What makes a review useful, funny or cool on Yelp.com

Amira Ghenai AGHENAI@UWATERLOO.CA
University of Waterloo, 200 University Ave W, Waterloo, ON N2L 3G1 Canada

Abstract

In the era of big data and social networks, user-generated reviews are becoming essential and valuable resources for product information. In this paper, we first explore the most relevant features that make a review 'useful', 'funny' or 'cool' in Yelp site using various feature selection techniques. We, then, apply different supervised machine learning techniques and evaluate the classification accuracy of each approach. Finally, by testing the performance of the classification approach, we reach a 95% accuracy to recommend the most 'useful', 'funny' and 'cool' reviews in Yelp.

1. Introduction

With the emergence of the web, consumers are being exposed to a huge number of choices so recommendation systems were introduced to help those users navigate over complex information spaces. Previously, recommender systems suggest items to users based on their interest from the user's past purchase decisions. Nowadays, social web services like Amazon, Netflix and TripAdvisor embrace the world of *user-generated content* and *social web* to help users in buying decisions. An example would be people's reviews and opinions about different products on online stores that can serve as *recommendation explanation* and can help users evaluate the products more effectively. The available growing number of reviews motivates a new challenge on how to predict the most meaningful ones. This is mainly important because some reviews may be vague and misleading, for example if reviews are poorly authored and hard to understand, and others may be unbalanced or written by self-interested parties. Some systems address this issue by assigning different helpfulness ratings for each review but this may be a very sparse and varied solution especially with the increasing availability of product reviews. Thus, there is a huge need to enhance user experience by building systems that assist users in navigating though the vast and sometimes unreliable source of user-generated content.

In this paper, we address these issues in the context of user-generated product reviews and describe a classification-based approach to suggest helpful reviews. Briefly, we aim to design a classification-based approach on user's reviews of Yelp site to identify whether they are helpful with a degree of confidence. The definition of a helpful review in Yelp site is a review that has either been rated as *useful*, *funny* or *cool* review as figure 1 shows. In this context, our classification based approach suggests whether a review is helpful or not by classifying it as either being helpful, funny or cool. In particular, we evaluate different machine learning approaches using a variety of different feature selection techniques on a large-scale dataset from Yelp site. We focus on features relating to users, businesses and the *structure* and *readability* of reviews and examine the classification performance provided by these features to better understand what makes a review to be classified as either helpful, funny or cool. The classifier results may be presented to users as review recommendations.

Briefly, this paper is organized as follows. In the next chapter, we present a list of related work to classifying reviews. Next we describe the feature sets used for classification. Then we explain the feature selection techniques and the classification approaches that we consider in this work, which is followed by an evaluation of our approach using large collections of Yelp reviews. We conclude by discussing the significance and applicability of our approach and an outline for future work in this area.

Project report for CS886: Applied Machine Learning.
University of Waterloo, Fall 2014.

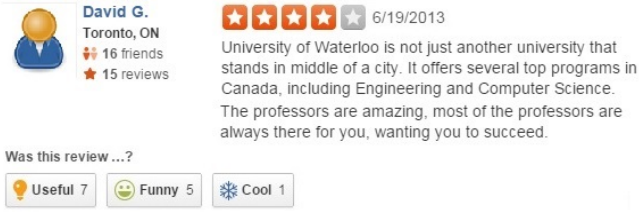


Figure 1. The interface of the review evaluation mechanism on Yelp.com

2. Related Work

A lot of effort has been invested into studying how information can be extracted from social web to help users make decisions more efficiently. The study (Fan and Khademi, 2014) used Yelp dataset to show that it is possible to predict a business rating from text reviews only. Selecting restaurants as a target business category, this approach created a bag of top frequent words as features and three feature generation methods were explored. Considering rating prediction from words bags as a regression problem, four different machine learning techniques were implemented to achieve best rating prediction.

