Tvrtko Zadro

# Report

## 1. Labeling Process:

Labeling process used is based on unigrams model. Labeling data consists of two parts: training the model and predicting data based on trained model.

### 1.1. Training model

To train the model we need large number of correctly labeled data. For every example we need word and tag the word was tagged with. The model then records all occurrences of tag for every word. At the end of training phase, for every word in the dictionary we can access total number of times the word was tagged, all tags the word was tagged with and number of times the word was tagged with that tag. That data can be saved in the file as TSV so it can be loaded into model the next time without retraining.

### 1.2. Predicting data

For given list of words, model predicts their tags. For every word in the list, model checks into it's dictionary and looks for all the tags the word *W* has been tagged with in the training set. For every tag *T* model calculates probability the word is tagged by that tag (1) and selects tag with highest probability.

$$(1)\ P(T|W) = \frac{count(W,T)}{count(W)}$$

### 1.3. Results

Model got accuracy of 98% for *test_1* and 94% for *test_2*. Smaller accuracy in *test_2* is caused by trying to predict tags for words we haven't seen during training.

## 2. Test_2 difficulties

My guess is the problem is when model has to predict tags for words it didn't come across during the training phase. It then has no knowledge by which to select the tag for the word. This model handles it by adding tag *NO_DATA* indicating model has no information about the word. I think this is the limitation of this model since it does not generalize good enough.

In this situation maybe the statistically best thing to do is in to see which tag was used the most in the whole dataset. So, using a priori probability of tag *T* (2) to calculate probability of labeling word *W* with that tag, where *alltags* represents set of all tags that occurred during the training phase.

$$(2)\ P(T) = \frac{count(T)}{\sum_{tag\ \in altags} count(tag)}$$