NLP HOME WORK 2 "SENSE EMBEDDINGS" REPORT

MANOOCHEHR JOODI BIGDELLO, 1860273

Data preparing and tensorflow model

I. This Home Work is Sense Embedding's and I used Eurosense dataset in both high Coverage and high Precision format.
II. As my model instead of libraries like genism I try to implement my own Neural Network model based on word2vec, so I Implemented a skip Gram model with tensorflow and for getting sense Embedding's I feed my model just with senses as Skip Gram input values and the context words in neighborhood as my output for train.
III. As dataset for train first, I used High Precision and High Coverage Separately and then Concatenation of Both for having more data to train, for preventing Out Of Memory (OOM) problem in colab I write all train input and desired labels in files and read data from file in size of batch and train on that. Writing on file and loading from file for every pair increase run time by 20%.
IV. I used window size of 5 and 10 for creating my feed data for Skip Gram model, and I created pair of <sense, context word> and then transform it to corresponding id based on my words and senses dictionary. Amount of data in each form reported in Table 1.
V. I used Skip Gram with two form of loss: noise-contrastive estimation loss (NCE) and sampled softmax loss with 64 sample as negative sampling, then for minimizing loss function I used Gradient Descent Optimizer from tensorflow with learning rate of 1.0. At the end I also normalized the Embedding vectors. I also used two Embedding size of 100 (for NCE) and 300 (for Softmax) for vectors.

Training phase

I. In training phase I used 20% of high precision dataset (40 M pair) with NCE loss for tuning hyper parameters and learning rate of 1.0 and negative sapling of 64 was faster and accurate ones. So I pick this hyper parameters as my starting point and I used more data for train.
II. I used 4 different configuration for my model with NCE loss with different amount of data and same hyper parameters. Hyper parameters, amount of data pairs and their results shown in Tables 1, 2, 3 respectively.

III. In second phase I changed my loss function from NCE to Softmax Sampling and I just changed Embedding size to 300 (instead of 100) and train this model with concatenation of high precision and high coverage for further experiments.

IV. I plot my results in figures 2 for different losses (NCE and Softmax Sampling), and also spearman correlation for both losses in figure 3.

Testing phase

I. After each training step I load train data and compute cosine similarity for pair of words appeared in wordsim353 dataset with respect to all senses of two words, then I compute spearman correlation and best results was in NCE 0.165 and for softmax sampling best result was 0. 079.

II. Source of some of errors: around 30 words from wordsim353 didn't have annotations in dataset I used (Eurosense), some bn:Id assigned for different senses for example we have same bn:Id of bn:00030751n for both ',' and 'endeavour'.

III. I used T-SNE plot of first 400 senses, below I put some parts of that for showing that model assign close vectors to same close senses.

Dataset	Window size	Amount of pairs <sense, context word>
Eurosense High precision	5	200 Million
Eurosense High precision	10	422 Million
Eurosense High coverage	5	580 Million
Eurosense High coverage	10	1.1 Billion
Eurosense High precision + High coverage	5	620 Million
Eurosense High precision + High coverage	10	1.5 Billion

Table 1. Amount of data pairs for different datasets with corresponding window size.

Hyper parameters	Window size	Negative sample	Learning rate	Steps	Embedding size	Vocab size	Sense size
NCE	10	64	1.0 and 0.1	15	100	269340	158689
Softmax Sampling	5	64	1.0	15	300	269340	158689

Table 2. All hyper parameter used for trainings

Dataset	Loss mode	Leaning Rate	step	Last step loss	spearman
Eurosense High precision	NCE[1]	0.1	8	1.04	0.01
Eurosense High precision	NCE[2]	1.0	16	1.004	0.06
Eurosense High coverage	NCE[3]	1.0	10	1.488	0.04
Eurosense High precision + High coverage	NCE[4]	1.0	16	1.327	0.165
Eurosense High precision + High coverage	Softmax	1.0	12	0.568	0.079

Table 3. Results based on loss and spearman[1]