In (O'Mahony and Smyth, 2010), a classification approach was presented to distinguish between helpful and non-helpful Amazon books reviews. Authors showed how the quality of characters representation in reviews effect its helpfulness. Readability test scores were implemented and showed to be correlated and significant in detecting how helpful a review is.

Furthermore, different machine learning techniques were investigated in (OMahony et al., 2010) on TripAdvisor hotel reviews to design a classification-based approach that identifies whether reviews are helpful or not with a degree of confidence. The classifier results were presented to users as review recommendations based on confidence ranking. Authors used social network information, user's general information, and reviews information to generate features for the classification approach.

3. Classifying and Recommending Reviews

The main objective of this study is to distinguish the different types of helpful reviews written for products using a machine learning technique. We considered a collection of reviews from Yelp challenge dataset where users can rate reviews as either being useful, funny or cool. To make sure that we are only using helpful reviews, we only considered reviews with at least one vote and to distinguish between useful, funny or cool reviews, we consider a review useful if most of the review raters have found it useful, funny if most of the review raters have found it funny and cool if most of the review raters have found it cool. A detailed explanation of the labeling strategy is presented in section 3.3.

Considering only reviews with a minimum of one vote does not consider ones which fail to attract users and would be helpful at the same time. For example, in the Yelp dataset (Yelp, 2004-2014) we will be using in this project, more than 45% of Restaurant reviews have zero votes and are not considered in the studies. In our approach, the classifier will be trained with reviews that attract a mass of votes in order to classify helpfulness of arbitrary reviews, even those receiving zero votes as feedback.

3.1. Description of Available Data

Yelp was funded in 2004 as a way for users to rate local businesses from 1-5 stars and write text reviews. We were mainly interested in Yelp dataset because, to our knowledge, there has been little research on this data recommendation improvement. Furthermore, Yelp consumer's interests are continuously growing.

The work presented in (Hajas et al., 2014) and Hood et al.[1] are examples of recent studies done on Yelp dataset. The Yelp dataset used in this project has been provided by Yelp website (Yelp, 2004-2014) and we will mainly be using the following files: Review.json, Users.json, Business.json and Checkins.json where each file is composed of a single object type, one json-object per-line. Reviews.json file contains information about the review such as the review text, the date when the review was written, the user_ID of the user who wrote the review and the business_ID of the business the review was written on. This file will be used to extract review features which will be explained in section 3.2.2. Users.json file gathers information about Yelp users such as the number of reviews a user wrote (review-count), the average user star rating, the number of friends and fans a user has, the number of optional compliments a user received, the votes counts showing the user's usefulness, funniness or coolness...This file will be mainly used to extract the user features presented in 3.2.1. Business.json lists information related to the business such as the number of total reviews a business received, the category the business belongs to, the business opening hours, the

[1] http://www.yelp.ca/html/pdf/YelpDatasetChallengeWinner_InferringFuture.pdf

average business star rating... Checkins.json contains information about the number of people visiting a special business within a specific time slot. Information about the business and the checkins will be used to extract the business features explained in 3.2.3.

Yelp dataset consists of 715 different business categories which present a variety of information domains. In this paper, we will mainly be focusing on "Restaurants" category (the category with the highest number of businesses as figure 2 shows) and "Shopping". We were mainly interested in implementing the feature

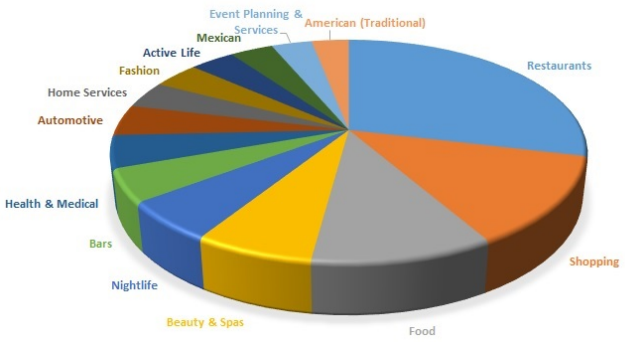


Figure 2. Top 15 categories available in Yelp dataset

selection and the classification-based recommender on different categories to gain more confidence in generalizing the outcomes of the experiments on other domains.

