Towards Emotionally Intelligent Machines: A Mood Classification System

Lucy Vasserman
Pomona College

Problem & Motivation

Affective computing, building machines with emotional intelligence, is an important field of Artificial Intelligence research because future machines will need to connect with users on an emotional level in addition to performing complex computations. An emotionally intelligent computer must be able to both identify emotions in its user and express emotions itself. Here, we focus on the former by working to automate emotion recognition, specifically in text.

This work comes in two parts. First, we develope an emotional classification system. Using Naive Bayes classification, the system takes a piece of text as input and identifies the mood conveyed in the text (happy, sad, angry, etc). Next, we explore ways to improve upon this type of text classification task through a new document represtation. While evaluating the initial mood classifier, it was clear that many documents are misclassified because they contain multiple emotions coming from off-topic asides, comparisons to past or other's emotions, or even sarcasm. Being able to identify these things could help the system more accurately recognize the main emotion being expressed. This problem inspired the creation of a new document representation that looks at emotions at the sentence level instead of the document level "bag of words" as before.

Background & Related Work

In the relatively short history of sentiment analysis work, many systems have used movie and product reviews as training data to classify the sentiment of a target document and most classify along a spectrum of positive, neutral, or negative (Pang 2002; Sood RTS 2007; Turney 2002). Innovation and increased accuracy of these systems has come from different machine learning approaches (Alm 2005) as well as new approaches to feature selection and dealing with differing emotional connotations across domains (Owsley 2006). Aue et al. take a unique approach to the cross domain problem by using unsupervised learning with unlabeled data in one domain supplemented by labeled training data in another domain (Aue 2005). We also use unsupervised learning to refine training data for a supervised task. A few researchers have also explored other dimensions of sentiment (Bradley 1999; Mehrabian 1996).

Uniqueness of Approach

Mood Classifier

The core system is a text-based mood classifier. The system is trained on a large set of blog posts labeled with the self-declared mood of the author and uses a large set of features. Aside from the diversity of features used in classification, this system is unique in that it leverages unsupervised clustering techniques in order to make use of a large training dataset with noisy document labels.

Training Data

A Naive Bayes Classifier requires a substantial set of labeled truth data for the system to learn from. Of the challenges in creating such a classifier, finding the appropriate dataset is of great importance. In this case, the training data is a corpus of more than six hundred thousand blog posts from the blog site LiveJournal (LiveJournal 2009). LiveJournal allows users to tag each post with a mood (e.g. – happy, anxious, angry, surprised); this label is an ideal truth-value for a mood classifier.

While the size of the LiveJournal blogs dataset is ideal, the site gives users one hundred and thirty moods to choose from. Using all one hundred and thirty moods in training and classification would hinder accuracy of the classifier and create feature frequency databases too large for a system to use in real-time. More importantly, it is not clear that there is a meaningful textual distinction between posts labeled as 'peeved' and 'annoyed,' for example. If such a distinction does not exist, then training a classifier to make such a distinction is mathematically impossible. The smaller the number of labels, the more likely it is that such a distinction exists and can be learned by the system. To preserve accuracy, the training dataset needed to be compressed into a smaller set of labels in a systematic way that would ensure statistical similarity within the training data with a particular label. A K-Means Clustering approach was able to achieve this necessary reduction in the label space.

K-Means Clustering

While one could simply use the posts labeled as 'happy' for the 'happy' posts training data, we found that this set was relatively small and wasteful considering that posts labeled as 'ecstatic,' for example, should likely fall into the same category. To this end, we used *K-Means Clustering* to organize the one hundred and thirty moods labels into three groups: angry, happy, and sad and discarded the mood labels that formed other outlying clusters.

To see why it was necessary to discard outliers, the following are "mood" labels from the *LiveJournal* dataset: 'hungry,' 'artistic,' 'sleepy,' 'blah,' 'working,' 'silly,' 'sore,' 'numb,' 'hot,' and 'okay.' As is shown in this list, several of the mood labels were meaningless for our purposes and would only introduce noise into the system. For this reason, outlying moods that do not fall neatly into the categories of 'happy,' 'sad,' or 'angry' were removed.

