A Measure of Sentence Relatedness using Convolution Neural Networks

(Nerissa D'Souza, CCE-NNSP1)

Abstract / Motive:
The Aim of this project is to understand the following questions: How can Convolution Neural Networks be applied for text analysis. How is text processed into an input suitable for Neural Networks. What type of basic network architecture we can start with. What are the network changes for a better result. For this I used sentence pairs and the expected result is a measure of similarity or a relatedness score between the given pair.

Word Embedding or Word Vectors:
GloVe is an unsupervised learning algorithm for obtaining vector representations for words. A word co-occurrence matrix is constructed using a weighted window of size 10 and used for training the model. [i.e word pairs that are d words apart contribute 1/d to the total count]. The resulting word vectors represent information about how every pair of words i and k occur, $w_i^T w_k + b_i + b_k = \log (X_{ik})$. The model then has a weighting function to ensure that rare or very frequent co-occurrences are not under or over weighted. The Euclidean distance or Cosine similarity between two word vectors provides an effective method for measuring the linguistic or semantic similarity of the corresponding words. In this project we use pre-trained vector obtained by training GloVe on the Wikipedia 2014+Gigaword5 dataset with 50, 200 and 300 dimensions. The dataset has 6 billion word tokens. Other algorithms for word vectors are word2vec, Senna.

Input Dataset: I use the SICK dataset (Sentences Involving Compositional Knowledge), which consists of compositional variant sentence pairs with an associated relatedness score between 1-5 which was formed via crowd scouring of 200,000 judgements. The dataset has score distribution in the following ratios [1-2):[2-3):[3-4):[4-5] => 10:14:39:37 .

Preparing the input data: Building each word in the sentence from the word embedding's, yields a matrix x_{nd} , where x_i represent d-dimensional word embedding for the ith word, and x_{ik} represents the kth dimension of the ith word. If a word embedding does not exist in the pre-trained vector vocabulary, use the embedding of an unknown token <unk>, which is initialized to a uniform distribution (-0.05, 0.05). The minimum sentence length is associated with the max convolution window size we experiment with.

Network Architecture:
Convolution Neural Network: Each sentence is then processed through a CNN, taking from the n-gram of NLP, to the convolution window size [1, 2, 3] across all dimensions d. Two types of Convolution were tried:

a) the convolution is applied and the d-dimensions are condensed to 1 in that operation. [Figure 1]
b) the convolution window is applied and the d-dimensions are condensed to r-dimensions. [Figure 2]

The challenge with selection of window size, is the minimum sentence length is associated with the max convolution window size. For the inclusion of window size ∞, convolution filters would differ according to sentence length, to make the model independent of this, a convolution filter cannot be applied, only pooling.

All Convolutions used Shared Weights. Pooling Layers of Max and Mean are used (1xn), which operates per dimension, across the length of the sentence. The weight learnings in the max layer is limited to the max element in the previous layer, while in mean pooling, the error is uniformly distributed among units which feed into it from the previous layers, as are the learnings in weights.

After the Convolution and Pooling layer of each sentence individually, measures of difference between the two sentences: Euclidean and Cosine Distance and the absolute element wise difference are fed to single layer Neural Network of 50 Neurons. [30-100 Neurons were experimented with]

