Machine Learning for Sentiment Analysis on the Experience Project

Raymond Hsu
Computer Science Dept.
Stanford University
hsuray@cs.stanford.edu

Bozhi See
Electrical Engineering Dept.
Stanford University
bozhi@stanford.edu

Alan Wu
Electrical Engineering Dept.
Stanford University
alanw@stanford.edu

Abstract

The goal of sentiment analysis is to extract human emotions from text. This paper applies various machine learning algorithms to predict reader reaction to excerpts from the Experience Project. Metrics such as accuracy of prediction and precision/recall are presented to gauge the success of these different algorithms. We propose a system to process the documents and to predict human reactions, as well as provide results. We discuss various methods and their advantages and disadvantages in sentiment analysis for these documents. Finally, we comment on applying our findings to sentiment analysis in a more general sense.

1 Introduction

One application of machine learning is in sentiment analysis. In this field, computer programs attempt to predict the emotional content or opinions of a collection of articles. This becomes useful for organizing data, such as finding positive and negative reviews while diminishing the need for human effort to classify the information.

1.1 Related Work

Much literature in the field of sentiment analysis have focused on different classification models for text. Previous approaches include hand-coded rules (Neviarouskaya et al., 2010), the winnow algorithm (Alm et al., 2005), random k-label sets (Bhowmick et al., 2009), Support Vector Machines (SVM) (Koppel and Schler, 2006), and Naive Bayes (Mihalcea and Liu, 2006). However, previous work has done classification only on three or fewer categories - typically positive, neutral, and negative. Our work attempts to extend this by inferring specific emotional reactions rather than broad categories.

1.2 Problem

We will perform sentiment analysis on confessions from the Experience Project[1] (EP), a collection of short, user-submitted posts reflecting the writers' thoughts and actions. EP further allows other people to express reactions to these pieces by voting on five predefined categories, thus providing labeled data of readers' reactions for use in a classifier. The five numbered categories in EP (along with our descriptions) are:

1. **Sorry, Hugs:** Offering condolences to the author.
2. **You Rock:** Positive reaction indicating approval and offering congratulations.
3. **Teehee:** Reader found the anecdote amusing or humorous.
4. **I Understand:** Show of empathy towards the author.
5. **Wow, Just Wow:** Expression of surprise or amazement.

We consider two tasks: In the first task, for a given confession, we predict which label will receive the most votes (the "max label" task). This is similar to traditional multi-class classification tasks, with the exception that our ground truth labels are only partially correct. It is likely that the same confession can elicit different emotions from different people, so we need to take into account that confessions are not something that can necessarily be neatly partitioned into disjoint categories. Hence we have a second task where we predict which labels will receive at least one vote (the "label presence" task). The label presence task tries to answer the question, "what are all the emotions that readers feel after reading this confession?".

Useful applications of our findings include sentiment detection and classification in social networking sites, where these kinds of text often appear. Many of the confessions appearing in EP are similar in style to status updates on popular platforms. Our findings on feature selection may be used to guide sentiment analysis for these social networks in the future.

The paper is organized as follows. In our model, we will describe how we plan on predicting the human reaction to the passages of text. We will describe each stage of our algorithm and its purpose. In the results section, we will look at the improvement in our metrics as each stage is added, for both tasks. In our discussion section, we will analyze the challenges faced as well

[1] http://www.experienceproject.com

as reasons why our techniques improved our prediction accuracy. We also discuss other attempted methods that were less successful in predicting reactions and explain why we think they did not work as well.

2 Model

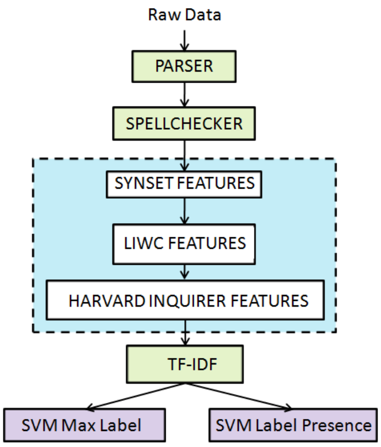

Figure 1: Final model diagram.

Our system consisted of first processing the confessions in order to extract a feature set, before passing the data into a supervised learning algorithm.

2.1 Parser

In order to refine our data and improve the feature set, we removed all HTML tags using a Python parser. This was essential towards refining our dataset because HTML tags do not convey emotions and would skew our feature vector by including phrases that have no semantic meaning (e.g. '&nbsp;').

Emoticons, on the other hand, are an excellent way of conveying emotions through text because it captures the emotion of the writer by including a facial expression. Therefore, we captured this unique feature set and used it to improve our feature vector.

2.2 Spell Checking

One of the issues we encountered in our earliest models was overfitting. On closer inspection of the raw data, we noticed that there were many spelling errors. In order to reduce problems of overfitting as a result of having too many unique spellings, we ran the raw data through a spell checker and corrected all the spelling errors.

