SAPIENZA
UNIVERSITÀ DI ROMA

Learning Sense Embeddings

1871871
Prof. Roberto Navigli
Natural Language Processing Homework 2

June 5, 2019

Contents

1.0 Introduction

The task is to implement Sense Embeddings for Word and Relational Similarity [4]. I implemented a CBOW model using word2vec whose structure can be seen in the Fig 1 below:

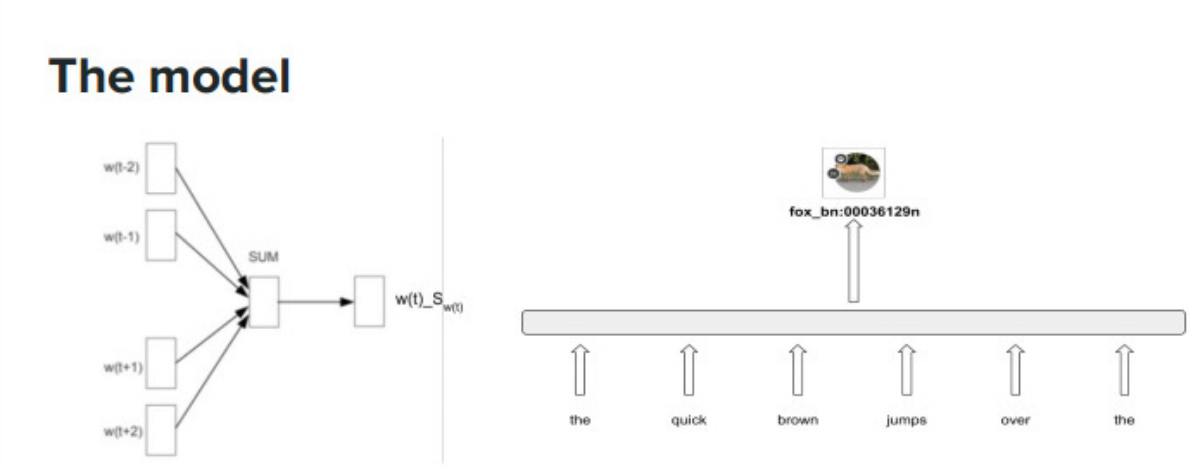

Fig. 1

That is; given a word and its context, predict the sense embeddings for that word.

2.0 Experimental setup

2.1 Dataset

I initially started with Eurosense high coverage corpus [1] which consists of a large XML file. It is a multilingual sense-annotated resource taken from the European Union Parliament. Afterwards, I merged the Eurosense high coverage corpus with a subset of Semantically Enriched Wikipedia (SEW) corpus [2] which is a sense-annotated corpus automatically built from Wikipedia with more than 200 million annotations of over 4 million different concepts and named entities in order to improve the performance.

2.2 Preprocessing

All the processing steps used in this project were performed in Python 3, with the help of key external libraries like gensim, glob, itertools, google translate and lxml. Only the English sentences in the corpus were parsed during this phase and google translate api was used for translating non-english sentences that were miscategoried as english from their current language to english so as to enrich the corpus more.

The parsing consisted of iteratively reading each corpus file and processing it line by line. Using special tags, sentences were extracted along with their annotations. For the purpose of this assignment,data extraction was performed only for the English language sentences. Each sentence, consisted of several texts for the English Language. However, not every texts had corresponding anchors in their annotations and these texts were discarded. Each text annotation includes its disambiguated *BabelNet id*, *Anchor* and *Lemma*. For the high coverage corpus [1], after preprocessing, the anchors in the sentences were replaced with their specific *lemma_ babelnetID*. The total number of sentences that was extracted from the dataset for the first round training (Phase A dataset) was over 1.9M sentences.

After merging and preprocessing a subset of SEW corpus [2] with high coverage corpus [1], the anchors in the sentences were also replaced with their specific *lemma_ babelnetID*. The total number of sentences that was extracted from the dataset for the second round training (Phase B dataset) was over 5.9M sentences.