Furthermore, we restricted our reviews to only consider instances with at least one vote for "Shopping" category and ones with at least 5 votes for "Restaurants" category. The reason behind restricting "Restaurants" reviews to have at least 5 votes is for scalability concerns in order to be able to run experiments with the restricted number of reviews. Finally, we ended up with 86645 "Restaurants" reviews from a total of 814165 and 36307 "Shopping" reviews from a total of 63422. Table 1 summarizes the details about every category we will be using in our experimental study.

Table 1. Restaurants and Shopping categories statistics from Yelp dataset

Category	Reviews	Business	'Useful'	'Funny'	'Cool'
Restaurants	86645	19445	69126	39	151
Shopping	36307	6428	25673	1674	1699

3.2. Classification Features

Prior to performing classification, reviews are represented as a set of features that are gathered from three different categories which are either extracted from individual reviews or from a wider reviewing activity in the community. We present each feature category as follows:

3.2.1. User Reputation Features

We believe that user characteristics effect how helpful a review can be. For example, if a user has been actively rating in the past and has a large number of friends, he/she is more likely to effect others' opinions. In this context, we will consider the user average useful votes (U1) and the standard deviation of useful votes (U2) to see how useful user's reviews are, the user average funny votes (U3) and the standard deviation of funny votes (U4) to see how funny user's reviews are, the user average cool votes (U5) and the standard deviation of cool votes (U6) to see how cool user's reviews are, the number of friends and fans a user has (U7) & (U8) respectively and the number of optional compliments user received (U9) to see how popular a user is, the user review count (U10) and the average review count among all reviews (U11) and the standard deviation (U12) to see how active a user is and finally the percentage of helpful reviews with votes count greater than H over all user helpful reviews (U13). (H=5 in this work)

3.2.2. Review Features

These features are divided into two sub-categories: *structural features* and *readability features*. Starting with the structural features which refer to the structure of the review text, we consider the percentage of uppercase and lowercase characters in the text(ST1), the percentage of uppercase characters in the text (ST2), the ratio of the number $< br >$ and $< p >$ HTTP tags in the text to the total number of characters in the text (ST3), the number of words in the text (ST4), the number of complex words (words with 3 or more syllables) in the text (ST5), the number of sentences in the text (ST6), the average number of syllables per word (ST7) and the average number of words per sentence (ST8).

Moving to the readability features, we will be considering the readability test scores computed from: Flesch Reading Ease (R1) that computes reading ease on a scale from 1 to 100, with lower scores indicating a text that is more difficult to read (e.g. a score of 30 indicates 'very difficult' text and a score of 70 indicates 'easy' text), Flesch Kincaid Grade Level (R2) that

translates the Flesch Reading Ease score into the US grade level of education required to understand the text, Fog Index (R3) which indicates the number of years of education required for a reader to understand the text, SMOG (R4) that indicates the years of education needed to completely understand a text and ARI (R5) that is designed to gauge the understandability of a text (DuBay, 2004).

3.2.3. Business Features

These features are related to the reviewed business. We consider the star rate given to a business by the user (B1), the number of optional attributes assigned to a business (B2) that shows how well the business is informative. Then, the number of users visiting the business per day (B3) and the number of users reviewing it (B4) give an idea about the business' successful level which might be strongly related to the review usefulness. Finally, we will consider the mean (B5) and standard deviation (B6) of all users ratings on a particular business.

Even if we end up having 32 different features, there is no guarantee that those features are not redundant or that they are all important in the classification. Additionally, the presence of irrelevant features may badly effect the classification. To get the best results, we will first train with all 32 features, then we will perform different feature selection algorithms to pick the K most important features and end up with a matrix of N reviews by K features. Considering only a smaller set of features, that end up being more important than others in the classification task, answers the question of what makes a review in Yelp either being classified as useful, funny or cool.