Given the original dataset of approximately six hundred thousand blog posts labeled with one of one hundred and thirty moods, the K-Means Clustering algorithm was used to determine which moods were similar enough to be grouped together and which were outliers. While one could intuitively hypothesize which labels should be clustered together, usage trends of each label might result in unexpected differences – for example, posts labeled as 'envious' and 'jealous' may be distinguishable – dissimilar enough to create noise in the data. K-Means Clustering assures the statistical similarity withing classes and difference between classes that is needed for Naive Bayes.

To carry out K-Means Clustering, each mood was represented as a vector of feature data; each position in the vector corresponded to a feature and the values were the total number of times that feature occurred in all posts tagged with that mood. The features used in our vectors were all unigrams that occurred across all blog posts in the entire dataset. K-Means Clustering was initiated on the 130 data points by using the data points for the 'happy,' 'sad,' and 'angry' labels as the initial 3 clusters.

The similarity calculation between data points (moods) was simply the distance between their corresponding vectors. The vectors were clustered into k groups based on their similarity. After running the algorithm multiple times, several distinct groups emerged. Of the one hundred and thirty original mood labels, thirty-one of them fell clearly into three groups, representative of happy, sad, and angry moods. Only the posts tagged with those moods were used as training data. The final grouping of mood labels, ignoring individual outliers and full outlying clusters, is shown in Table 1 and the frequency of posts with those labels is shown in Table 2. From the original six hundred thousand posts in the corpus, about one hundred and thirty thousand posts, tagged with these thirty-one different moods, were used in the final training set.

Angry	**Happy**	**Sad**
Enraged	Giddy	Envious
Pissed off	Hyper	Lonely
Infuriated	Energetic	Depressed
Angry	Ecstatic	Intimidated
Irate	Giggly	Scared
Bitchy	Cheerful	Guilty
Aggravated	Bouncy	Confused
	Jubilant	Rejected
	Excited	Crappy
	Happy	Distressed
		Gloomy
		Jealous
		Sad
		Crushed

Table 1. The three prominent mood groups that emerged from K-Means

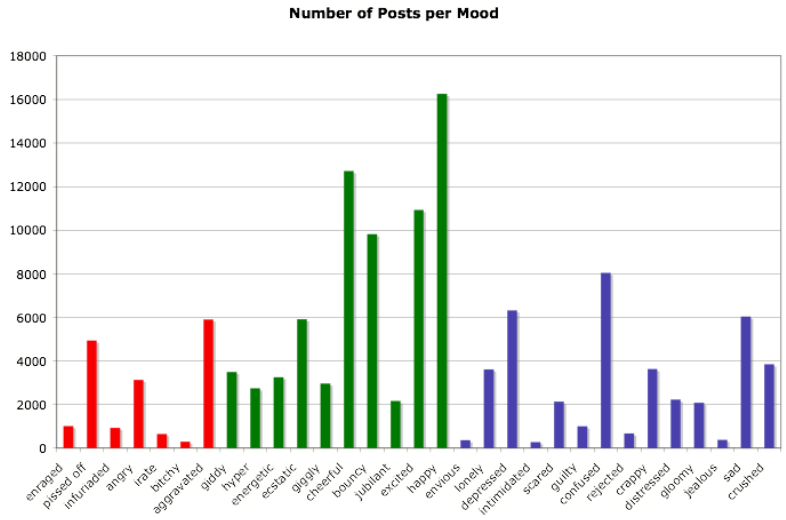

Table 2. A graph showing the distribution of blog posts among the mood labels. The horizontal axis gives the label, and the vertical axis gives the number of posts with that

Clustering on the set of LiveJournal mood labels

label in the corpus. The bar color represents the mood group (angry, happy, sad).