Phase A and B datasets were tokenized by space and during this tokenization phase, all punctuation marks were removed except `:` and `_` (because the replaced annotation is of the format *Love_ bn:00049573n*). Additionally, each token was converted to lowercase in order to avoid duplicates senses i.e. *"love_ bn:00049573n"* and *"Love_ bn:00049573n"*.

2.3 Training

The training was performed using the Word2Vec model from gensim. At first, the model's vocabulary was built using the corpus that was preprocessed above and afterwards, the vocabulary was used to train the model.

2.3.1 Model Parameters

Across the literature on word embeddings with Word2vec, the default gensim parameters frequently yielded good results, so I fixed some of those parameters while I varied the model structure (*CBOW* and *SKIPGRAM*). I also varied the *embedding size* and *number epochs* for training. This made a huge difference in the quality of the model. The table below contains the fixed parameters:

alpha	0.025	sample	0.001
window	5	negative	5
min_count	5	Hierarchical softmax	True

2.3.2 Model Evaluation: Similarity measure

The best way to evaluate the model's performance is to perform a similarity measurement task. In this work, WordSimilarity-353 dataset [5] was used for the model's evaluation. For each pair of words in this dataset, cosine similarity was used to calculate the similarity between the pair of words in the model's embedding file and this is scored. Afterwards, Spearman correlation was used to compare the cosine similarity score of the model against a gold similarity score (annotated by human). I did not incorporate any other similarity measures.

2.3.3 Hyperparameters Tuning

The following hyperparameters were explored in order to maximize the correlation score. The best hyperparameters were chosen based on the Spearman's correlation performance.