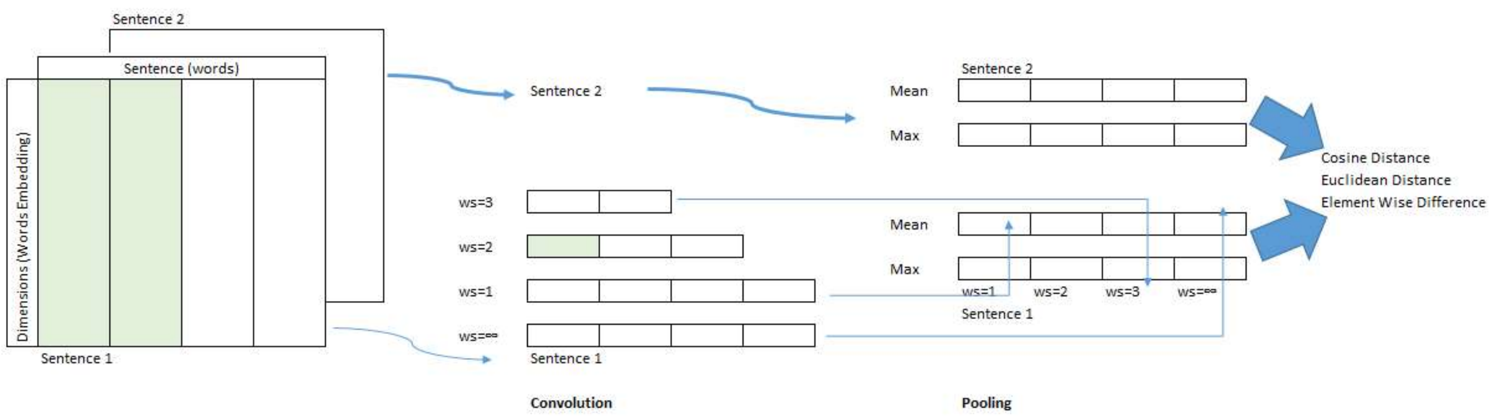

Figure 1: Convolution Architecture (a)

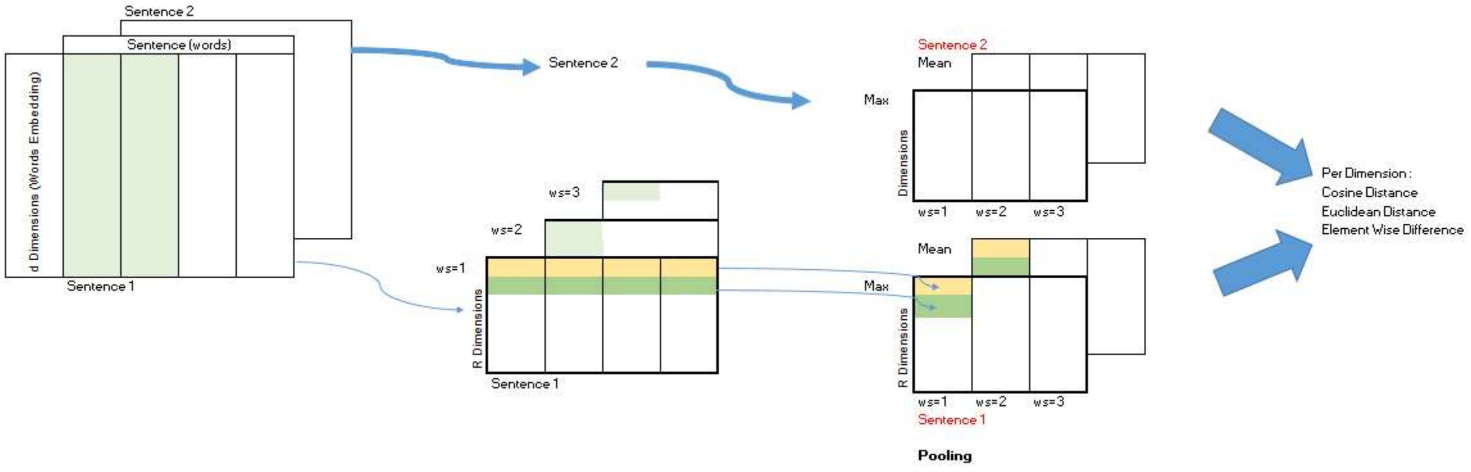

Figure 2: Convolution Architecture (b)

Final Layer, Feed Forward Network: The final layer is a log-softmax layer with 5 output neurons (i.e 5 classes), each output being a probability of belonging to that absolute measure of similarity (i.e 1 or 2 or 3 or 4 or 5). The measure of similarity being given by $\sum_{i=1}^{5} P(i) * i$, where $\sum_{i=1}^{5} P(i) = 1$

Loss Function: Since the output is a probability distribution (softmax layer), a loss function of Kullback Leibler Divergence was chosen. The KL divergence is a non-symmetric measure of the difference between two probability distributions p(x) and q(x). Network learning via Back Propagation, gradient decent. $Loss \quad \frac{1}{N} \sum_i P\_Target(i) \; * \; \ln\left(\frac{P\_Target(i)}{P\_Predicted(i)}\right)$

Regularization: L2 Regularization term, $0.5*\lambda|w|^2$. This regularization has the intuitive interpretation of heavily penalizing peaky weight vectors and preferring diffuse weight vectors. Due to multiplicative interactions between weights and inputs this has the appealing property of encouraging the network to use all of its inputs a little rather that some of its inputs a lot.

Evaluation of the Network:
Dataset vocabulary size: 2312
Number of absent pre-trained vectors: 132
Pearson Correlation is calculated as a measure of linear correlation between predicted and target values to evaluate the network.
Number of Hidden Layer Neurons: 50 neurons was found to be sufficient. Without much change with an increase for all combinations of embedded dimensions.
Choice of Embeded Dimensions: 50 did not yield comparable results, while 300 was highly computational intensive for a minor improvement. 200 embedded dimensions were sufficient.