3.3. Review Recommendation as Classification

Yelp dataset provides its users the ability to rate the review as being 'useful', 'funny' or 'cool'. To label the reviews, we compute the votes percentage of every vote type in every review and we say that a review is *useful* if the highest percentage of votes was for 'useful' and a review as *cool* if the highest percentage of votes was for 'cool' and a review is *funny* if the highest percentage of votes was for 'funny'.

It is usually common to find a review which combines at least two of those labels, for example users can find a review funny and cool and helpful at the same time especially because they are able to rate one review by more than one type, which will make it harder to decide what label best describes this review. In our dataset, the only case where the three classes have the same rating percentage is when those percentages are zero and this kind of review is excluded from the classification. Additionally, 12% of the reviews, in the dataset, have equal rating percentages for only two from the three class labels. For example useful and funny rating percentages happen to be equal for some reviews in our dataset but this only happened when the percentages are zero and, in this case, the label will be cool for those reviews.

Labeling the reviews as explained above, the data is presented as a supervised training set and unseen reviews with the absence of votes can be classified too. The supervised learning techniques used in this study return a confidence score for every classification instance which gives an idea of how useful, cool or funny a review is. This way, we can use the review classifier to order reviews based on confidence score and recommend ones with the highest score values.

4. Evaluation

4.1. Feature Selection

It is interesting to figure out what are the main factors that make reviews different than other, for example what makes some reviews more useful than others. Feature selection is the process that produces a subset of features sufficient to achieve good classification performance and, at the same time, ranks features based on their importance. In this paper, we will be mainly using the following three different feature selection techniques:

Information Gain (IG), which is one of the most popular *Filter approaches* that attempt to remove irrelevant features from the initial feature's list before implementing the learning algorithm, is the amount of information a feature brings to the training set which is measured by the reduction of entropy caused by dividing the training set using this feature and computed as follows: (Cord and Cunningham, 2008)

$$\mathrm{IG(D, c, f)} = \mathrm{Entropy}(D, c) - \sum_{v \in \mathrm{values(f)}} \frac{|D_v|}{D} \mathrm{Entropy}(D_v, c) \tag{1}$$

While IG feature selection technique is considered effective, this strategy considers features in isolation and ignores the relationship between features. In other words, two features with high IG values may be considered even if there is a strong correlation between them. Additionally, the filter criterion for IG is separate from the induction algorithm used for classification. To address those issues, we introduce the remaining feature selection techniques.

Greedy Backward Elimination (BE) , which is one

of the *Wrapper approaches*, starts with selecting all the features, then considers the options with one feature deleted and then selects the best of these and continues to eliminate features until reaching a certain threshold[2] (Cord and Cunningham, 2008).

Recursive SVM, which implements the recursive feature elimination procedure for linear support vector machines (SVM), is a feature selection method where, starting with the full feature set, attributes are ranked according to the weights they get in a linear SVM. Subsequently, a percentage of the worst ranked features are eliminated and the SVM is retrained with the left-over attributes. The process is repeated until a K number of features is retained (Abeel et al., 2009).

4.2. Classifier Selection

The most important points we need to consider when choosing a classifier is the accuracy and the ability to rank reviews in terms of how well the instances are classified. The probabilities or confidence scores will be used to recommend the most useful, most funny and most cool reviews in the system. The classifiers we have chosen for detailed analysis are *Naïve Bayes* and the *Random forests* because those classification techniques perform relatively well in ranking classifications for recommendation (Zhang and Su, 2004; O'Mahony and Smyth, 2009). Briefly, we explain every classification technique as follows:

4.2.1. Naïve Bayes Classifier

Naïve Bayes is one of the most efficient and effective classification algorithms. Let's assume we have a training set with A_1, A_2,..., A_n where n is the total number of attributes. An example E from this set is a vector of $a_1, a_2, ..., a_n$ where a_i is the value of A_i. Let's present the class label as C. A naïve Bayesian classifer is defined as: (Zhang and Su, 2004)

$$C_{nb}(E) = \arg\max_{c} p(c) \prod_{i=1}^{n} p(a_i|c) \qquad (2)$$

Where the probability $p(a_i|c)$ can be estimated from the training examples. Even though Naïve Bayes is based on the assumption that the features are conditionally independent which is rarely true in reality, it is found to work well with classification problems.