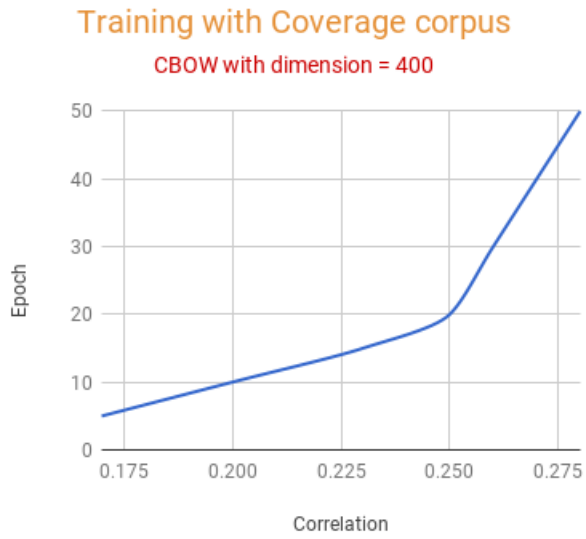

Fig. 2

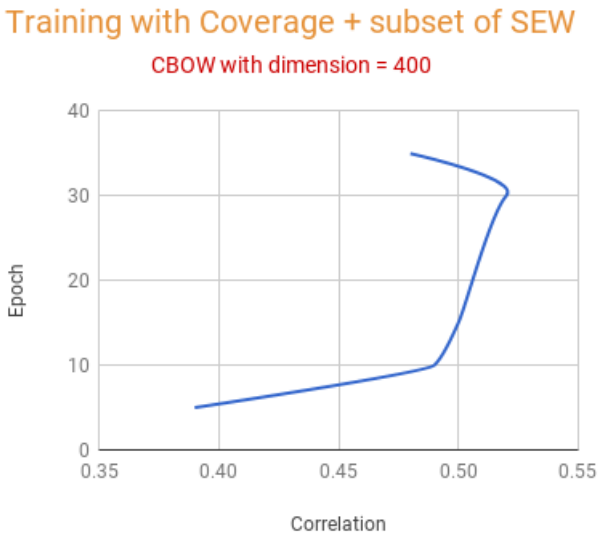

Fig. 3

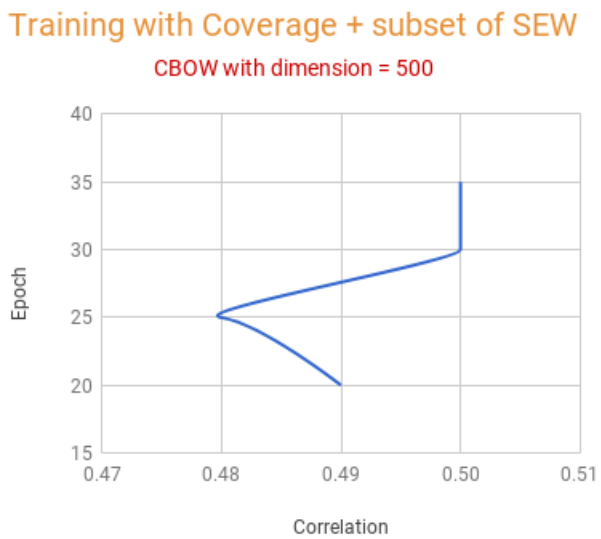

Fig. 4

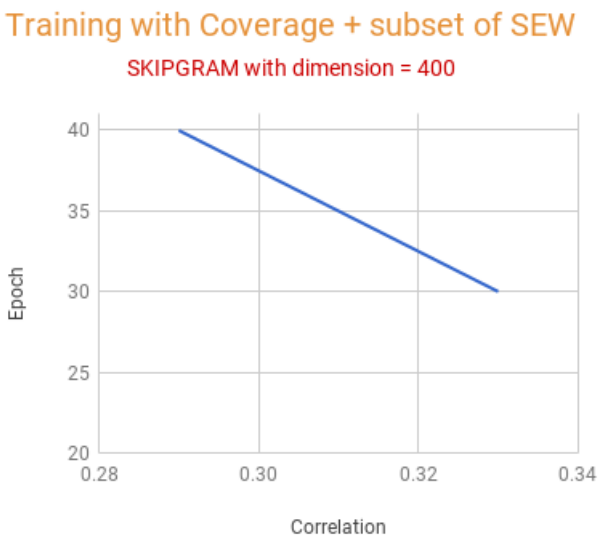

Fig. 5

From Fig. 2, it is evident that the best correlation is 0.28 at the 50th epoch after training with only the coverage corpus when the vector dimension is 400 using CBOW and all other parameters fixed. So I figured that feeding more data to the model could improve the correlation.

After adding more data to the model and training on the merged version of coverage and a subset of sew corpus with the same vector dimension (400) using CBOW, the correlation rapidly increased to 0.52 at the 30th epoch but the correlation started reducing afterwards. This can also be seen in Fig. 3. I also decided to increase the vector dimension to 500 since the model's vocabulary is now larger after merging the corpus and this could yield a better result. However,the model's correlation did not improve. It got stucked at 0.50 at the 30th and 35th epochs. This is evident in Fig. 4.

Fig. 3 yields the best result of the model so far.

Finally, using the same dataset that was used in Fig. 3, the model was trained using SKIPGRAM to see if there would be any improvements. However, there was no improvement in Fig. 5.

The combination of parameters that yielded the best result for this model is Fig. 3 (Training with the merged dataset using vector dimension of 400,CBOW,epochs = 30 and all other parameters being fixed).

3.0 Embeddings Visualization

Visualization of embeddings is very useful to understand how Word2Vec works and how to interpret relations between the vectors that was learned from the corpus during model training. T-SNE [3] is quite useful in this case to visualize similarity between objects which are located into multidimensional space. A T-SNE visualization of the top 50 similar senses to 6 different synsets can be seen in Fig. 6 below. T-SNE maps multi-dimensional data to two or more dimensions, where points which were initially far from each other are also located far away, and close points are also converted to close ones.