Classifier

After compressing the dataset into the categories of happy, sad and angry, the classifier itself was trained using this data. The system uses a Naive Bayes approach in order to calculate the conditional probability of a document *d* being a member of a class *c*, where the three possible classes are *happy*, *sad* and *angry*. Given these three probabilities, a document is then classified as the class with the highest conditional probability.

By Bayes theorem, the conditional probability of a class *c* given a document *d* is calculated as the prior probability of the class *c* multiplied by the probability of each feature in *d* given that class *c*. The conditional probability of a feature *f* occurring given that a document is of class *c* is equal to the training frequency of feature *f* in class *c* divided by the sum of all of frequencies of features in class *c*. The prior probability of a class *c* is simply equal to the fraction of training documents from class *c*.

The probability that a target document *d* is "happy" given the set of *n* features of *d,* which we'll call *f* is:

$$P(happy|f) = P(happy) * \prod_{i=1}^{n} P(f_i|happy)$$

The first term on the right side of the equation is the prior probability of any document being "happy". The second term on the right side of the equation is the product of the conditional probabilities of each of the features occurring given that a document is "happy".

Finally, after the three probabilities are calculated for a given document the highest probability class is interpreted as the most likely class and returned as the classification of the document.

Features

Along with the data used in training, the features selected for training and classification are the most important part of a text classification system. The system was originally trained using unigrams as the sole features, but was expanded to include many other classes of features that increased system accuracy.

In addition to three standard classification features, unigrams, bigrams and stems, the system includes features that are specific to the nature of blogs. A corpus of emoticons (emotional faces created from sequences of punctuation, e.g. :-) and :/) was used to define a feature set. People often use emoticons to explicitly state their mood, so it is clear that emoticons are a powerful feature for mood classification. One drawback is that the coverage of this feature set in the blogosphere is small, however, when present, emoticons are quite indicative of the author's mood.

A list of Internet slang, such as "lol" (which means 'laughing out loud') and "omg" (which means 'oh my god'), was gathered and used as a feature set in the classification system. While these terms would already have been used as individual features in the unigrams feature set, keeping them as a separate feature set enables more weight to be placed on them in the probability calculation. Similarly, a list of known highly emotional terms was gathered and used as a separate feature set. Again, while emotional terms are counted in the unigram feature set, giving them distinction as a separate feature set allows them to carry more weight in the classification.

These last three feature sets are typically only present in a small portion of texts. However, emotion words, Internet slang, and emoticons are generally highly indicative of the mood of the text and therefore are worthy to be included as features in the final system. Given that classification accuracy differs by feature sets in differing document types, it was critical that the features and weights used in classification were left configurable. The final version of the classifier allows the user to set how important each of the six features will be in classification on a scale of zero to ten. This will allow for customization based on the type of document being classified (emoticons and internet slang will be much more useful for classifying blogs or online conversations than for news articles).

Document Representation Approach to Improve Mood Classifier

Corpora

To explore ways to improve the above and similar systems we devised a novel document representation for a classification task, to improve upon the limited "bag of words" representation. A document is represented as an ordered list of emotions, one for each sentence. With a document represented in this format, there are many ways the overall emotion of a document could be revealed: perhaps the emotion is typically summarized in the first sentence, making that the best place to look for emotions; or maybe a document with many emotional changes is angry. We examined two corpora of tagged documents represented in this way to discover how the emotional sequence of a document relates to its overall emotion and how this information can aid in classification.

We created two corpora of documents represented as sequences of emotion, each tagged by the author with their overall emotion. The first corpora was a subset of the original blog data described above. Each sentence of the documents was classified as happy, sad, or angry by the core classifier and given an intensity score representing how much more likely the reported emotion was over the other two. A low intensity score represents a neutral document. Documents were represented as an ordered list of these emotion/intensity pairs along with the original truth-value tag.