4.2.2. Random Forest Classifier

Random Forest (Zhang and Su, 2004) is a collection of K classifiers $h_1(x), h_2(x), .., h_K(x)$. Each of these classifiers votes for one class and every instance is classified base on the majority class. Every instance of the n training set instances is drawn at random and some instances are nor used in building each tree. Those instances are useful in the internal estimation of the length and correlation of the forest. Random forests are computationally effective and offer good prediction performance. They are proven not to overfit, and are robust to noise and offer possibilities for explanation and visualization of its output. (Zhang and Su, 2004)

It is straightforward to assign a confidence to a prediction produced by naïve Bayes because the classifier directly calculates a posterior probability. Random forest produces confidence scores based on the distribution of training instances classified by the rule or leaf node which is called the class distribution (OMahony et al., 2010).

5. Experiments and Results

Having a rich dataset, like Yelp, gives the opportunity to build approaches with good coverage. In our project, the feature selection by the three different techniques and the classification performance using the two classifier algorithms were trained using a randomly selected 80% of dataset instances. The remaining 20 % instances were used to test the most accurate combination of classifier and feature subset and produce the best recommendation results. Additionally, to evaluate every classifier, we report the accuracy achieved using 10-fold cross validation for every classifier. The accuracy value represents the percentage of correctly classified instances among all instances in the dataset.

Regarding the classification algorithms implementation, we used the open source JAVA-ML 0.1.7 library (Abeel et al., 2009) which contains a collection of machine learning and data mining algorithms that aim to be a readily usable and easily extensible API.

The platform used for "Shopping" category feature construction and classification experiments were implemented in Java and compiled on a Windows based PC with Intel Core i5-4300 CPU having a speed of 1.90GHz and 8GB of RAM. Because "Restaurants" category experiments require more computational power (the dataset size is doubled), they were run on the University of Waterloo central CS server environment 'linux.cs.uwaterloo.ca'.

5.1. Feature Selection Results

To consider the relative importance of individual features, we picked the top 9 features from information gain (IG), Recursive SVM and Greedy BE feature sub-

[2] http://oucsace.cs.ohiou.edu/ razvan/courses/ml6900/lecture04b.pdf

sets. Table 2 and 3 present the results of feature selection algorithms on "Shopping" and "Restaurants" categories respectively. The output of IG and Recursive SVM produces a feature and a corresponding ranking and the features presented in the tables are rank-ordered. On the other hand, Greedy BE, outputs a subset of features with specific size, ie, when we have 1 feature, the Greedy BE will output the most important feature and when we choose 2 features, the algorithm picks the most important feature and the second most important feature and so on.

Table 2. Feature Selection on "Shopping" reviews

Features	IG	Recursive SVM	Greedy BE
1	R5	R1	U1
2	U1	R3	U13
3	U3	U11	U4
4	U4	B3	ST4
5	U5	U8	U6
6	U13	U4	ST5
7	U2	B5	U3
8	U8	U6	ST6
9	ST1	U3	U2

Table 3. Feature Selection on "Restaurants" reviews

Features	IG	Recursive SVM	Greedy BE
1	U3	U4	U5
2	U1	R5	U3
3	U5	B2	U1
4	U2	U7	U6
5	U6	U3	U2
6	U4	U12	U4
7	ST3	U13	U13
8	U8	R3	ST2
9	ST2	B5	U8

From the tables 2 and 3 we can notice that each feature selection algorithm picked different feature subsets. More specifically, Greedy BE ranked user features as the most important ones. Similarly, IG considered features related to the user, such as user's average helpfulness, funniness and coolness, as the most important features in addition to some structural review features, such as the percentage of uppercase characters and the ratio of tags. On the other hand, Recursive SVM considered features related to the review readability, such as the ARI score and the Fog Index, and business features , such as the optional number of business attributes and mean business rating, as significant features in addition to user features.

