Finding Common Opinions in User-Generated Reviews

Brett Miller, *brettm@cs.stanford.edu*

A large and growing body of user-generated reviews is available on the Internet, from product reviews at sites like Amazon.com to restaurant reviews at sites like Yelp.com. For users making a purchasing or dining decision, the opinions of others can be an important factor. Although some aggregate information -- like average-star ratings -- for multiple reviews is sometimes available, in general the only way to determine common views among users is by reading through many reviews. As the number of reviews for a single product or restaurant becomes large (on the order of hundreds), it becomes increasingly impractical to read every review. Some techniques are commonly employed to compensate for this, such as ranking reviews by usefulness, as determined by readers. Since readers are most likely to read only the top-ranked reviews, however, this approach likely leads to a reinforcement of existing useful reviews, while relegating new, unread reviews to the bottom of the list.

A more sophisticated approach, and the focus of this paper, is to apply machine-learning techniques to the problem. The goal of reading multiple reviews is viewed to be determining the most common specific opinions of reviewers, and informally, we wish to let a machine exhaustively read every review for a product or restaurant and automatically find cohesive groups of opinions that are both closely related and widespread. Formally, we can break the problem into two concrete machine-learning tasks: (1) apply supervised-learning techniques to classify each sentence in every review as either *opinion* or *non-opinion* and (2) for all sentences classified as opinions, apply unsupervised learning techniques to cluster those that are closely related.

The remainder of this paper describes the system proposed to achieve this goal. Following a description of the corpus used to test the system, implementation details, test results, a discussion of their meaning, and conclusions will be given.

Corpus

For the purposes of this project, the corpus consisted of diner-generated restaurant reviews available at Yelp.com. Reviews were collected exhaustively for 6 restaurants. Reviewer names were discarded. Each review consists of a star rating (1 to 5) and the body of the review. Statistics are shown below in Figure 1.

Figure 1. Corpus Statistics

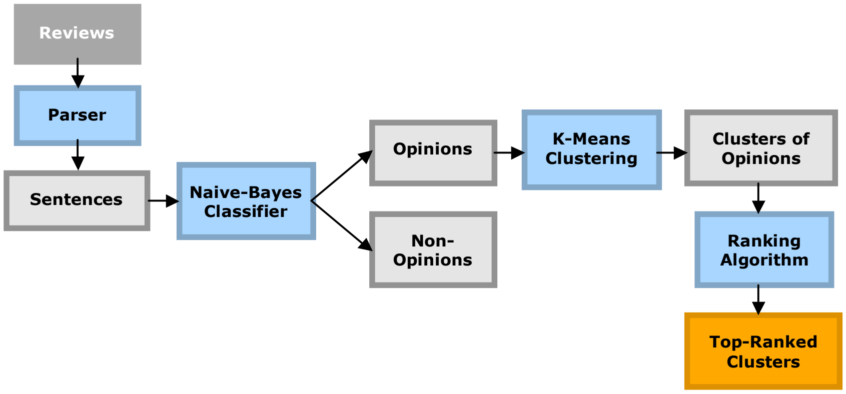
Restaurant	Total Reviews	Total Sentences	Length Range (words)	Average Length (words)
Coi	139	2556	6-804	240
Cortez	326	3920	2-810	154
Evvia	332	3777	2-832	137
Pesce	243	2320	7-825	117
Plouf	343	3406	1-693	122
Tamarine	348	3802	5-856	130

Training data for classification was obtained by sampling without replacement 1210 sentences (approximately 6%) from the corpus and manually labeling each as *opinion* or *non-opinion*.

Implementation

In order to achieve the objective of finding common opinions across multiple reviewers, the processing pipeline of Figure 2 is proposed. Each major stage of processing including sentence parsing, Naive-Bayes classification, K-means opinion clustering, and cluster ranking, is described in more detail below.

Figure 2. Processing Pipeline



Review Parsing

A sentence is considered to be a sequence of one or more words delimited by a period, exclamation point, or question mark. Sentences are tokenized on white-space after case-folding and removal of non alphanumeric characters.

Opinion Classification

The first stage of the processing pipeline requires classification of sentences into either the *opinion* or *non-opinion* class. Because it is relatively simple and often competitive with more complex classifiers for text applications, a Multinomial Naive-Bayes classifier with Laplace smoothing was chosen for implementation. The input feature for each sentence is an N-dimensional term vector, with N equal to the size of the dictionary over the entire training set. Thus, element i in an input vector contains the raw term count for term t_i in the dictionary. The training and testing algorithms are implemented as detailed in the lecture notes and they will not be repeated here.

Also, in an attempt to compensate for our skewed training data (only 34% of examples are labeled *opinion*), an alternative Complement Naive-Bayes (CNB) classifier, as described by Rennie et. al [1], has also been implemented.

