Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification

Team-10 ---Project -3

Abstract :

Our Project main goal is to learn and implement **Sentiment-Specific word embedding** and test on twitter dataset ,try to do experiments to improve the accuracy.

Implementation:

Implementation is in two phases.First phase is to implement the SSWE.The output of first phase is vectors for each word with Syntax Score and Semantic Score.
In second phase,we train the model by SVM and above vectors ,predict the semantics for test data.

Experiments:

While implementing neural networks for syntax and semantics.
we can do by having both nn same lookup-table and train the model we get the vectors for each words combinely.

But we train the nn's separately for both language and semantics ,we end up with both sets of vectors. For training model in SVM, we create a weighted vector form both.
In weighted vector, the weight for semantic vector is more and we can vary to get the accuracy better.

Accuracy:

For weighted method depending on the weight it varying.
For small portion of dataset from whole tweets the accuracy is about 68% - 75% (considering variable data corpus) .
For weighted method also around same above.

Model	Accuracy
Cbow model (syntactic)	65.43
SSWE_h (sentiment score)	69.38
SSWE_u (syntactic and sentiment scores)	**72.8**

*The accuracy mainly depends upon the sample taken randomly from the corpus. Since, the words we are training in the SVM model may not have word embeddings in the SSWE model and so they are given random embeddings.

*If all the words training in the SVM model have the corresponding word embeddings in the SSWE model , the accuracy will be at **maximum**.

Experiment Model --- Weighted Model

Syntactic weight	Sentiment weight	Accuracy
0.2	0.8	69.73
0.4	0.6	71.96
0.5	0.5	70.25

Experiences :

As the given data corpus is huge we adopted different methods to overcome this problem.

i) We have taken a random sample of 1 million tweets from the given corpus and trained the SSWE model.

ii) While training the SVM classifier some words may not have word embeddings. So, to overcome this problem we have generated a random vector of length 50.

iii) One other way to deal with the above problem is to define a unknown word in the SSWE model and can be used for all the words which don't have word embeddings.

iv) When we are training the neural network the dataset given to the model will work only for small sample.

When we try to deal large samples we we can't give the whole dataset directly to the nn. So, we repeatedly trained the model for each context although it is taking more time.