2.3 Features

We considered three features in our model: bag of words, WordNet[2] synsets, and sentiment lexicons.

2.3.1 Bag of Words (BoW)

The BoW model is the most basic feature model in sentiment analysis. It treats each unique word token as a separate feature. We use BoW features as our initial feature set for our system. This basic model acted as a test bench for us to observe the changes needed to make to our model better.

2.3.2 WordNet Synsets

In order to further improve the quality of the feature set and decrease overfitting, we used WordNet to map the words in the confessions onto their synonym set (synset). By mapping words into their synset, we made the assumption that the words of similar meaning elicit similar emotions. This reduces the number of unique features we have and also improves the coverage of each feature. This technique also allows us to handle words that do not occur in our training data if they happen to be in the same synset as words that do occur in our training data.

2.3.3 Sentiment Lexicons

Sentiment lexicons are groupings of words into emotion and content categories. We used two of them in our system because we found they improved performance. We used them by replacing the original words with their sentiment lexicon category. The first sentiment lexicon we used was Language Inquiry and Word Count (LIWC) (Pennebaker et al., 2007), a hand-engineered set of words and categories used by psychologists to group words by similar emotional and subject content. We also used features from the Harvard Inquirer (Stone et al., 1966), which also categorizes words by emotional and subject content. Like LIWC, the Harvard Inquirer was also hand-engineered by psychologists for the purpose of analyzing text. Both lexicons have been used in previous work on sentiment analysis.

2.4 TF-IDF

Not surprisingly, function words such as 'and', 'the', 'he', 'she' occur very often across all confessions. Therefore, it makes little sense to put a lot of weight on such words when using bag of words to classify the documents. One common approach is to remove all words found in a list of high frequency stop words. A better approach is to consider each word's Term Frequency-Inverse Document Frequency (TF-IDF) weight. The intuition is that a frequent word that appears in only a few confessions conveys a lot of information, while an infrequent word that appears in many confessions conveys very little in formation. We produce weights for each word via the following equation:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$$idf_i = \log \frac{|D|}{|d_{t_i}|}$$

$$tfidf_{i,j} = tf_{i,j} idf_i$$

- $tf_{i,j}$: importance of term i in document j

[2] http://wordnet.princeton.edu/

- $n_{i,j}$: number of times term i occurred in document j
- $\sum_k n_{k,j}$: total number of words in document j
- idf_i: general importance of term i
- $|D|$: total number of documents in the corpus
- $|d_{t_i}|$: number of documents where the term t_i appears

2.5 SVM

We used Support Vector Machine (SVM) as the final classifier to make predictions for both tasks. For the label presence task we train five SVMs that perform binary-class classification, one for each category. However, this is insufficient for the max label task since it is a multi-class classification task. A common solution is to build multiple one-versus-all SVM classifiers and combine them to perform multi-class classification (Rifkin and Klautau, 2004). For each category, we use the five binary-class SVMs from the label presence task to predict whether a confession belongs to that category or not. We make a prediction in the max label task by running all five binary-class SVMs and choosing the category with the most positive value.

3 Results

We discuss the results of the two tasks separately.

3.1 Max Label Task Results

To evaluate our results on the max label task, we first established a naive baseline for comparison. The baseline is to simply always predict the most popular category (category 4). We then compare the baseline performance across different models. Results are shown in figure 2.

The baseline accuracy is 37% with very low precision and recall. Our most basic model, SVM with BoW features, improves upon the accuracy of the baseline by 3%. Even though the increase in accuracy is small, we see signficiant increases in precision and recall. The increase in precision and recall is because our BoW model makes predictions across all five categories. Thus, we conclude that raw words without any attempt at feature reduction or sentiment labeling are sufficient to give some information about what sort of reaction users will have to that text. However, upon further analysis of the BoW model we find that it overfits the training data, achieving upwards of 90% training accuracy (compared to 40% testing accuracy). The BoW model is therefore not generalizable and we turned to synset features to reduce the overfitting.

Using synsets in place of words further improves the values of all three metrics most notably in the form of an additional 5% increase in accuracy. We also no longer have the signficant overfitting problem encountered under the BoW model. Adding the LIWC and Inquirer features on top of synsets further improves accuracy by 2%, giving us our highest accuracy for the max label task of 47%. This produced our best model: SVM with synset and sentiment lexicon features. Notably, precision increases significantly when we added sentiment lexicons. We considered adding features from additional sentiment lexicons as well, but did not find any improvement in accuracy so we omit them from our model. This demonstrates that the sentiment lexicons are a useful feature for sentiment analysis and that groups of related words can provide very useful information about expected user reactions.

