ILS Z534 Information Retrieval Project Report

Yelp Dataset Challenge

Team Members

Janak Bhalla (jbhalla@iu.edu)
Naveen Jetty (njetty@indiana.edu)
Pradeep Kumar Ravilla (pravilla@iu.edu)
Sujith Shivaprakash (sujishiv@iu.edu)
Jaini Vora (voraj@iu.edu)

Under the guidance of
Professor Xiaozhong Liu
at
School of Informatics and Computing
Indiana University Bloomington

Fall 2016

Introduction

The yelp dataset includes information about different businesses in 10 cities across 4 countries. In this project, we used this dataset to 1) predict the label of a business from review and tip text and 2) predict the rating from its review text. Accurate classification of a business into categories can help Yelp to improve business suggestions to the Yelpers. Predicting the rating from a review will be useful to Yelp and the customers, as new customers will receive handy inputs about a business from reviews written by customers who visited earlier and will be able to mitigate the ratings given by spam users without providing any reviews.

Task 1: Predict Business label from Review and Tips of the business.

1. Data Preprocessing

The yelp dataset provided was properly organized and shared in .json files. The major problems which arise from data is due to missing values or null values in the dataset. Fortunately, the Yelp challenge has gone through 7 rounds, so the dataset provided given was clean and did not have any these issues which made working with it easier. Even then we had to process the data to filter it according to our needs.

We did the following things to filter the text and extract only meaningful data relevant to the current problem in hand:

- Converted all words to lowercase.
- Removed stop words in English language using nltk library.
- Removed punctuations from the text. This way we can remove the smileys which does not add any value to business category.

We collected all the reviews and tips for each business into one string. This string was then used to create the features to be used for training the model.

2. Feature Engineering

We followed two approaches for creating feature vectors using the review and tips text of the businesses.

First Approach: Reviews and tips of each business was used to create a long string. These were collated for all the reviews and then vectorized using tf-idf vectorizer to find scores of each word present in the review word vector. We converted the words in number vectors because machine learning algorithms work well with numbers only. These tf-idf vectors were passed as training input to machine learning in the vector space model and were used as features to be given as input to an appropriate machine learning algorithm.

Second Approach:

Hypothesis:

Classification of businesses from the reviews and tips will be majorly decided from the nouns, and the adjectives which describe the nouns present in the reviews. Based on this hypothesis, we build our second approach to creating the feature vectors.

We used 'The Penn Treebank POS tagset' to tag each word with its appropriate part of speech. After tagging each review and tip with its part-of-speech, we filtered out and kept only below categories.

- NN : Noun, singular or mass
- NNS : Noun, plural
- NNP : Proper noun, singular
- NNPS: Proper Noun, plural
- JJ : Adjective
- JJR : Adjective, comparative
- JJS : Adjective, superlative

Using these tagged words we created the text corpus and then used the tf-idf vectorizer to create feature vectors to train the model.

3. Classification & Evaluation

The problem at hand is a good example of a multilabel classification problem, which means each business can belong to multiple categories. To achieve this, we use the One vs Rest

approach. We train as many classifiers as there are class labels, and each classifier learns a model for one label, and predicts whether a feature vector belongs to that target label or not. For the classification of each individual label in the OnevsRest classifier, we used Linear SVC and Multinomial Naive Bayes.

To evaluate our approach we used cross-validation as suggested by Professor. The outputs of all classifiers was collated, and evaluated metrics like precision, recall and F1 score for the model.

4. Results & Conclusion

	First Approach				**Second Approach**			
Category	Precision	Recall	F1	Training samples	Precision	Recall	F1	Training Samples
Restaurants	0.95	0.95	0.95	9687	0.94	0.92	0.93	437
Salad	0.88	0.26	0.40	226	0.67	0.2	0.31	10
Sandwiches	0.85	0.55	0.67	1020	0.88	0.6	0.71	47
Seafood	0.88	0.39	0.54	264	0.5	0.17	0.25	12
Shopping	0.96	0.87	0.91	1746	0.94	0.83	0.88	89

From both the approaches, we see that the precision & recall of second approach was much better than first even with less number of training examples. Since the second approach uses word_tokenize and POS_tagging it took a lot of time to execute even to build a single model. We trained the model with just 5000 examples in the second approach.