The second corpus used a subset of training documents for Sood, et al.'s *Reasoning Through Search* system (Sood RTS 2007), which rates a document on a scale from -2, being very negative, to 2, very positive. These documents are movie and product reviews each tagged with one, two, four, or five stars, and their sentences were classified using the RTS system.

Grouping by K-Means Clustering

In analyzing the new corpus for relationships between emotional sequences and overall document sentiment, the first step is to identify common types of sequences. K-Means Clustering was used to find groups of documents with similar sequences. This required the creation of a similarity measure for two sequences of emotions.

We defined an algorithm based on string matching using dynamic programming. The algorithm finds the best alignment between two sequences of emotions by minimizing the emotional and intensity difference of aligned sentences. The unique feature of the algorithm is that is flexible to length differences, allowing three very happy sentences in one document to align with just one very happy sentence in another at low cost. This is useful because documents are considered similar when they have similar emotions in the same order, regardless of variations in length.

Using this similarity measure, we run k-means clustering on both datasets to find the common types of sequences. The sequences found corresponded to expected emotion sequences one would find in writing such as [*happy, sad, happy*]. Interestingly, there were almost no entirely negative documents (in the RTS dataset, which looks only at the positive/negative spectrum), while there were many fully positive ones. The clusters found with mostly negative documents always had small positive sections, usually at the beginning or end of the document. This suggests a trend for writing online, and would be worth more investigation.

Looking for Trends

Once the sequences had been grouped into clusters and outliers were removed, we used a popular data mining tool called Weka to look for relationships between sequence types and overall mood. As input Weka takes a list of features for each document and can create classifiers or run clustering algorithms on any or all of the data. The features we used to describe a document were the emotional sequence type, the number of sentences, the number of emotional changes (sentences whose emotion differed from that of the previous sentence), the overall classification from the core mood classifier, and the happy, sad or angry label. We experimented with classifiers to identify this label using the rest the features.

Results and Contributions

Mood Classifier

To evaluate the accuracy of the mood classification system, four test classifiers were trained on different combinations of feature sets. Each classifier was trained on 95% of the data, and tested on the remaining 5% with separate tests performed on the testing data for each of the three major mood groups (angry, happy and sad). Precision, recall and f-measure scores were calculated for each classifier, as seen in Table 3. The average f-measure was taken as an overall score for each classifier. Given a selection of text, each classifier returns a classification of either 'happy,' 'sad,' or 'angry.' Given the three possible classifications, the baseline performance would be 0.33.

In analyzing the results in Table 3, you will notice that only one of the test systems involved the three innovative feature types (Internet slang, emoticons and emotional words). The decision not to build a classifier solely based on each of these feature sets was due to the fact that most of the blog posts do not include these features. However, the performance of the mood classifier using all six features shows that including these feature sets is beneficial as they serve as good indicators of mood when they are present. The final system, shown at the bottom of Table 3 uses an ensemble of classifiers, each weighted by their importance, in order to reach a final classification. They weights are:
Unigrams – 9, Bigrams – 10, Stems – 9, Emotion words – 1, Internet slang – 1, and Emoticons – 1. This configuration of weights is the default for the system and performs with an average f-measure of 0.661, but the standalone mood classification system allows users to alter these weights as desired.

The f-measure also varies significantly by mood, with happy being much easier to classify than sad or angry ones. This is likely caused by the fact that there was more training data for happy texts. In addition, happy is a positive mood while sad and angry are both negative, and so identifying happiness is a much easier task than distinguishing sad and angry, even for humans.

	recall	precision	f-measure
Mood classifier using unigrams only.			
angry tests	0.499	0.465	0.481
happy tests	0.751	0.813	0.781
sad test	0.654	0.596	0.624
average	0.635	0.625	**0.630**
Mood classifier using bigrams only.			
angry tests	0.503	0.491	0.497
happy tests	0.763	0.821	0.791
sad test	0.67	0.602	0.634
average	0.645	0.638	**0.642**
Mood classifier using stems only.			
angry tests	0.502	0.457	0.478
happy tests	0.747	0.808	0.776
sad test	0.646	0.594	0.619
Average	0.632	0.620	**0.626**
Mood classifier using a weighted sum of all six different feature sets.			
angry tests	0.577	0.491	0.531
happy tests	0.751	0.84	0.793
sad test	0.685	0.62	0.651
Average	0.671	0.650	**0.661**

Table 3: Recall, Precision and F-Measure values for four test versions of the mood classification system.