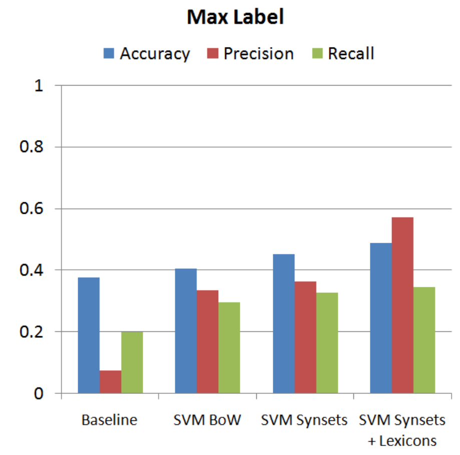

Figure 2: Performance of different models on max label task.

3.2 Label Presence Task Results

We used the same models in the label presence task as the max label task. Once again we established a naive baseline. In this case the naive baseline is to just predict that a label is present for all confessions if it is present for the majority of confessions; otherwise the baseline predicts the label is not present for all confessions. The baseline accuracy is 64%. Results are reported in figure 3 and values are the unweighted averages across the five categories. We use unweighted rather than weighted average to account for the unbalanced distribution of votes across categories.

Once again the basic BoW model is able to beat the baseline. Accuracy increases to 68% along with a signficant increase in precision. In contrast to the max label task, when we move to synset features we actually see a decrease in performance compared to BoW. However, when we added lexicon features to synsets, this model was the best, achieving a high accuracy (70%), precision (66%), and recall (61%). Therefore the best model for both tasks is to use synset and lexicon features.

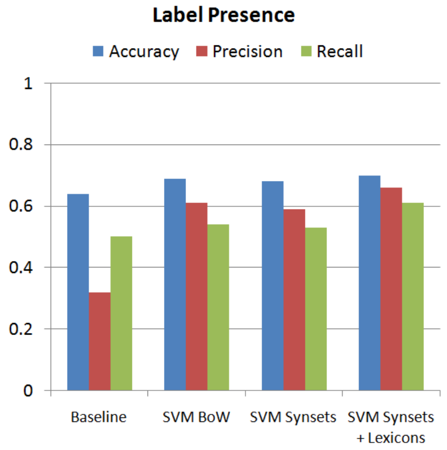

Figure 3: Performance of different models on label presence task.

4 Discussion

4.1 Comparison to Human Prediction

One might ask what is the difficulty of our two tasks and what level of accuracy would be considered successful. To answer the question of how hard the two tasks are, we can compare our system's performance against that of humans. We conducted an scaled-down version of the experiment where we had humans attempt the same two classification task as our models. Performance at the human level is often considered the target goal in sentiment analysis. Notice that we are not asking humans what are their reactions; we are asking them to predict what they think *other* people would have voted on.

We provided training examples and asked two human subjects to perform the two tasks on 75 testing examples for each task. Figures 4 and 5 show the performance of our system against two human subjects[3]. For the max label task our system had lower accuracy than humans while on the label presence task our system had slightly better accuracy than humans. On the whole, performance of our system approaches the level of humans. A more interesting finding is that both tasks are difficult for humans as well. The max label task is especially challenging and neither human subject was able to reach 40% accuracy.

4.2 Other Attempts

In addition to what we used in our final model, we had other work that taught us more about extracting emotion from EP.

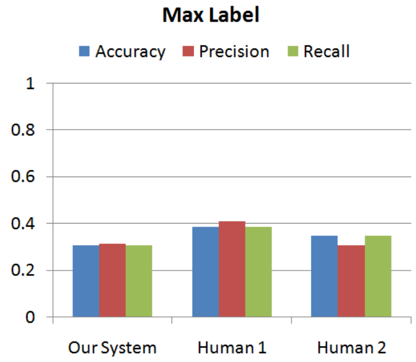

Figure 4: Human comparison for max label.

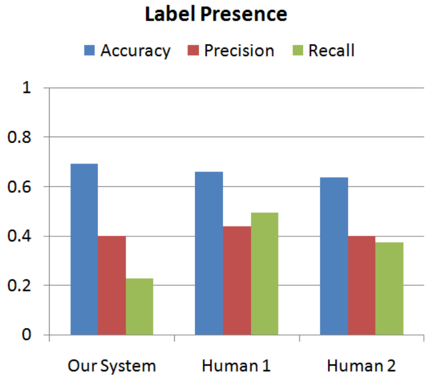

Figure 5: Human comparison for label presence.

4.2.1 Naive Bayes

Initially we worked with both Naive Bayes and SVM classifiers; however, due to the unbalanced distribution of categories, Naive Bayes tended to classify the vast majority of test examples as the most popular category. We were unable to correct for this and dropped its use in favor of SVM.