Both Linear SVC and Multinomial NB are parameter dependent classifiers. We run both of them with their default parameters. There may exist a set of parameters for SVC that would have caused SVC to surpass the performance of NB. Also, SVC in general performs better when there is a lot of overlap in the features and the support vectors are not affected by it. Since many words appear in a lot of reviews, there may have been feature overlap in the problem at hand.

We could further improve the better prediction of business category if we use word embeddings instead of using Tf-IDf vectorizer.

Task 2: PREDICTION OF STAR-RATING USING USER FEEDBACK

1. INTRODUCTION

The given Yelp Data had the following structure.

```
{
            'type': 'review',
            'business_id': (encrypted business id),
            'user_id': (encrypted user id),
            'stars': (star rating, rounded to half-stars),
            'text': (review text),
            'date': (date, formatted like '2012-03-14'),
            'votes': {(vote type): (count)},
}
```

In this task we tried to predict the star rating given any user feedback by using the data provided. Given the data which consisted of the above structure, we extracted ***'stars','text'*** from the data which was stored in a MongoDB server.
On performing a brief data exploration on the number of words used in most of the reviews, we zeroed on considering reviews which had review text between 100-200 words and extracted them.

2. FEATURE ENGINEERING

Since the entire data taken into consideration had text reviews along with star rating which was considered as a categorical data, we initially had to convert the text data into numeric data using text preprocessing.
We first indexed the entire corpus using **Lucene** using a Standard Analyzer, this took care of getting rid of Stop-Words and performing Stemming, here each review was stored as a document having the following fields:
<REVIEWID> (STRING FIELD), **<STAR RATING> (STRING FIELD), <REVIEW> (TEXT FIELD)**

2.1 BAG-OF-WORDS MODEL

Our brute-force approach to convert each user-review into numeric data was using a term frequency approach, our idea was to consider the top 1000 frequent words from the entire text corpus which had been indexed then use a bag-of-words model to add each word in a review to its respective bin generated from the top 1000 frequent words. However, 1000 words still consisted of unwanted stop-words and a few expressions like smileys and numerals, we performed the operation of removing stop words the second time thereby reducing the number of frequent words to **708.** One dataset constructed out this had 0.1 million reviews along with 708 different features and included a class label taking five different values **{1,2,3,4,5}** depicting the star rating. We converted the data into a **<Feature , Label>** format.

2.2 CHI-SQUARED TEST FOR FEATURE SELECTION

Suspecting that there might be several redundant features extracted using Bag-of-Words we performed a Chi-Squared goodness of fit test to extract the relevant features that might help during the process of classification. The Chi-Squared test uses **Cramer's Coefficient** to find association between two different discrete Random Variables, where a value of 0 corresponds to no association between the variables while 1 represents a complete association (when the two variables have the same value).
On performing this process, the number of features reduced to 118, which seems to be a very significant reduction.

2.3 INFORMATION GAIN FOR FEATURE SELECTION

Information.gain is a function that selects the best combination of attributes according to its "Information Gain". This function is based on entropy. Entropy characterizes the purity/impurity of an arbitrary collection of words. Information Gain is the expected reduction in entropy caused by partitioning the examples according to a given attribute.On performing this process, the number of features reduced to 113, which seems to be a very significant reduction.

2.4 TF-IDF MODEL

Since selecting top 1000 frequent words would result in domain specific stop words as features, it would be difficult to obtain considerable results. Initially, we performed general preprocessing involving stopwords removal, stemming and lemmatization. Then for this purpose, we made use of TF-IDF model to achieve our task. For this task, our approach was to consider all the terms in all the reviews as features. Since we are computing TF-IDF, the domain specific stopwords would be an assigned a lower value helping us to achieve our task.

