning for Named Entity Recognition`