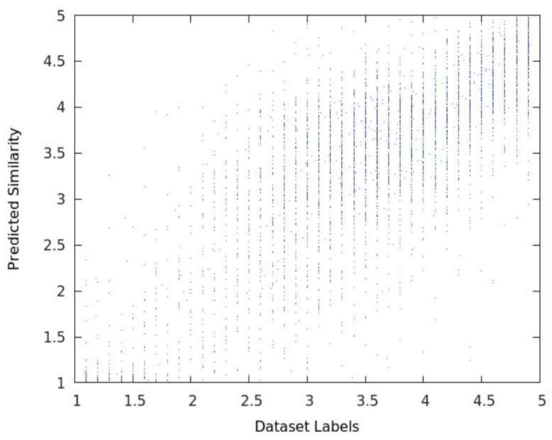

Figure 3: Predicted Similarity

Choice of Window Size: While we see an impact to the loss function for win=1, vs win=1,2,3, there isn't much change in the p-scores. However, this might be the case only for the chosen dataset. In NLP n-gram (n= 1,2,3) are standard choices.

Choice of Compaction Factor by Convolution: In Architecture (a) the accuracy achieved was not comparable to architecture(b). In the case of (b) a factor of 4 i.e 200/4=50 yielded results compared to the un-compacted 200 dimensions through the network, while better than the un-compacted 50 dimensions. In case of using 50 dimensions pre-trained word vectors, there is information lost at the input stage itself. While dimension reduction by the Convolution layer, ensure information is learnt and compacted by the current network.

Choice of Pooling: Max Pooling is the main contributor to network, compared to Mean, for the same number of epochs. The learnings by the network by the Max pooling layer are faster and steady, while the learnings of Mean pooling are more gradual.

Choice of Distance Measures: Cosine distance was experimented with because of the GloVe algorithm, it being a measure of similarity of words/synonyms. Element wise difference was found to be main contributor to the accuracy, while the cosine distance perhaps creating a negative contribution

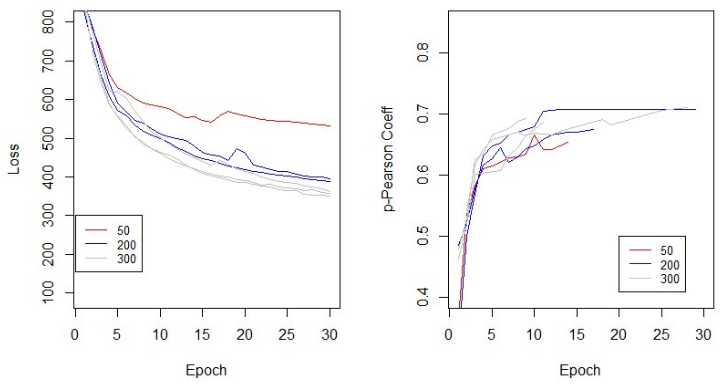

Figure 4: Results from Architecture (a)

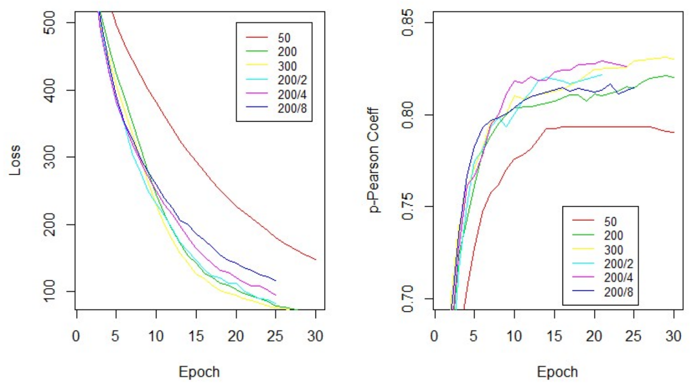

Figure 5: Results from Architecture (b)