3. CLASSIFICATION & EVALUATION:

We have used two different classification algorithms for this task, **Multinomial Naïve Bayes** and **Softmax Regression** or **Multinomial Logistic Regression** and we have 0.1 million of randomly sampled data for training and 40k randomly sampled rows of data as testing. Taking into account Naïve Bayes as a baseline solution along with the data set constructed using the **Bag-of-Words** model**.**
This being a task of classification of categorical variables we are using **PRECISION** and **ACCURACY** to be our evaluation metric.

3.1 BAG-OF-WORDS MODEL

3.1.1 NAÏVE BAYES

On performing a classification task using a Naïve Bayes model, constructed a confusion matrix having the following cells shown below.
Confusion Matrix for each five different class labels:

	actual				
Predicted	1	2	3	4	5
1	2012	923	586	510	496
2	465	832	82	844	635
3	230	736	1531	1449	780
4	129	365	1140	2769	1737
5	822	1086	2322	6660	10120

Using this as a reference to calculate precision and accuracy on a dataset of 0.1 million rows we get an **Accuracy of 43.16%** and a **Precision of 44.44%, 23.13%, 32.39%,45.09% and 48.16%** for Class Labels **{1,2,3,4,5}** respectively.

3.1.2 SOFT-MAX REGRESSION

Confusion Matrix for each five different class labels:

	actual				
Predicted	1	2	3	4	5
1	2282	784	332	142	113
2	350	906	418	154	65
3	326	1106	2290	1197	366
4	179	534	2107	5417	2237
5	521	612	1253	5322	10987

Using this as a reference to calculate precision and accuracy on a dataset of 0.1 million rows we get an **Accuracy of 54.7%** and a **Precision of 62.46%, 47.8%, 43.33%,51.71% and 58.76%** for Class Labels **{1,2,3,4,5}** respectively.

The method of soft-max regression appears to have a better accuracy compared to Naïve Bayes model.

3.2 CHI-SQUARED MODEL

This model consisted of 118 relevant predictors selected from the BAG-OF-WORDS MODEL using chi-squared goodness of fit test.

3.2.1 NAÏVE BAYES

	actual				
Predicted	1	2	3	4	5
1	2024	894	514	337	267
2	381	797	803	609	392
3	263	784	1411	1263	649
4	179	396	1270	2402	1364
5	811	1071	2402	7621	11096

Using the confusion matrix we get an **Accuracy of 44.32%** and a **Precision** of **50.14%, 26.72%, 32.28%,42.80% and 48.24%** for Class Labels **{1,2,3,4,5}** respectively.

3.2.2 SOFT-MAX REGRESSION

	actual				
predicted	1	2	3	4	5
1	2041	821	379	175	141
2	501	960	636	260	113
3	184	612	1280	683	192
4	483	1038	3069	6736	3867
5	449	511	1036	4378	9455

Using the confusion matrix we get an **Accuracy of 51.18%** and a **Precision** of **57.37%, 38.86%, 43.37%,44.33% and 59.73%** for Class Labels **{1,2,3,4,5}** respectively.
We see a similar observation here that the soft-max regression performs significantly better than that of Naïve Byes classifier.

We also observe that the soft-max regression for CHI-SQUARED MODEL though having similar accuracy to that of soft-max regression of BAG-OF-WORDS model, yet we see there being a

significant difference in precision of both the models, with BAG-OF-WORDS model having a higher precision.

3.3 INFORMATION GAIN MODEL

In this model, we calculated the total number of words with non-zero Information Gain. The count was 113. So we selected top 100 as the cut off from the 113 features.

3.3.1 NAÏVE BAYES

	Actual				
Predicted	1	2	3	4	5
1	1912	821	465	329	214
2	465	776	744	582	383
3	268	764	1423	1294	593
4	181	445	1274	2345	1350
5	795	1068	2458	7913	11138

Using the confusion matrix we get an **Accuracy of 43.98%** and a **Precision** of **51.10%, 26.30%, 32.77%,41.91% and 47.65%** for Class Labels **{1,2,3,4,5}** respectively.