Figure 1. Some Learned Sense Embedding's based on T-SNE plot

1- [1] The numbers on NCE[x] is for making difference between them in figure 2

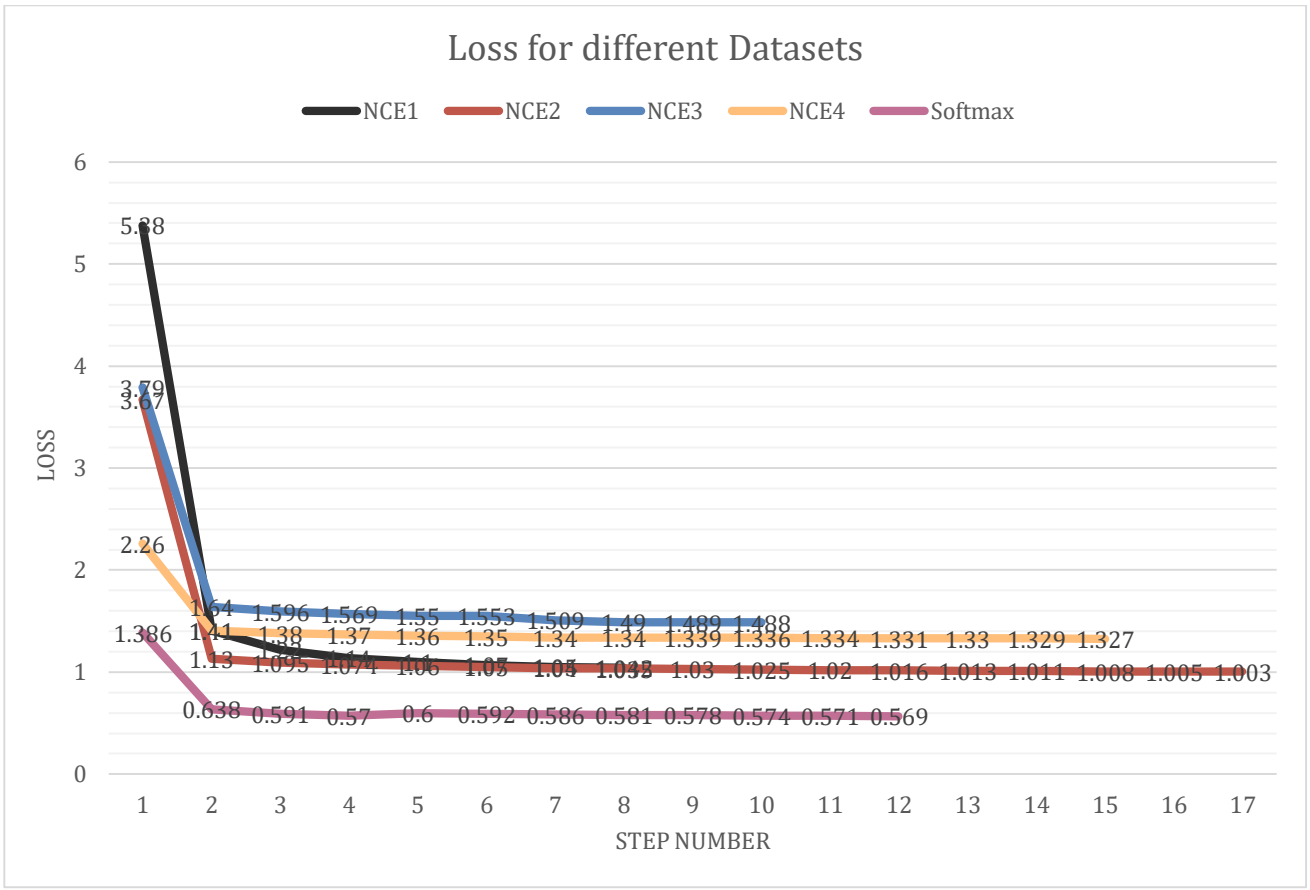

Figure 2. Loss Values for different datasets

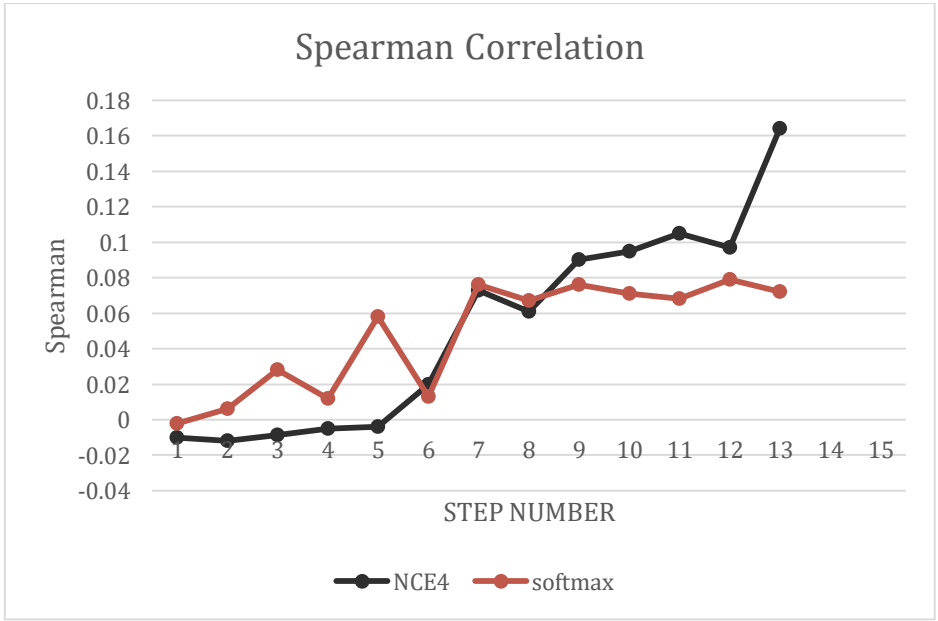

Figure 3. Spearman Correlation.

References

1- wwww.tensorflow.com