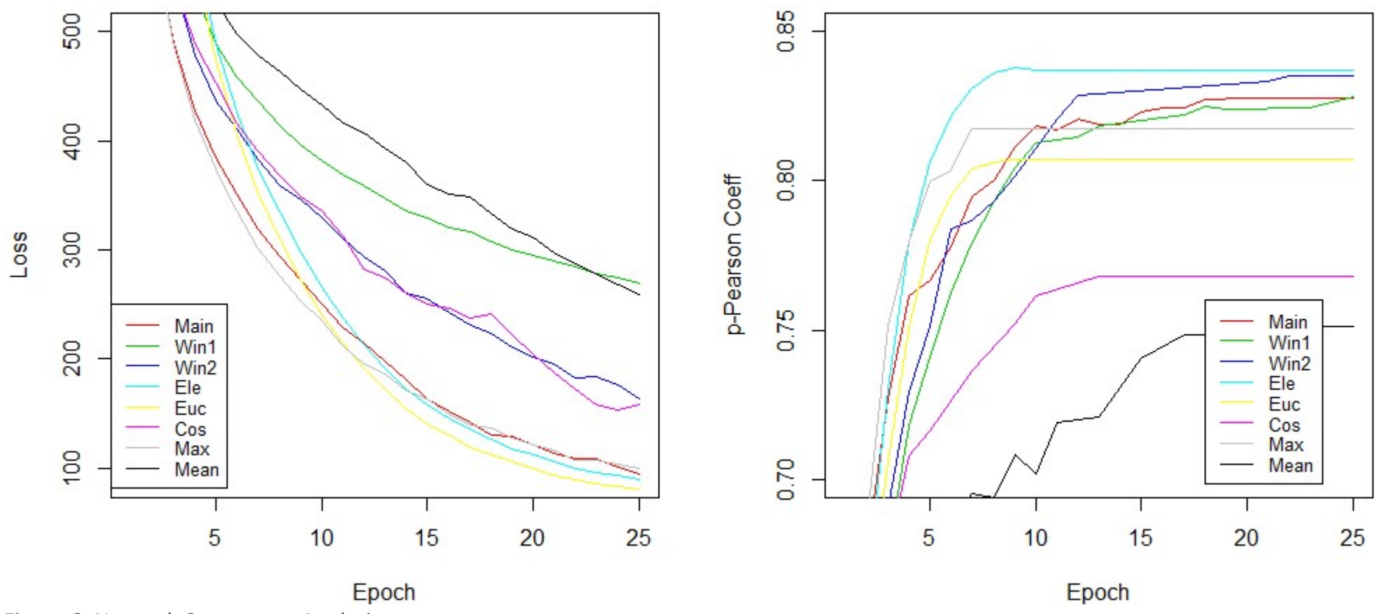

Figure 6: Network Component Analysis

Description	Embed Dimension	Reduced Dimension After Convolution	p (Pearson Coefficient)	Nr CCN o/p	Nr w params
(a)	50	1	0.65303	12	1511
(a)	200	1	0.7077	12	3311
(a)	300	1	0.70929	12	4511

Description	Embed Dimension	Reduced Dimension After Convolution	p (Pearson Coefficient)	Diff (p)	Nr CCN o/p	Nr w params
(b)	50	50	0.79339	-0.03436	600	55605
(b)	200	200	0.82144		2000	581505
(b)	200	100	0.8217		1000	290905
(b)	200	50	0.82775		500	145605
(b)	200	25	0.81469	-0.01306	250	72955
(b)	300	300	0.82913		3000	1232105
(b) win 1	200	50	0.82803	0.00028	300	35405
(b) win 1,2	200	50	0.835	0.00725	400	80505
(b) Element Wise	200	50	0.83686	0.00911	300	135605
(b) Euclidean	200	50	0.80692	-0.02083	100	125605
(b) Cosine Distance	200	50	0.76792	-0.05983	100	125605
(b) Max Pool	200	50	0.81759	-0.01016	250	72955
(b) Mean Pool	200	50	0.75097	-0.07678	250	72955

Next Steps to be Tried:
A different dataset, involving variations in other characteristics associated with NLP.
In the software industry, detecting similarity in defects raised or trouble tickets, historic and present is of value. However, the language used is not necessarily in structural and grammatical English. If the word vector embedding is pre-trained on such data, and used in a CNN network as described in the project, would the network yield similar results.

References:
[1] http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/
[2] http://clic.cimec.unitn.it/composes/sick.html and
http://clic.cimec.unitn.it/marco/publications/marelli-etal-sick-lrec2014.pdf
[3] https://nlp.stanford.edu/projects/glove/
[4] Multi-Perspective Sentence Similarity Modeling
with Convolutional Neural Networks, Hua He, Kevin Gimpel, and Jimmy Lin
http://ttic.uchicago.edu/~kgimpel/papers/he+etal.emnlp15.pdf
[5] Convolutional Neural Networks for Sentence Classification, Yoon Kim, New York University,
https://arxiv.org/abs/1408.5882
[6] Natural Language Processing (Almost) from Scratch,
http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf
[7] http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression/
[8] http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/
[9] http://alt.qcri.org/semeval2014/task1/
[10] http://web.engr.illinois.edu/~hanj/cs412/bk3/KL-divergence.pdf