Yelp Data Challenge

Team Members
Aniket Bhosale
Shivraj Nimbalkar
Suraj Sonawane
Mayur Tare

1.Introduction

1.1 Motivation

Nowadays people’s decision on which place to visit and what to eat is subject to other people’s opinion. Yelp is a website which has turned into a massive database of such opinions casted as reviews by people. With such a rich source of information it is possible to deduce some interesting findings which would help users make decisions. This motivates us to think of zillion ways in which we could analyze the underneath relationships across the different dimensions of the dataset.

1.2 Problem Statement

This report addresses two problem statements:

- **Task 1: Predict the categories of restaurants**
  Category information is useful for business recommendation. Not all the restaurants have high quality metadata or informative categories. Thus in this task we predict the categories for the restaurant based on the review text alone.
- **Task 2: Predict restaurant ratings based on review text alone**
  Business ratings on Yelp may suffer from being subjective and biased towards users personality. Also user need to get a better understanding about a business from its ratings rather than reading through long text of reviews. Thus in this task we predict the user ratings for a restaurant based on the review text alone.

2. Data Pre-processing

2.1 Dataset[1]

The academic dataset includes data from Phoenix, Las Vegas, Madison, Waterloo and Edinburgh which contains information about 42,153 businesses, 1,125,458 reviews, 31,617 check-in sets, 252,898 users and 403,210 tips. Out of this massive dataset we narrowed down our category prediction task for Restaurant businesses and useful votes for reviews > 4. Leaving us with 55,303 reviews for 6965 unique business IDs.

2.2 Preprocessing steps

- Load review and business JSON data into MongoDB
- Select reviews for Restaurants with useful votes greater than 4 and club them with businessID
- **Task 1:**
  - Join the business and review JSON to fetch the corresponding categories
  - Filter all unique categories and club the reviews with categories
  - Clean up the data by doing stop word removal using Standard Analyzer and custom stop word list
- **Task 2:**
  - Categorize reviews into three classes Low(1-2 stars), High(4-5 stars), Medium(3 stars)
  - Build custom list of adjectives
  - Apply IR techniques to the custom list to build the feature vector.

2.3 Tools and Libraries

- RoboMongo (MongoDB GUI)
- Weka ToolKit (Machine Learning Algorithms)
- JSON Simple (JSON Parser)
- Lucene API
- Apache POI (Java API for Microsoft Documents)
- Python Scripts

3. Approach

We decided to combine both machine learning as well as IR approach to solve the category and rating classification problem. For the machine learning approach category of the restaurant is the class label and the words appearing in the review act as features and for task 2 Star rating for a particular review is the class label and adjectives appearing in the text act as features. We implemented the IR approach to fetch the top words to build the final feature vector.

3.1 IR approach for Feature Selection

Our implementation fetched the top words based on the TF $\times$ IDF scores. In our case the TF-IDF was calculated for each term occurring across all the reviews for a given category thus we name it as Local TF and Local IDF. With this assumption we executed our program and found for one of the categories example INDIAN following words were fetched by our program:

With Local TF/IDF	food indian chicken buffet restaurant lunch curry rice time masala

As you can observe from the above results many words such as “food”, “lunch” , “restaurant” are less informative which do not represent the category even being in top occurring word list.Moreover after observing the results from few other categories we could see a trend of such common words occurring in top word list across various categories.Thus we concluded that Local TF/IDF wouldn’t suffice our requirement and we need to device some factor which would suppress these words. Thus we came up with the concept of Global IDF.
Global IDF reduces the weights of the words that appear in the top word list across various categories terming them as less informative.
Thus our final formulation can be given as:

Weight of Word = (Local TF $\times$ IDF) * Global IDF

Global IDF = N / O where,
N = Total number of class labels
O = Number of Class Labels in which word occurs in Top words

As a result of implementation of the above weight formulation for calculating top words resulted in much relevant and informative results which can be seen in below table.

With Local TF/IDF & **Global IDF**	indian masala india naan buffet chicken naan paneer tikka curry

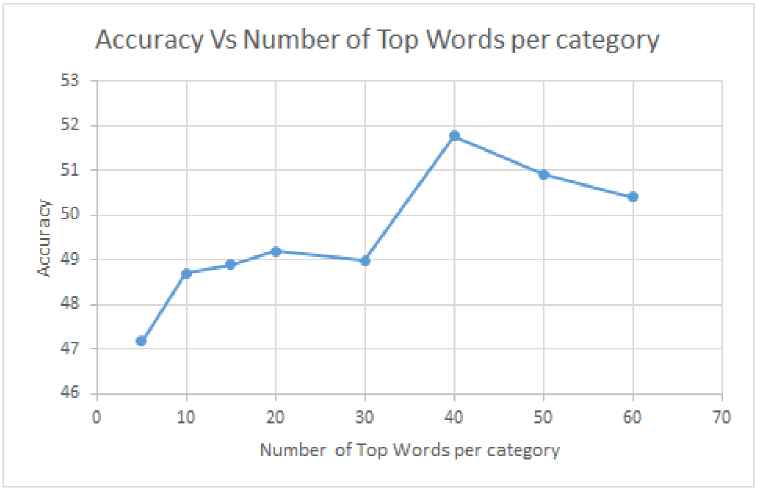
3.2 Machine Learning Approach
Once the top words are fetched for each category we combine all such words to form a feature vector and then build the data set for the reviews with useful votes > 4. We performed n-cross validation on our dataset and reported the result in Weka.

4. Results, Evaluation Metrics and Analysis

Experiments (Naive Bayes Algorithm):

4.1 Task 1

- **Number of top words per Category:**Below graph shows that the accuracy is maximum when the number of top words chosen from a category is 40 after which the accuracy starts to fall. Thus we decided to choose top 40 words from each category to build our feature vector.