None of the features selected from IG and Greedy BE is related to business which is a surprising result, given that we expected more informative business reviews to be more helpful in determining the review type. It may be that cool and funny reviews are hard to relate with business information. Furthermore, as expected, all the feature selection techniques agreed that user's reviewing behaviour is a strong predictor of whether the review is useful, cool or funny. Different from the other techniques, Greedy BE considered three structural review features (number of words, complex words and sentences) as a good indicator of whether easy or hard structured reviews reflect the review usefulness.

In the next section, we examine the classification performance when review instances were constructed using all features and the top 9 features ranked from the three feature selection technique to be able to decide which option produces the most accurate results.

5.2. Classification Results

In the following section, the classification evaluation was performed using the accuracy value which represents the percentage of correctly classified instances over the total number of possible instances in the dataset. Additionally, we report every classifier accuracy using 10-fold cross validation for this classifier. For the Random Forest classifier, to choose the best number of trees T, 10-fold cross validation was performed on experiments with different T values. Results showed that T=14 gave the best accuracy of Shopping category (93%) and T=9 had the best accuracy (99%) for Restaurants category.

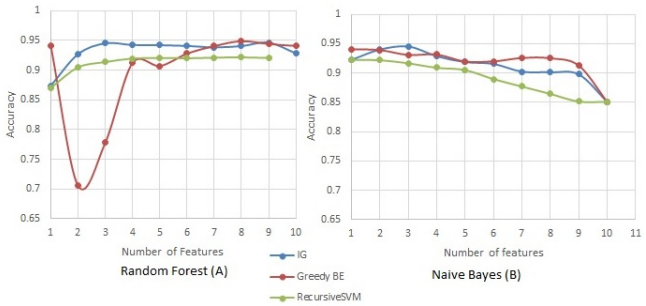
5.2.1. Classification by feature category

Let us begin by looking at the classification performance when different subset of features are used for classification. Picking the top 9 features from every feature selection technique presented in the previous section, we start with the highest scoring feature, evaluate using 10-fold cross-validation the performance of a classifier built with that feature, then add the next highest ranking feature and evaluate again. We repeat until no further improvements are achieved.

Figure 3 represents the performance of Naïve Bayes (B) and Random Forest (A) classifiers in "Shopping" dataset using subsets from the IG, Greedy BE and Recursive SVM features selection techniques. Starting with Random Forest, Greedy BE was very sensitive to the number of subset features compared to IG and Recursive SVM. Additionally, the best accuracy was achieved using 8 features from the Greedy BE technique and IG technique (0.9487 and 0.9451 respectively). Moving to the Naïve Bayes classifier, the best accuracy was achieved using IG using 3 features

with a value of 0.9453.

Figure 3. Classification by feature category for "Shopping"



Moving to "Restaurants" Category, figure 4 shows that, for Random forest classifier, the best accuracy was achieved using 6 Recursive SVM features with an accuracy of 0.9953. For the Naïve Bayes, the best accuracy achieved was 0.9953 using 2 features for all the three feature selection techniques.

Figure 4. Classification by feature category for "Restaurants"

As soon as the Random Forest classifier achieves its best accuracy with a specific feature subset, the classifier starts to be insensitive to the number of features we add. The main reason for this behaviour is that Random Forest is an embedded approach that applies the feature selection process as an integral part of the learning algorithm so irrelevant features are not considered even if we increase the number of features. On the other hand, Naïve Bayes classifier gets worse as the feature subset size increases which is not surprising as we are incorporating irrelevant features that badly effects the classification task.

