**Text Categorization of Businesses**

**Charlotte Chen Dan Hui David Neiman David Tsui**

chc190@pitt.edu dth26@pitt.edu dsn9@pitt.edu dat83@pitt.edu

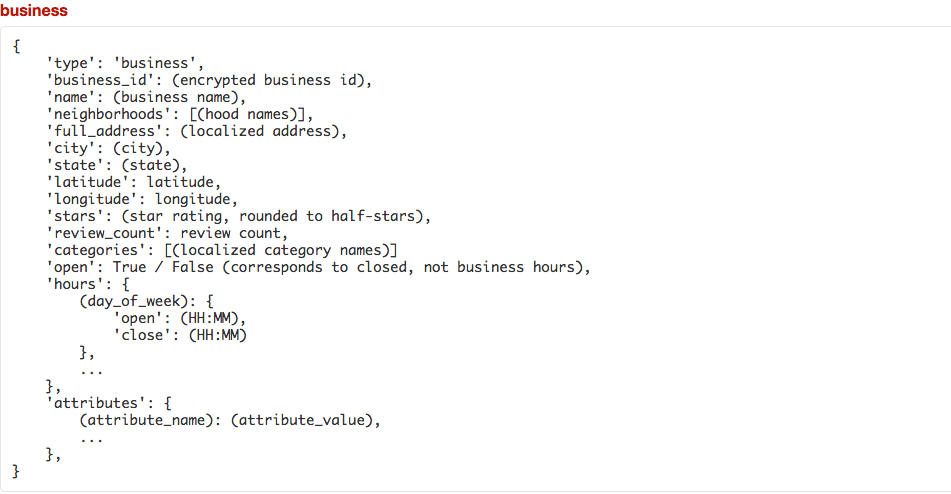
**Abstract**

**Introduction**

**Data**

We used the data from the Yelp Data Challenge Set, specifically the yelp\_academic\_dataset\_business.json file and the yelp\_academic\_dataset\_review.json file.

The business file was formatted with a unique business id, information such as address and number of reviews, and then a list of attributes. Some attributes, such as "Good for Dancing", were given true or false values, while attributes such as "Alcohol" were given more descriptive values. Other attributes, such as ambience, contained even more nested information, such as information about being trendy or romantic. One of the most important pieces of information was the list of categories provided. Business were grouped into many categories which were not mutually exclusive. We were specifically interested in the business id, categories, and set of attributes.



The review file contained the unique user id, review id, business id, information about the type of votes given, and the text review. We were specifically interested in the business id and the text review.



**Preprocessing**

We began by preprocessing the text reviews, looking to remove unnecessary words and symbols in the text. We began by removing apostrophes, and then fixing up those words by removing parts of the apostrophe phrase - for example, removing the "ll" in "she'll", or the "ve" in "would've", such that we were left the base word. We also removed extra punctuation, such as periods, question marks, and exclamation marks. After dealing with these, we moved on to tackling the actual words in the review. We removed "to be" verbs - "are", "is", etc. - as well as conjunctions, pronouns, and prepositions. We also removed numbers and additional spaces.

**Category Grouping**

The original Yelp dataset contains hierarchical grouping of categories, with 22 primary categories, 498 secondary categories, and 178 tertiary categories, for a total of 698 categories. Using different classifiers for each of these categories would have been unfeasible for both our language model and logistic regression models, so we aggregated the groupings into the primary categories. In the original setup, categories were not mutually exclusive; businesses could be grouped in multiple different categories that could be related. For example, there are businesses in the data set that are placed in three categories: as a restaurant (primary category), as Chinese restaurant (secondary category), and as a Taiwanese restaurant (secondary category). Our grouping procedure categorizes this business simply as a restaurant. In this way, we now assign businesses to one distinct category. Additionally, categories are now mutually exclusive.

Used sample of Yelp dataset (business and review). probably something like 5% of the businesses and the reviews

Problem Solution: Categorize businesses into categories

-Supervised learning: Language Model and Logistic Regression and Naive Bayes

-Evaluate: ten-fold cross validation. 90/10 split

-Precision and recall based on the number of categories

guessed correctly (precision) and the recall (of those predicted, how many out of the correct ones were selected)

Preprocessing: Using two files,

* business: 5% business
* All in json. general layout = ID (unique), category (how we evaluate), attributes (list of characteristics the business has - ie. parking yes no, mostly true/false, with a few with several choices like dresswear)
* review file: 5% review
* Straight review with customer id, business id, etc. What we need are the business id and the text
* So we get the business
* bid
* cat
* attributes
* 729 categories and that's too many classifiers to both run through for logistic regression and language model - too low accuracy, etc. So we create a hierarchy of concepts
* So we realize a lot of the categories appear flat, they are not actually flat. Such as Taiwanese, Chinese, and restaurant. But these are grouped under restaurant. We originally planned to manually assign it but after brief search found that Yelp provided hierarchical category tree in format of Excel files. So we had to go through excel and find the root category and get rid of the subcategories. We now had 24 categories. **Mutually exclusive!** One business cannot be classified into 2 different categories.
* Also eliminated stopwords. And used lemmatisation.
* So we decreased the vocab
* use those to train, evaluate
* then go back and use 10-fold cross-validation on the big file

Language Model

* Need frequency count of words said given categories
* Use dictionaries in Python with frequency count
* Use a business object
* bid uniquely identifies a business. Then use that for the category's frequency counts
* Originally we had multiple categories. Created category object to handle multiple categories and etc., this became a bit redundant when we simplified to one category per business
* Refer to language model equation

[http://sebastianraschka.com/Articles/2014\_naive\_bayes\_1.html](http://l.facebook.com/l.php?u=http%3A%2F%2Fsebastianraschka.com%2FArticles%2F2014_naive_bayes_1.html&h=aAQENduXC" \t "_blank)

* read that in pdf form
* multinomial equation #37.
* laplace smoothing
* Custom-made precision and recall

Logistic Regression

* Weka uses special file format called .arff file
* We had to convert JSON to arff
* No easy conversion
* features are @relation \_\_\_\_
* such as parking {T, F}
* @attribute
* @attribute
* @class {categories}
* @data [T, T, T, t] [cat]
* each data entry is a business. Array of features. And then what category it is
* generate one large arff file given the business file.
* Used Java to go through the folder of all the arff files
* Gives us the 10-fold cross, the prediction, the probability distribution, precision, recall