3.3.2 SOFT-MAX REGRESSION

	Actual				
Predicted	1	2	3	4	5
1	1893	714	337	169	98
2	546	909	621	272	135
3	230	785	1492	846	231
4	407	900	2771	6203	3241
5	545	566	1143	4973	9973

Using the confusion matrix we get an **Accuracy of 51.18%** and a **Precision** of **58.95%, 36.60%, 41.62%,45.87% and 57.98%** for Class Labels **{1,2,3,4,5}** respectively.

We see a similar observation here that the soft-max regression performs significantly better than that of Naïve Bayes classifier.

We also observe that the soft-max regression for INFORMATION GAIN MODEL though having similar accuracy to that of soft-max regression of BAG-OF-WORDS model, yet we see there being a **significant difference in precision** of both the models, with BAG-OF-WORDS model having a higher precision.

3.4 TF-IDF MODEL

3.4.1 NAÏVE BAYES

On performing a classification task using a Naïve Bayes model, constructed a confusion matrix having the following cells shown below.

Confusion Matrix for each five different class labels:

	Actual				
Predicted	1	2	3	4	5
1	4773	1210	459	264	356
2	445	1013	495	194	94
3	181	666	1693	794	182
4	152	485	1878	5351	2527
5	365	340	637	3741	11705

Using this as a reference to calculate precision and accuracy on a dataset of 0.1 million rows we get an **Accuracy of 61.33%** and a **Precision of 67.58% 45.20% 48.15% 51.48% 69.72%** for Class Labels **{1,2,3,4,5}** respectively.

3.4.2 SOFTMAX REGRESSION

Confusion Matrix for each five different class labels:

	Actual				
Predicted	1	2	3	4	5
1	4281	1406	972	1326	2356
2	932	1317	1373	1657	1196
3	255	536	1511	1816	1143
4	161	291	1000	3919	3905
5	287	164	306	1626	6264

Using this as a reference to calculate precision and accuracy on a dataset of 0.1 million rows we get an **Accuracy of 43.23%** and a **Precision of 41.39% 20.34% 28.72% 42.24% 72.44** for Class Labels **{1,2,3,4,5}** respectively.

The method of Naïve Bayes appears to have a better accuracy compared to soft-max regression model.

4. INDIVIDUAL CONTRIBUTION:

Jaini Vora	● Creation of Lucene Index for calculation of top 1000 words for all reviews ● Wrote the code for Information Gain and analysed its behavior with Naive Bayes and Softmax Regression
Janak Bhalla	● Created TF-IDF vector components for all words in the reviews and analysed its behavior with Naive Bayes and Softmax Regression.
Naveen Jetty	● Data preprocessing. ● Cleaned the data and exported it to the Mongo . ● Wrote code to create Feature Vectors from the reviews and tips using NLTK packages Lemmatizer and removed stop words. ● Worked on various partitions of dataset training with Multinomial NB and cross validating the test and train data to get the Precision, Recall and F Measure.
Pradeep Kumar Ravilla	● Wrote code to reduce the dataset to remove business with less number of reviews. ● Worked on the OneVsRest Classifier with Linear SVC Classifiers. ● Wrote code to process tokens that and tag POS_Tagger, using NLTK's toolkit ● Thought about the second approach to feature vectorization to reduce the corpus of Reviews and Tips to contain only nouns and adjectives. ● Cross validated the test and train data on the POS_tagged corpus to get SVC and Multinomial Classifiers and to get Precision, Recall and F Measure.
Sujith Shivaprakash	● Creating Data Set for Term frequency for top 1000 words ● Chi-Sqaure test to extract relevant features, Multinomial Naive Bayes and Soft Max Regression , Evaluation.

5. REFERENCES:

1) Prediction of rating based on review text of Yelp reviews - Sasank Channapragada, Ruchika Shivaswamy [https://cseweb.ucsd.edu/~jmcauley/cse255/reports/fa15/031.pdf]

2) Automatically Categorizing Yelp Businesses [https://engineeringblog.yelp.com/2015/09/automatically-categorizing-yelp-businesses.html]