"Restaurants" classification achieved very high accuracy percentage (99.53%) which is good compared to a flip of a coin classification approach (33%). However, if we remember the class label distribution, the dataset had 99% of its reviews as 'useful'. Having such high distribution for only one class makes the classifier results less reliable. This problem is known as 'Rare Class Classification' and we will be discussing it and suggesting solutions in section 7.

5.2.2. Classification by all features

In this section, we examine the classification performance when all the 32 extracted features are used in the classification approach. Table 4 shows that the lowest accuracy is achieved when using all features with Naïve Bayes classification approach. When only relevant features are used, Naïve Bayes achieved a higher accuracy (94.54%). The best classification based approach is Random Forest with the 8^{th} Greedy BE features subset with an accuracy value of 94.87%.

Table 4. Classification by All/Subset of features,"Shopping"

	All features	Best Subset of features
Naïve Bayes	85.10%	94.54% (3 features)
Random Forest	92.83%	94.87% (8 features)

The distribution of labels in "Shopping" dataset (88% of the reviews are 'useful' and 12% are for 'cool' and 'funny'), makes the good accuracy value achieved by the classifier less reliable. To better interpret the classifier behaviour, the confusion matrix of the three different labels is presented in table 5.

Table 5. Confusion Matrix for "Shopping" labels

	Predicted Useful	
Actual Useful	25123	550
	1559	1814

	Predicted Funny	
Actual Funny	914	760
	214	27131

	Predicted Cool	
Actual Cool	790	909
	419	26928

From the 'useful' confusion matrix in 5, we observe that 98% of 'useful' reviews in the training set are correctly classified and 46% of the 'funny' and 'cool' reviews are miss-classified as 'useful'. Particularly, 45% of 'funny' reviews are incorrectly classified and 53% of 'cool' reviews are incorrectly classified. Furthermore, the F-score of 'funny' and 'cool' classes were 0.6470 and 0.5475 respectively while 'useful"s F-score was 0.96041.

Even though the average accuracy of the classification approach is high, the classifier could not perform well for all the three classes, mainly, because two of those classes have very low distribution numbers across the training set. In section 7, we suggest solutions to over-

come such 'Rare Class Classification' problem.

5.3. Recommendation Evaluation

Using the reasonable classification performance achieved from the previous sections, we can be optimistic that this classification approach provides a basis for high quality recommendation. The classifier results can suggest the most 'useful', 'funny' and 'cool' reviews navigating in Yelp yelp website by raking those reviews using the class distribution probabilities of every instance. We randomly selected our test set and we choose only the 8^{th} Greedy BE subset for the Random Forest classification approach.

To evaluate the recommendation performance, we consider how frequently the system manages to select 'useful', 'funny' or 'cool' review according to the definition given in section 3.3. Using this approach, 95.35% of the reviews were correctly labeled compared to only 35% of a random approach for labeling. Our approach achieved much more improved results compared to randomly selecting the most "useful", "cool" and "funny" reviews.

6. Conclusion

In this paper, we have presented a feature selection followed by a classification-based approach to the recommendation of 'useful', 'funny' and 'cool' reviews in Yelp site. We have considered various feature selection techniques and examined their performance in terms of accuracy and the number of features included in the classification approach. The learning task proved that Random Forest was more robust to the presence of noisy features, while Naïve Bayes achieved best accuracy when only considering top ranked features.

User features, such as the user average helpfulness votes and the percentage of useful/funny and cool reviews the user writes, and structural features ,such as the number of words/complex words and sentences, proved to be most useful in terms of classification performance. Business features were less successful. Such results give us an insight of what makes reviews 'useful', 'funny' or 'cool' in Yelp.com.