Opinion Clustering and Ranking

The next stage of processing requires finding common opinions among the set of opinions output by the classifier. Again, for simplicity and because it often produces good results, an implementation of the K-means clustering algorithm was chosen. As usual, input is the set of m opinion vectors, where each is normalized and of dimension equal to the size of the dictionary over all reviews from a single reataurant:

$$\left\{o^{(1)},\ldots,o^{(m)}\right\}\text{, where un-normalized } o_j^{(i)} = \text{occurrences of term } t_j \text{ in opinion } i$$

The iterative algorithm to produce K clusters from the m input vectors is implemented exactly as detailed in the lecture notes, and only the relevant notation is reproduced here, for clarity:

$$c^{(i)} = \text{cluster assignment for opinion vector } o^{(i)}$$
$$\mu_j = \text{cluster centroid } j$$

Because we do not know, *a priori*, the optimal value of K, we need a way to evaluate the quality of the clusters produced by the algorithm for a given value of K, and then choose K such that the quality of the clusters is maximized over some reasonable range of K. For this specific application a cluster of high quality is considered to be one that contains many opinions, tightly grouped around the centroid. Formally, the quality of cluster j is measured as the product of inverse residual sum of squares (or RSS, a standard measure of internal cluster quality [2]) and cluster cardinality:

$$Q(j) = \frac{\left|\left\{o^{(i)} \mid c^{(i)} = j\right\}\right|}{\sum_{i:c^{(i)}=j} \left\|o^{(i)} - \mu_j\right\|^2}$$

Thus, large clusters of unrelated opinions have low quality, as do very small clusters of closely related opinions. Having obtained cluster assignments for the optimal value of K, each cluster j is then ranked on the basis of its quality $Q(j)$.

Results

First, results for the Naive-Bayes classifier are presented. Ten-fold cross-validation test-set and training-set error as a function of the number of training examples, m, is shown in Figure 3. Note that the alternative CNB classifier had nearly identical performance, so the results are omitted.

It's also informative to look at the top 10 words with the highest predictive value for the *opinion* class:

```
excellent, loved, those, atmosphere, yum, best, setting, slightly, leaving, client
```

There are two important means of evaluating the results of the clustering and ranking algorithms. First, we can observe how the average quality measure changes as a function of K. These results are shown in Figure 4. Second, we can manually inspect the top clusters (after ranking) for optimum K produced by the algorithm to assess how well they represent common opinions. We save this for the discussion.

Figure 3. 10-Fold Hold-Out Cross-Validation Error vs. Training Set Size

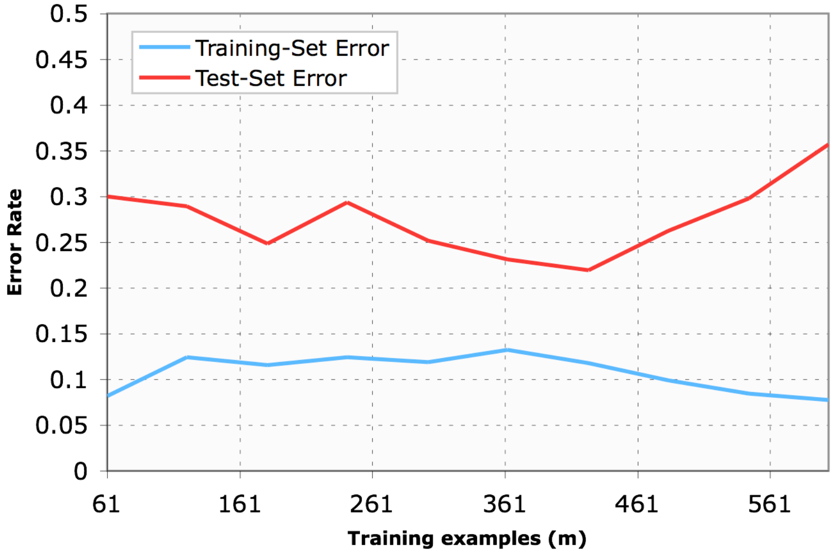
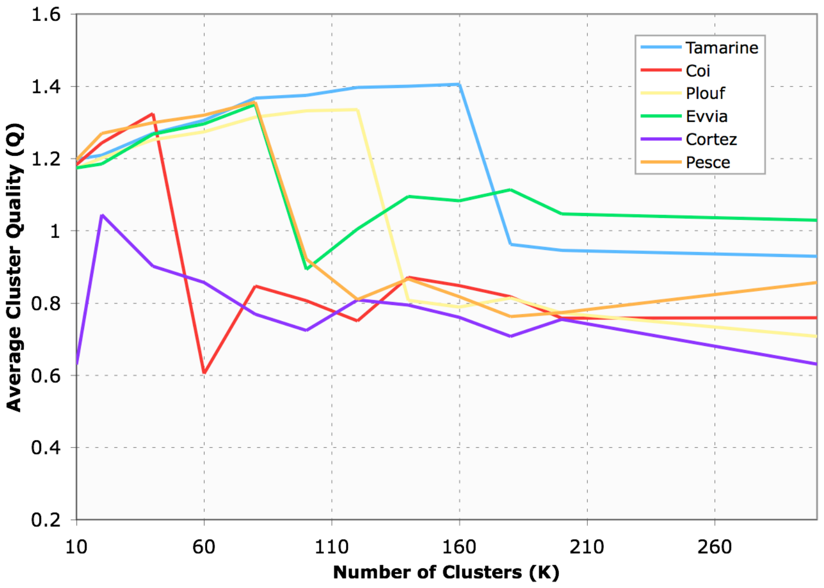


