ORDERING EVENTS CLASSIFIER

Proposal Solution: - in the baseline solution we have used Word2Vec embedding to train our model. There are many techniques that we can use for word embedding like Skip Gram, CBOW, GloVe,, Embedding layer etc., but average word vectors do not consider the sequential order of the tokens. So to consider the sequential order, I thought of using document mean vector instead of word to vector. To make mean document vector I used Glove model because our corpus is very small and it uses local context based learning with global statistics of matrix factorization techniques. To make our classifier more accurate I removed the stop words and digits and punctuations. Apart from this, I thought of learning event1 and event 2 together, not get vector of both explicitly and then concatenate them. It would keep the context of document more precisely. I decided to go with decision tree classifier because it runs faster and we have nonlinear dependency in our data so I thought it would work better. I also applied nonlinear kernel in svm but it did not change the accuracy overall.

Analysis: -

I used SVM, random forest classifier, Decision tree classification to train the model and compared them. The accuracy is around 42% for word embedding based on our corpus using word2vec and SVM and with glove word embedding the accuracy is around 45%.

I also analyzed the accuracy on different classifier. The accuracy of SVM is around 45%, for Random forest classifier is around 36 and for Decision tree is around 35%.

With word embedding		Without word embedding	
Classifier	Accuracy	Classifier	Accuracy
SVM(word2vec)	42	Mul_nb_tf_idf	38
SVM(GloVe)	45	Svc_tf_idf	39
Random forest	36	Bern_nb	37
Decision Tree	35	Bern_nb_tf_idf	39
Logistic Regression	41	Multi_nb	38
		SVC	40