User Review Sentiment Classification and Aggregation

Steven Garcia (garcias@stanford.edu) and Ping Yin (pingyin@stanford.edu)

INTRODUCTION

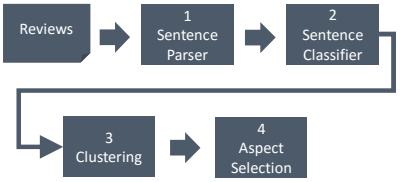
User reviews on product websites such as Yelp or Amazon provide a wealth of information about the subject of the review. In many cases when selecting a service (i.e. restaurant, doctor, etc.) or when choosing a product (i.e. book, album, dishwasher etc.) the amount of information available in user reviews can be of greater volume and more trustworthy than the official product descriptions provided by the vendor.

However the number of reviews for a single item can at times be overwhelming. For instance, the Amazon Kindle keyboard 6" e-reader has over 42,000 user reviews today. The review reader can gain an understanding of overall satisfaction by looking at summary statistics such as the average rating or score, but the details about what make each product great (or terrible) are hidden inside the body of the reviews themselves.

We propose an approach to summarize reviews that leverages trained models for classification and clustering to derive sentiment and the key aspects of a reviewed item. Given a set of item reviews, our system will output snippets from the review set that are representative of key aspects that users had strong positive and negative opinions about.

ARCHITECTURE

Our system is structured is as follows:



1. Sentence parser utilizing NLTK to split reviews into sentences. Each sentence is assigned a proxy label based on the overall review score.
2. The sentence classifier is a trained classification model that assigns a positive or negative label to an input sentence. The classifier is trained using only highly positive and negative labels (1 or 5 stars).
3. Having classified a set of review sentences, we cluster predicted positive and negative sentences using K-means clustering. Where each cluster represents a common positive (or negative) aspect of the item review set.
4. Finally we rank the clusters and select representative sentences from the highest ranked groupings.

METHODS

Here we list noteworthy decisions that allowed for improved performance:

Sentence classifier:

- Applying a threshold on the classifier to achieve target precision and discarding sentences with low confidence from the classifier.
- Model sweep to select model parameters with lower test error.
- Stemming terms to reduce the number of feature dimensions and therefore allowing the model to better leverage the features.

Clustering:

- Feature reduction using TF and IDF to reduce noisy clustering.
- Dynamic selection of K based on review set properties.

Aspect selection:

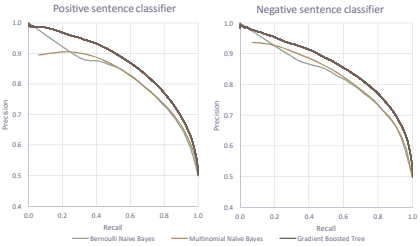
- Identifying the common features in a cluster to assist cluster ranking and final aspect sentence selection.

Further details can be found in the full report.

RESULTS

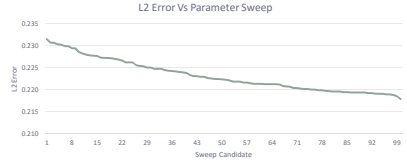
Classification

Comparison of classification algorithms for positive and negative sentence classifier:



The gradient boosted tree classifier outperformed the Naïve Bayesian classifiers significantly due to the model's ability to better capture the relationships between co-occurring terms in the data.

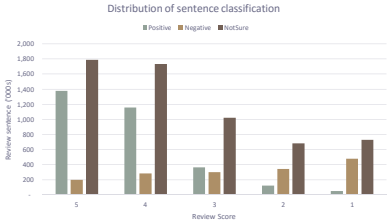
Gradient boosted trees have a large number of parameters. To get the best performance out of the model we run a sweep across the parameter space selecting 100 possible parameter sets and choose the best performing model.



To ensure accuracy for our final sentiments we apply a threshold to the classifier targeting 90% precision. This results in classifiers with performance of:

Classifier	Threshold	Precision	Recall
Positive	0.77723	90%	51%
Negative	0.18952	90%	45%

When we apply the classifier to the rest of the data set the results are broken down as follows:



As anticipated, sentences from positive reviews are more likely to be classified positively and visa-versa.

Clustering

Review sets differ in length and accordingly in the number of aspects. We devise a unique approach to the selection of k for our k-means clustering model.

First we define a **valid cluster** as one with any single feature that appears in more than half of the cluster elements, and where the cluster has more than ten sentences (other parameters explored in report).

The K selection algorithm is follows:

```
Let k = 10; p = 0; step = 20
while true:
  run K-means with parameter k
  c := count valid clusters
  if p >= c or c = 0:
    break
  k := k + step
end while
```

To quantify the validity of this approach we manually labelled detected clusters as *good* or *bad*. Below we present results for two sample review sets using this approach and highlight the point of convergence (further details in report):

	Example 1		Example 2	
K	C	good/bad	C	good/bad
20	4	3/1	5	3/2
40	9	8/1	6	4/2
60	15	14/1	6	4/2
80	13	10/3	3	1/3
100	15	15/12	4	¼

End-to-end example:

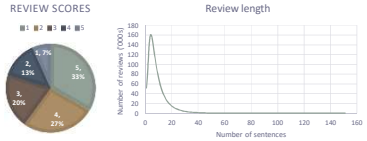
Finally we present a sample of the results. The candidate business is Hugo's Cellar in Las Vegas, and below are sampled positive aspects with the principle feature highlighted (negative aspects found in report):

- "The long stem **rose** for the ladies is a nice touch."
- "The best part of the dinner are the complimentary **chocolate** covered strawberries and dried fruit…"
- "Truly one of the most **romantic** restaurants in the world."

DATA

We utilize the Yelp Challenge Dataset containing 1.6M reviews spanning 61K businesses and 366K users.

Data distribution:



We also present additional findings using the Amazon Reviews dataset in the full report.

CONCLUSIONS

We have developed a system to process a large number of reviews and report sentences from the reviews that are representative of the best and worst aspects.

We found that with only review level labels, we could adequately train a classifier to predict positive and negative sentences.

We developed an algorithm to select the number of clusters required for each review set by leveraging a heuristic based on feature representation in the clusters.

Finally we show that that our system correctly identifies key aspects with end-to-end examples.

FUTURE WORK

- The classifier was trained on noisy data (not every sentence in a positive review is positive). Techniques to filter the labels before training would result in lower error at the classification phase and less data thrown away with an unconfident prediction.
- Bag of words features provide limited information for the models. Related work has shown that POS tagging and parse trees can provide a wealth of information to improve the features for learned models. Specifically this can help target aspects to less general concepts.
- Due to the data set selection we were unable to measure aspect identification recall. The success of any system in this field depends on its ability to adequately identify aspects of a review set and we intend to measure this dimension in next.