CS 839 Stage 1 Report

Team: Lord of the Fries

Shantanu Singhal (singhal5@wisc.edu)
Sukanya Venkataraman (venkatarama5@wisc.edu)
Yogesh Chockalingam (ychockalinga@wisc.edu)

Entity Type

- We are identifying food entities the Yelp reviews dataset. Food entity is defined as any noun that can occur in a food menu excluding generic terms like food, water, lunch etc.
- Each food entity is annotated in its entirety using XML tags < />
    - I ordered <pizza/> and <kale juice/>

Training and Test Set Information

We labeled 364 documents with 1978 mentions of food entities. Then, we shuffled the documents and split them as follows:

	Number of Documents	**Number of Mentions**
Set I	243	1402
Set J	121	576

Model Performance

In addition to the 5 classifiers given (decision tree, random forest, support vector machine, linear regression, and logistic regression), we also used a gradient boosting classifier, as it is known to be robust and performs well.

The classifier which performed best on **set I** the first time was indeed the gradient boosting classifier. We will refer to this as classifier M.

Classifier (M)	**Precision**	**Recall**	**F1 Score**
Gradient Boosting Classifier	0.7637	0.8228	0.7922

We debugged to observe the misclassifications and found the best performing classifier (X) was still the gradient boosting classifier. This indicates the performance on the test set **(set J).**

Classifier (X)	**Precision**	**Recall**	**F1 Score**
Gradient Boosting Classifier	0.5270	0.7959	0.6341

We then came up with an extensive set of pruning rules to reduce the number of negative samples. We did this by creating a list of commonly found adjectives (adjectives.txt) and adverbs (adverbs.txt). Words which did not fall under these two categories were added to a blacklist (blacklist.txt). In addition to this, we also removed pronouns and other stopwords. (pruning.py)

We obtained classifier Y by combining X and the post-processing rules. This table indicates the performance of classifier Y on **set I**.

Classifier + Postprocessing rules (Y)	**Precision**	**Recall**	**F1 Score**
Gradient Boosting Classifier	0.9197	0.7713	0.8390

We observe that the precision now exceeds 0.9 while recall remains above 0.6.

Finally, we applied Y to the test set (**set J**).

Y	**Precision**	**Recall**	**F1 Score**
Gradient Boosting Classifier	0.5993	0.7448	0.6642

We can now see that both precision and F1 score has improved on set J after the addition of the post processing rules.

We finally save the model as a pickle file named *GradientBoostingClassifier.pkl* for later use, if necessary.