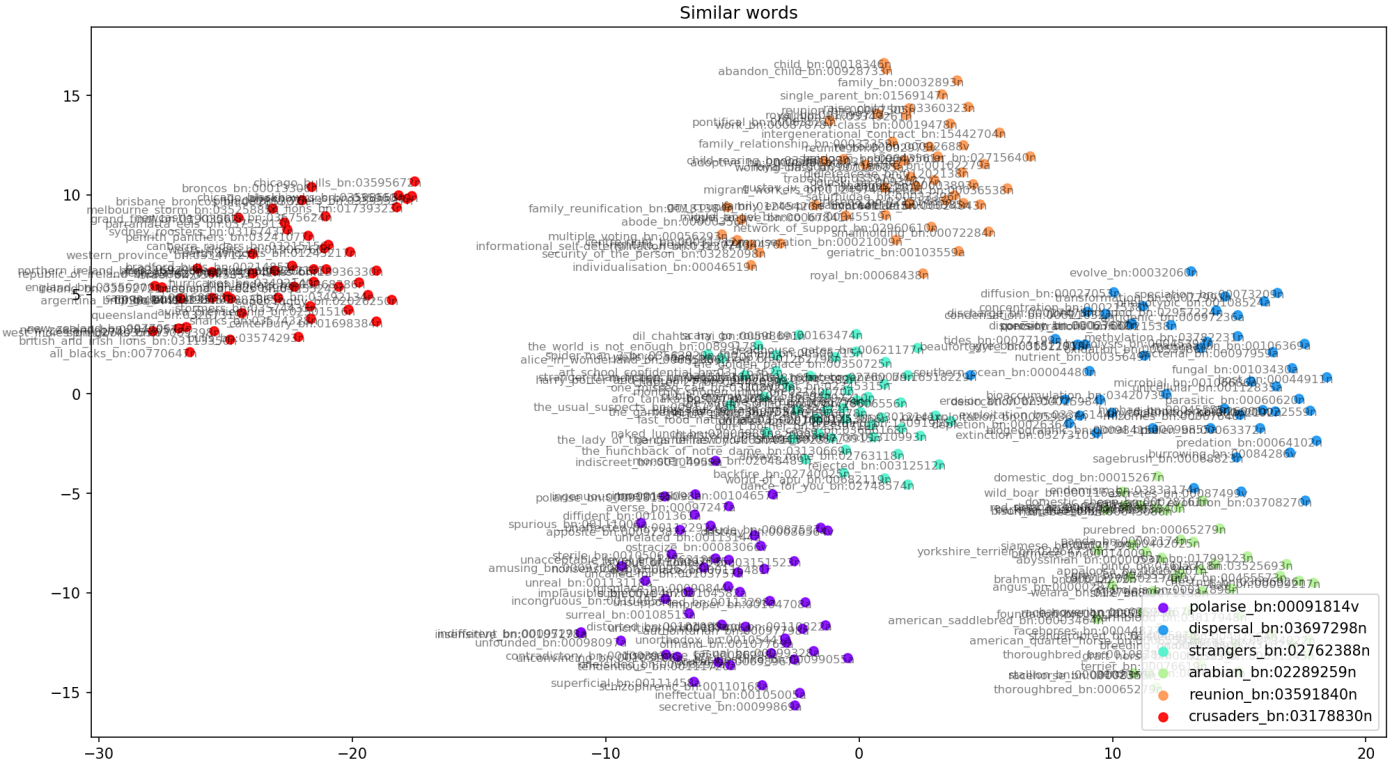

Fig. 6: T-SNE [3] visualization of the top 50 similar senses to 6 different synsets with perplexity = 15 and n iter = 5000.

The visualization in Fig. 6 can be even more informative if we map the sense embeddings in 3D space. Fig. 7 portrays a 3D visualization of 5000 senses in the vocabulary of the model. This plot is used to depict how diverse the domain of the corpus that was used for training the model is. I couldn't plot for all the senses in the vocabulary because of computation time.