Document Representation Approach to Improve Mood Classifier

Weka, a powerful data mining toolkit, was used to look for a relationships between the types of emotional sequences found in the clustering and the overall mood of a document. The initial analysis did not reveal any of the expected trends, and no correlations strong enough to aid in classification were found in the initial study.

The fact that there were no immediately obvious relationships between a document's emotion sequence and its overall emotion does not mean that these relationships do not exist. In the future, a deeper analysis of these corpora is necessary to truly determine what these

relationships are, if any. As a final step to this work, we extracted subsets of the corpora that are likely to contain useful information. One subset contains all documents that are misclassified by the original classifier. These should be analyzed to determine why they are difficult and if the document structure could be of use in those documents specifically. Another subset contains documents that are mostly monotone either positive or negative but with a short, isolated portion of the opposite sentiment (these documents were part of the RTS corpus). Our hypothesis is that these opposite areas are likely to be sarcasm. Mining this subset could help create a system that recognizes sarcasm, a perpetual difficulty for sentiment analysis systems. These and other subsets are set up to aid the continued analysis of the corpora and search for ways to improve text classification through document structure.

Conclusion

This work provides multiple distinct contributions to the field of sentiment analysis. First, the mood classifier is a concrete contribution of a novel system. The classifier is one of the first sentiment analysis systems to move beyond a two dimensional valence classification and with its good accuracy it is already usable for a variety of applications. One of these is the search engine ESSE, which uses the classifier to allow users to narrow search results by emotion as well as topic (Sood 2009). In addition, it provides a method of using unsupervised learning to make poorly labeled data usable for supervised learning, a new idea to the field.

The final contribution is more abstract, yet still meaningful. The over-simplification of documents through the bag of words representation is a pervasive problem in most text-based work. This work produces a new path for research on solving this problem and two corpora of tagged documents. We plan to make this corpora available to the sentiment analysis community to aid in the continuation of this research.

References

Alm, C.O., Roth, D., and Sproat, R. *Emotions from text: machine learning for text-based emotion prediction*. In Proceedings of HLT/EMNLP, 2005.

Aue, A. and Gamon, M. *Customizing sentiment classifiers to new domains: a case study*. RANLP, 2005.

LiveJournal. http://www.livejournal.com/, 2009.

M. M. Bradley and P. J. Lang. *Affective norms for English words (ANEW): Stimuli, instruction manual, and affective ratings*. Technical Report C-1, Center for Research in Psychophysiology, University of Florida, Gainesville, Florida, 1999.

Mehrabian, A. *Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament*. Current Psychology, Vol. 14, No. 4. (21 December 1996), pp. 261-292.

Owsley, Sara, Sood, Sanjay, Hammond, K. *Domain Specific Affective Classification of Documents*. AAAI Spring Symposia Computational Approaches to Analyzing Weblogs, 2006.

Pang, B., Lee, L. and Vaithyanathan, S. *Thumbs up? sentiment classification using machine learning techniques*. In Proceedings of EMNLP, pages 79 to 86, 2002.

Sood, Sanjay, Owsley, Sara, Hammond, K. and Birnbaum, L. *Reasoning Through Search: A Novel Approach to Sentiment Classification*. Northwestern University Tech Report Number NWU-EECS-07-05, 2007.

Sood, Sara Owsley and Vasserman, Lucy. *ESSE: Exploring Mood on the Web*. International Conference on Weblogs and Social Media Data, 2009.

Turney, P.D. *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*. In ACL, pages 417 to 424, 2002.