4.2.2 Latent Dirichlet Allocation (LDA)

The idea that a given topic can elicit a given emotion can be useful to predict the presence of categories. One possible feature model is to select the topic of the confession, and find the vote distribution for that particular topic. If the confession contains a single topic, we can model the probability of a reader selecting vote category v of confession k as:

$$p^{(k)}(v) = \sum_i p(v|t_i)p^{(k)}(t_i)$$

where t_i is each topic, and $p^{(k)}(t_i)$ is the probability of the topic occurring in confession k. In the max label task, we find the v that gives maximum $p^{(k)}$. This formula also works if the article contains multiple topics, by assuming that $p^{(k)}(t_i)$ is the proportion of the article containing topic t_i, and that the relationship between topics and vote distributions is linear. We create a heuristic in which we find a least squares estimate of the parameters $p(v|t_i)$, assuming

[3] For the max label task, due to the unbalanced distribution of categories we used a balanced human testing set instead of a random subset of the original testing set. Note that this is a harder problem for our SVM classifier since it was trained on an unbalanced training set. As a result the numbers reported here are lower than the ones reported in Results.

$p^{(k)}(v) = 1\{v = v^{(k)}_{max}\}$, in order to increase the difference between the maximum category and the others.

LDA allows us to estimate the presence of topics (Blei et al., 2003), whether it be the likelihood of a particular topic in each document, or the proportions of various topics within each document. Thus, we ran LDA on our documents after the preprocessing to get a new set of features over BoW, with various numbers of topics. Next, we took the derived features and tried our heuristic. First we ran GibbsLDA++[4], varying the number of topics assumed to be in the collection. Next we used our heuristic on the predicted $p^{(k)}(t_i)$ in each scenario. Our resulting test accuracies are shown in figure 6.

In addition, we tried to improve our feature set by including LDA-derived features. We ran an SVM on the combined features but found that this did not give significantly better performance.

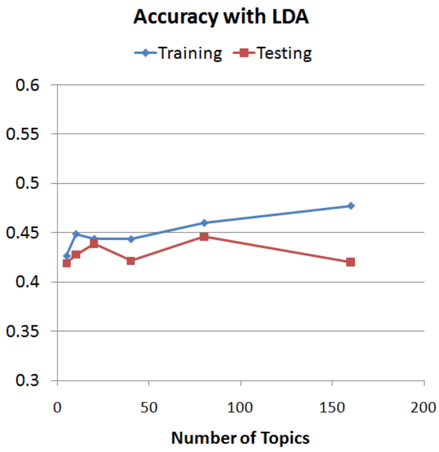

Figure 6: Accuracy using LDA-derived features.

5 Conclusions

Overall, the success and failures of all these different approaches gave us a good overall picture of the challenges of sentiment analysis on the Experience Project, and provide some guidelines for sentiment analysis with other sets of data in the future. First, we note the use of colloqiual and slang language in most of the confessions. The use of spell checking corrected for this somewhat. Nonetheless, the synset and sentiment lexicons we used are better suited to more formal styles of writing. An alternative approach is to replace our synsets and lexicons with "slang" versions or even the automatic generation of sentiment lexicons on a slang corpus.

Another area of interest is the difficulty in correlating topics with sentiment. Intuition says that topics themselves should portray different sentiments, and so should be useful for sentiment analysis. This method turns out to be fairly crude, as sometimes topics may be too neutral or too general to actually be good indicators of mood. For example, one of the topics found with LDA turned out to contain the topic about relationships. It is possible for someone to complain angrily about their current relationship, cry over the impending end of a relationship, or laugh because of a happy moment during the relationship. All of these get mapped into the same topic, but each has a substantially different mood.

6 Acknowledgments

The authors would like to thank Josh Falk for his help, and Professors Dan Jurafsky, Andrew Ng, and Chris Potts for their guidance.

References

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for textbased emotion prediction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 579–586.

Plaban Kumar Bhowmick, Anupam Basu, and Pabitra Mitra. 2009. Reader perspective emotion analysis in text through ensemble base multi-label classification framework. *Computer and Information Science*, 2(4):64–74, November.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January.

Moshe Koppel and Jonathan Schler. 2006. The importance of neutral examples in learning sentiment. *Computational Intelligence*, 22(2):100–109.

Rada Mihalcea and Hugo Liu. 2006. A corpus-based approach to finding happiness. In *AAAI 2006 Symposium on Computational Approaches to Analysing Weblogs*, pages 139–144. AAAI Press.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2010. Recognition of affect, judgment, and appreciation in text. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 806–814.

James W. Pennebaker, Roger J. Booth, and Martha E. Francis, 2007. *Linguistic inquiry and word count: LIWC2007 operator's manual*. University of Texas.

Ryan Rifkin and Aldebaro Klautau. 2004. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, December.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, MA.

[4] http://gibbslda.sourceforge.net/