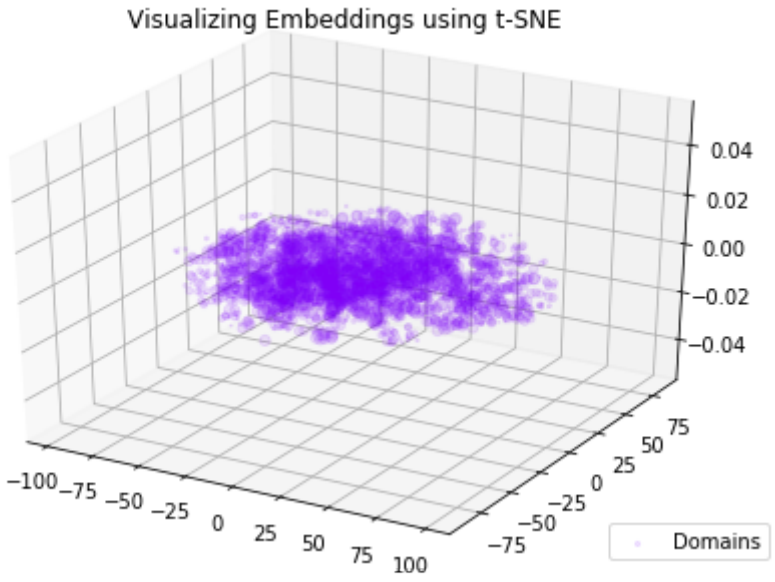

Fig. 7: 3D visualization of 5000 senses in the vocabulary using T-SNE [3]

Final results of the model is shown in the table below with the best parameters in bold with the merged corpus:

Size	**400 (CBOW)**	500 (CBOW)	400 (SKIPGRAM)	400 (CBOW)
Epochs	**30**	30	30	50
Correlation	**0.52**	0.50	0.33	0.27

4.0 Conclusion

The best performance of the model in this work on the word similarity task yielded a correlation score of 0.52 while using the CBOW structure.

This correlation score was due to the OOV words in the similarity dataset. Also, since we saw a rapid increase in the correlation score while training the model with the merged corpus unlike when the model was trained with the coverage corpus alone, then it is safe to say that adding more data from different domains can further improve the model's performance on the word similarity task. CBOW is best for this task because the corpus that was used has higher number of samples. SKIPGRAM does not work well for this task because the corpus has higher number of samples.

Furthermore, the whole point of performing word similarity task it to help with word sense disambiguation,most importantly in the multilingual space. We can use the sense embeddings that was generated from this model to also check for similarities between word pairs in other languages with the help of the *babelnetID* that is present in the sense embeddings.

References

[1] Claudio Delli Bovi, Jose Camacho Collados, Alessandro Raganato and Roberto Navigli. EuroSense: Automatic Harvesting of Multilingual Sense Annotations from Parallel Text. Proceedings of 55th annual meeting of the Association for Computational Linguist(ACL 2017), pages 594{ 600, Vancouver, Canada, 30 July-4 August 2017. http://lcl.uniroma1.it/eurosense/

[2] Alessandro Raganato, Claudio Delli Bovi and Roberto Navigli. Automatic Construction and Evaluation of a Large Semantically Enriched Wikipedia. Proceedings of 25th International Joint Conference on Artificial Intelligence (IJCAI-16), pages 2894{ 2900, New York City, New York, USA, 9-15 July 2016. http://lcl:uniroma1:it/sew/

[3] https://github:com/sismetanin/word2vec-tsne

[4] Iacobacci, Pilehvar and Navigli. SENSEMBED: Learning Sense Embeddings for Word and Relational Similarity. Sapienza University of Rome,2015.

[5] http://www.cs.technion.ac.il/~ gabr/resources/data/wordsim353/wordsim353.zip

[6] https://gitlab.com/mokeam/mariam_garba_1871871_nlp19hw2