Figure 4. Average Cluster Quality (Q) vs. Number of Clusters (K)



Discussion

As Figure 3 makes clear, classifier performance is poor. Even in the best case, 10-fold cross-validation test-set error is 21.9%, while training-set error is around 11.8%. Because both forms of error are relatively high, even for increasing training-set size, the classifier likely suffers from high bias, probably due to our choice of term frequencies alone as features. There are several important observations to make about the nature of the classification problem that might help explain the poor performance. First, unlike related sentiment-classification problems [3], we are trying to classify short sentences, consisting of few words, so there is very little evidence for the sentence belonging to either class. Second, it was probably unrealistic to expect opinion to be highly correlated with word features alone. There are clearly cases where strong words like *excellent*, *loved*, and *yum* (from our top-10 list) easily predict opinion, but in general the expression of opinion is quite nuanced, and the true sentiment of a reviewer might be implied rather than explicit. Even hand-labeling the training examples was sometimes difficult, as the distinction between opinion and non-opinion was unclear. Taking into account these and other challenges, such as the presence of sarcasm, it seems that a more sophisticated set of features is needed for good classification.

In contrast to the difficulty of classifying opinion, the results of K-means clustering are very promising. First, looking at Figure 4, we see that for all restaurants, there exists a clear peak in average cluster quality as a function of K. Additionally, the shape of the plots makes sense, with quality initially increasing as the model more closely fits true clusters in the data and then sharply falling off as K becomes too large and good clusters start to fragment.

Furthermore, when we examine top-ranked clusters for each restaurant, many seem to be useful in that they indeed contain closely-related opinions that are expressed by many reviewers, although there are frequently non-opinion sentences in the clusters as well, due to the poor performance of the classifier. Several examples of top-ranked clusters are provided in Figure 5.

Figure 5. Examples of High-Ranking Clusters

Highly-ranked cluster of 21 opinions on sea-bass dish from restaurant Evvia (K=80)	**Highly-ranked cluster of 8 opinions on mahi-mahi dish from restaurant Plouf (K=120)**
- Sea bass - Very light but had the right amount of flavor - We ordered the striped sea bass, the moussaka, and the lamb chops for our entrees - The striped sea bass was served on a bed of wilted greens and was delicious - The Lavraki Psito (sea bass) is also a great entree if you're looking for seafood - The sea bass was fresh and light in flavor, allowing the natural qualities of the fish to shine - The sea bass I ordered was simply grilled and dressed with lemon juice and oregano - The Sea bass is likewise excellent ... (15 more)	- The Mahi Mahi appetizer was great - I ordered the Mahi-Mahi atop cranberry fusion cous-cous and grilled bok choy - The mahi mahi was drizzled with crushed olives, which was a bit overpowering in taste - Mahi Mahi with Five-Spiced Couscous, Baby Bok Choy and Cranberry-Onion Compote - However, the mussels (which IS their specialty) and mahi mahi was delicious - I could just sit there all day munching on those mussels, and the mahi mahi was soo moist - Afterwards I ate their mahi mahi - The mahi mahi seemed undercooked

There are also many examples of high-ranking clusters that meet all of our criteria for quality but are probably less useful because the words they have in common are descriptive, but not specific. For example, a cluster might form with opinions that all use the adjective *excellent* but describe different aspects of a restaurant.

Conclusions

To address the growing body of user-generated reviews available on the Internet, a system to automatically extract clusters of widespread, common opinions using a combination of supervised and unsupervised learning techniques has been proposed. A simple application of a Naive-Bayes classifier using words as features performs poorly at classifying short sentences as either *opinion* or *non-opinion*, likely suffering from high bias. This is probably due to the fact that classification is at sentence-level granularity and that words alone are insufficient as features. However, K-means clustering, coupled with an application-specific quality measure both for finding optimum K and ranking the resulting clusters, performs well. Many top-ranked clusters meet the subjective criteria initially proposed.

References

[1] Rennie, Jason D. M.; Shih, Lawrence; Teevan, Jaime; and Karger, David R. “Tackling the Poor Assumptions of Naive Bayes Text Classifiers.” Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington D.C., 2003.

[2] Manning, Christopher D.; Raghavan, Prabhakar; and Schutze, Hinrich. Introduction to Information Retrieval. Cambridge: Cambridge University Press, 2008.

[3] Popescu, Ana-Maria and Etzioni, Oren. “Extracting Product Features and Opinions from Reviews.” 2005.