- **Baseline Method Analysis:** We checked the performance on some baseline methods i.e., only use local, and only us global IDF and we found that our method outperforms these baseline methods with an accuracy of 51.9. Also on further adding weights to the Local and Global IDF we could see an improvement in AUC_ROC value marked in gray below

Feature selection using	Performance Measure	
	Accuracy	AUC-ROC
only Local TF/IDF	42.54	0.73
Local TF and Global TF	46.97	0.72
Local TF/IDF and Global TF	51.9	0.7
weighted Local and Global TF/IDF (weights : Local IDF - 2, Global IDF - 1)	51.78	0.845

- **Weight for Global and Local:** It can be observed that the accuracy is highest when the importance of local IDF is made twice as global IDF

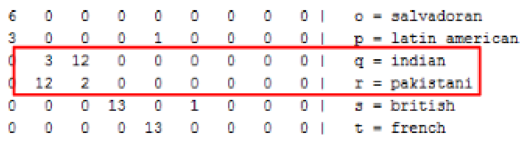
Weights Analysis for Local and Golabal IDF

Weights	Accuracy	AUC-ROC
Local IDF - 0.25 Global IDF - 0.75	46.1	0.85
Local IDF - 0.75 Global IDF - 0.25	51.68	0.835
Local IDF - 2 Global IDF - 1	51.78	0.845

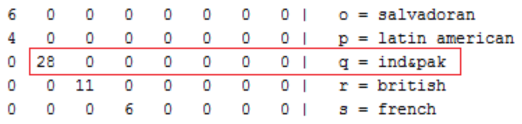
- **Comparison of Feature Selection Techniques:** It is observed that Information Gain for feature selection outperforms Principal Component Analysis.

	Accuracy	AUC-ROC
InfoGain Weka	50.81	0.847
PCA Weka	22.9066	0.648

- **Confusion Matrix Analysis:** In the confusion matrix we observed that in case of 'indian' and 'pakistani' cuisine 12 of the indian examples were classified as pakistani cuisine and vice versa. This behaviour can be because these two cuisines are quite similar and thus clubbed them together as a single category name ind&pak. We observed that doing this we could classify all the 28 examples correctly improving our precision for these categories.



```
 6    0    0    0    0    0    0    0    0 |   o = salvadoran
 3    0    0    0    1    0    0    0    0 |   p = latin american
 0    3   12    0    0    0    0    0    0 |   q = indian
 0   12    2    0    0    0    0    0    0 |   r = pakistani
 0    0    0   13    0    1    0    0    0 |   s = british
 0    0    0    0   13    0    0    0    0 |   t = french
```



```
 6    0    0    0    0    0    0    0 |   o = salvadoran
 4    0    0    0    0    0    0    0 |   p = latin american
 0   28    0    0    0    0    0    0 |   q = ind&pak
 0    0   11    0    0    0    0    0 |   r = british
 0    0    0    6    0    0    0    0 |   s = french
```

4.2 Task 2:

- **Precision-Recall Analysis:** It can be observed that we get a very high recall value for the prediction of class label "High ratings" which suggests that our method does a good job of building a vocabulary for predicting the review ratings for range 4-5, but it doesn't quite do a good job with predicting low and medium class.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.246	0.066	0.529	0.246	0.336	0.693	0 (Low)
0.01	0.004	0.296	0.01	0.02	0.547	1 (Medium)
0.94	**0.8**	**0.663**	**0.94**	**0.778**	**0.663**	**2 (High)**

● As you can observe a good true positive rate for class "High" in confusion matrix shown below,

Predicted result			*Confusion Matrix*	
Low	**Medium**	**High**		
3146	103	9547	**Low**	**Actual Result**
812	80	7008	**Medium**	
1986	87	**32623**	**High**	

● **Machine Learning Algorithms:** After running various machine learning algorithm in weka we could observe that the accuracy more or less remains the same across different algorithms.

Algorithm	Accuracy
Naïve Bayes	64.72
Bayesian Network	65.99
Decision Treess	65.30
Bagging with AdaBoost	64.40
Random forest	63.90

● We also compared principal component analysis and Attribute selection using information gain methods with our proposed methodology using local TF-IDF with Global IDF. The performance is compared using accuracy as measure as follows,

	Accuracy
InfoGain Weka	63.87
PCA Weka	34.56

Attribute selection using information gain performs almost similar to our approach. The reason for this behaviour can be explained as the concept of TF-IDF and Global IDF is very similar to Information gain and hence similar results. Feature selection method using Principal Components analysis performed worse than our approach which can be observed from above accuracy numbers.

5. Conclusion

Based on the analysis results for category prediction task we can broadly conclude that our method of calculating weights of the word for feature selection outperformed the baseline methods such as only Local IDF. Also we analysed various ways in which we could club the class labels without affecting their significance purely on basis of similarity of cuisines. In case of rating prediction task starting with pure adjectives list and building on features based on this custom adjectives list yielded a good recall on predicting stars 4-5 but didn't perform as well against the stars 1-2-3.Also the performance across various machine learning algorithm was consistent for all.Thus we need to think of a more sophisticated ways of to build features corresponding to predicting stars 1-3. Also in rating prediction task information gain for feature selection performed better than PCA.

6. Code Repository on Github

Our code can be found below mentioned link:

Task1: https://github.com/InformationRetrieval/YelpDataChallege

Task2: https://github.com/InformationRetrieval/RatingPredictionFromTextYELP

7. References

[1] http://www.yelp.com/dataset_challenge