Labelling of sentences in review for presence of a category and a positivity index

Labelling for presence of a category:

For the purpose of labelling the presence of a category we have first manually decided the most relevant categories in the restaurants or food business on which people express their opinions. The categories decided for the purpose are food, service, ambience and value for money

The Algorithm for detection of presence of categories is given in the following stages:

1. Training the knowledge base: For detection of correspondence or similarity between the words to the category we would need a knowledge base to quantify the semantic relation of the words. We use the word2vec model which is based on co-occurrence based relations between words in the sentences to create a vector representation for each word in the space. The dot product between two word vectors can then return the similarity between words. Where this cosine value 0 would indicate no relation and a value 1 would indicate the same word.

We have trained the word2vec model on the all the Yelp reviews given in the data set using the genism python module implementing the word2vec model.

1. Training the tagging engine based on pre-tagged data: The algorithm for training of the dataset after tokenization into sentences is as follows:
   1. The algorithm from here will work in parallel for each category to find the existence of reviews of that category in the sentence.
   2. For each word in the sentence we find the similarity rating with the concept (The method for finding similarity rating is given later). And the values from all the tagged data is then clustered into 8 groups using k-means clustering.
   3. Now we again iterate over all tagged review sentences to find the frequency distribution of the words in the clusters of k-means clustering. This frequency distribution is our representative feature vector to train the Support vector machine decision boundary against the tagged result. After the training is complete the trained clustering object and the SVM object is stored for use for prediction of existence of a category in other reviews.
2. Prediction of category existence and polarity: We will use the trained clustering and SVM objects to predict existence of a category review in the sentence.
   1. The sentence is first broken down into words and then using the trained K-means classifier we classify their score into one of the categories to find the frequency distribution for each sentence. Each topic uses a separate K-means and SVM classifier to given the final output as existence or non-existence of a topic. The probability returned by the SVM in classification is also returned along with the classification result for each topic in a sentence.
   2. The prediction of polarity is facilitated by the textblob python module which predicts the polarity with a positive or negative index between 1 and -1. The final result is converted into strict +1 or -1 by condition across 0.

Similarity rating of a word: The process of finding similarity rating for a word to a category is as follows:

1. Manually select a few words from the corpus whose existence in the sentence gives a very high probability of existence of that topic.
2. Find the maximum similarity between the word and the representative words from each category.
3. The category with the maximum similarity is assigned that similarity rating for use in clustering and rest all categories are given a similarity of 0 to avoid the representation of similarity due to other contexts.