7. Future Work

Although the classification based approach proved to achieve high accuracy results, we believe that the training set distribution for both "Shopping" and "Restaurants" categories had a high influence on the accuracy. Further analysis is required to solve such "Rare Class Classification" problem either by doing 'Undersampling' where we remove instances from major classes but we end up loosing a large amount of training instances or by 'Oversampling' which is a way to create duplicated instances for rare classes or by using non-state-of-the-art classification approaches modified to deal with such dataset. The work presented in (He and Ghodsi, 2010) is an example of a modified version of SVM that deals with highly imbalanced datasets that proved to consistently outperforms regular SVM and the two re-sampling methods.

The labeling strategy suggested in this paper (section 3.3) is highly correlated with the dataset characteristics. More specifically, in our dataset, it is very rare for the average coolness and funniness and usefulness to be the same in one review but this is not the case in other real life systems. If this characteristic is missing in the dataset, the current strategy may miss-label many reviews and this will badly effect the overall classification performance. More robust and sophisticated labeling strategies need to be investigated to come up with the most accurate and less sensitive to the data characteristics strategy.

One of the most important questions asked in user-generated content is how helpful or unhelpful a review is. To answer this question, we need to build a classifier that classifies reviews as either 'helpful' or 'unhelpful'. We tried to answer this question using the provided Yelp dataset at the beginning of this project. The state of the art Naïve Bayes classifier trained on "Shopping" reviews achieved an accuracy of 54% which is very close to a flip of a coin chance. The classifier's accuracy was quite poor because of the way reviews were labeled. Better labeling could be achieved if, instead of picking 'useful', 'funny' or 'cool' votes, Yelp users were able to pick either 'helpful' or 'unhelpful' vote for rating the helpfulness of the review. This strategy will help Yelp site suggest the most accurate helpful reviews from the unhelpful ones.

Furthermore, to come up with the best classification approach, we used engineered features (section 3.2). Another approach would be using the words in the review text to perform automatic feature extraction such as n-grams. At this point, we can not claim that the engineered features are better than automatically extracted features; only experiments can prove so. Additionally, instead of using state of the art feature extraction techniques, we can try using improved ones such as the Greedy Column Subset Selection which is a fast and accurate Map-Reduce greedy algorithm that minimizes an objective function by measuring the reconstruction error of the data matrix based on the subset of selected columns. (Farahat et al., 2013)

References

Thomas Abeel, Yves Van de Peer, and Yvan Saeys. Java-ml: A machine learning library. *The Journal of Machine Learning Research*, 10:931–934, 2009.

Matthieu Cord and Pádraig Cunningham. *Machine learning techniques for multimedia: case studies on organization and retrieval*. Springer, 2008.

William H DuBay. The principles of readability. *Online Submission*, 2004.

Mingming Fan and Maryam Khademi. Predicting a business star in yelp from its reviews text alone. *arXiv preprint arXiv:1401.0864*, 2014.

Ahmed K Farahat, Ahmed Elgohary, Ali Ghodsi, and Mohamed S Kamel. Greedy column subset selection for large-scale data sets. *arXiv preprint arXiv:1312.6838*, 2013.

Peter Hajas, Louis Gutierrez, and Mukkai S Krishnamoorthy. Analysis of yelp reviews. *arXiv preprint arXiv:1407.1443*, 2014.

He He and Ali Ghodsi. Rare class classification by support vector machine. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 548–551. IEEE, 2010.

Michael P O'Mahony and Barry Smyth. Learning to recommend helpful hotel reviews. In *Proceedings of the third ACM conference on Recommender systems*, pages 305–308. ACM, 2009.

Michael P O'Mahony and Barry Smyth. Using readability tests to predict helpful product reviews. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, pages 164–167. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2010.

Michael P OMahony, Pádraig Cunningham, and Barry Smyth. An assessment of machine learning techniques for review recommendation. In *Artificial Intelligence and Cognitive Science*, pages 241–250. Springer, 2010.

Yelp. Yelp dataset challenge. http://www.yelp.ca/dataset_challenge/, 2004-2014.

Harry Zhang and Jiang Su. Naïve bayesian classifiers for ranking. In *Machine Learning: ECML 2004*, pages 501–512. Springer, 2004.