UNIVERSITY OF ILLINOIS AT CHICAGO
CS521 - STATISTICAL NATURAL LANGUAGE PROCESSING

Project Progress Report

MS.Umberto Di Fabrizio, MS.Vittorio Selo
November 15, 2015

I. PROBLEM STATEMENT

Our purpose is to create a recommendation system for Yelp. In order to achieve this we decided to exploit NLP techniques to extract *hidden features* in the users' reviews. Once we have a model with those features we can predict the score that a user would give to a business thus we can recommend to the user a business he will like.

II. IDENTIFIED MODEL

The main idea is to understand which are the features that a particular user observes in a business therefore we can 'profile' an user. For each user we build a model that generalize his tastes and we use it to predict the likelihood that the user will like an other business.
We use the reviews of a certain user together with the stars (=rank) that he gives to the businesses to create our features. Because of the dimensions of the dataset and the computational power we have access to, we decided to limit our investigation to businesses in the area of Edinburgh and only those which are restaurants (11611 reviews, 2375 users,1112 businesses). Anyway this work can be easily generalized to the all dataset.
For each user we scan his reviews, we detect the nouns that he uses[1] and for each noun we collect three features. Let M be the set of all users, R the set of all reviews. Given user m $\in$ M let r_m be the set of all reviews of user m then define X_m as the set of all nouns of the user m. Let n_{xmz} be the number of times that the word x belonging to X_m appear in review z $\in r_m$. Let s_{mz} be the rank that user m gave to review z.
Now we can define:

- **Frequency**: frequency of that noun compared to the other nouns used
  i.e. how much the user talks about X?
$$f(x,m) = \frac{\sum_{i \in r_m} n_{xmi}}{\sum_{i \in r_m} \sum_{k \in X_m} n_{kmi}}$$
- **Regularity**: how constantly is that noun used in the reviews
  i.e. does the user talks about X in most of the reviews or not?
$$r(x,m) = \frac{\sum_{i \in r_m} n_{xmi} \frac{1}{\sum_{k \in X_m} n_{kmi}}}{|r_m|}$$
- **Relevance**: how influent is the noun to predict the rank?
  i.e. is X important to the final business score?
$$i(x,m) = \frac{\sum_{i \in r_m} n_{xmi} \frac{s_{mi}}{\sum_{k \in X_m} n_{kmi}}}{|r_m|}$$

Once we have those 3 features for each noun of a user we select the top 30 (= best nouns) between those that have the highest values in both of the 3 features: we want nouns that are frequent AND regular AND relevant. Because it is an intersection operation in the best case we will have 30 nouns in the worst 0 (never happened, average is 26).
We run the same algorithm to collect noun and their features for the businesses.

III. PREPROCESSING

We tokenize our dataset using the tweeter tokenizer (keeps smilies ':)'), then we use the nltk package for python trained on the Penn Tree Bank dataset for the POS-tagging. For each noun we calculate the three features explained in the previous section. In the end for each user we have his *best nouns* with their relative 3 hidden-features and for each business we have its nouns with their 3 hidden-features. For each best noun of a user we look in the business to check if it has that word, in this case the business values for that word are collected otherwise [0,0,0] is assigned. In the end we have a vector:
[User best nouns][Business word in common with user]
for each user-business couple. So the length of the feature vector is 2* |best user noun|.

IV. CLASSIFICATION

For each user which has more than 20 reviews, we select 90% of reviews to be used in the train and 10% in the test. We train an SVM with the features vector as input and the rank of the business (accordingly to the user) as target output.
We take the business in the test (we know how the user

[1] we believe that those are the '*things*' that he observes in the businesses

ranked them) and we predict the score with the SVM. Right now the accuracy is 44.5%, whilst the majority class is 43.5%. The improvement right now does not seem exciting, anyway we believe that some of the words selected by the algorithm are not meaningful such as 'bit','restaurant' and other 'common' words. Our plan is to try to improve the word selection in order to extract better words and to use user that have more than 50 reviews so that each user can be modeled better.