Pulse: Mining Customer Opinions from Free Text

Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger

Natural Language Processing, Microsoft Research, Redmond, WA 98052, USA,
(mgamon|anthaue|simonco|ringger)@microsoft.com,
http://research.microsoft.com/nlp/

Abstract. We present a prototype system, code-named *Pulse*, for mining topics and sentiment orientation jointly from free text customer feedback. We describe the application of the prototype system to a database of car reviews. Pulse enables the exploration of large quantities of customer free text. The user can examine customer opinion "at a glance" or explore the data at a finer level of detail. We describe a simple but effective technique for clustering sentences, the application of a bootstrapping approach to sentiment classification, and a novel user-interface.

1 Introduction

The goal of customer satisfaction studies in business intelligence is to discover opinions about a company's products, features, services, and businesses. Customer satisfaction information is often elicited in a structured form: surveys and focus group studies present customers with carefully constructed questions designed to gather particular pieces of information a company is interested in. The resulting set of structured, controlled data can easily be analyzed statistically and can be conveniently aggregated according to the specific dimensions of the survey questions or focus group setup. The drawbacks of structured studies are the expense associated with the design and administration of the survey, the limit that is necessarily imposed on the free expression of opinions by customers, and the corresponding risk of missing trends and opinions that are not expressed in the controlled situation. Additionally there is the risk of missing whole segments of the customer population that do not like to respond to a guided and structured set of questions.

Another potential source of information for business intelligence, which is becoming more and more pervasive and voluminous, is spontaneous customer feedback. This feedback can be gathered from blogs, newsgroups, feedback email from customers, and web sites that collect free-form product reviews. These can be rich sources of information, but these sources are much less structured than traditional surveys. The information is contained in free text, not in a set of answers elicited for a specific set of questions.

Paying people to mine this free-form information can be extremely expensive, and given the high volume of such free text is only feasible by careful sampling.[1]

With the advent of automatic techniques for text mining such as clustering and key term extraction, free-form customer opinions can be processed efficiently and distilled down to essential topics and recurring patterns of content. When trying to assess customer opinions, however, topic is only one of the dimensions that are of interest. As well as identifying what topics customers are talking about, it would be useful to characterize the opinions that they express about those topics.

Researchers have begun to focus on the analysis of opinion ('sentiment classification') typically using supervised machine learning techniques. [2] The project that we describe in this paper, code-named *Pulse*, combines the two dimensions of topic and sentiment and presents the results in an intuitive visualization. Pulse combines a clustering technique with a machine-learned sentiment classifier, allowing for a visualization of topic and associated customer sentiment. Pulse provides both a high-level overview of customer feedback and the ability to explore the data at a finer granularity. Pulse requires that only a small amount of data be annotated to train a domain-specific sentiment classifier.

Both sentiment detection and topic detection in Pulse are performed at the sentence level rather than at the document level. Document-level assessment, which is the focus of most sentiment classification studies, is too coarse for our purposes. In a review document, for example, we often find mixed positive and negative assessments such as: "OVERALL THE CAR IS A GOOD CAR. VERY FAST, THE ENGINE IS GREAT BUT FORD TRANSMISSIONS SUCK." Of course, even sentence-level granularity is too coarse in some instances, for example: "Its [sic] quick enough to get you and a few other people where you need to go although it isn't too flashy as far as looks go."[3] As we will discuss in further detail below, sentence-level granularity of analysis allows the discovery of new information even in those scenarios where an overall product rating is already provided at the document level.

We first describe the data to which Pulse has been applied (Section 2). We then describe the prototype system, consisting of a visualization component (Section 3.1), a simple but effective clustering algorithm (Section 3.2), and a machine-learned classifier that can be rapidly trained for a new domain (Section 3.3) by bootstrapping from a relatively small set of labeled data.

[1] It is worth noting that business intelligence is not the only scenario where customer satisfaction is of interest: individual customers often use resources on the web to find other people's reviews of products and companies to help them reach a decision on a purchase.

[2] Two notable exceptions are [1, 2].

[3] In future work we intend to investigate sentences with mixed sentiment, analyzing them at the level of the clause or phrase.

2 Data

We applied Pulse to a sample of the car reviews database[3]. This sample contains 406,818 customer car reviews written over a four year period, with no editing beyond simple filtering for profanity. The comments range in length from a single sentence (56% of all comments) to 50 sentences (a single comment). Less than 1% of reviews contain ten or more sentences. There are almost 900,000 sentences in total.

When customers submitted reviews to the website, they were asked for a recommendation on a scale of 1 (negative) to 10 (positive). The average score was 8.3 suggesting that people are enamored of their cars, or that there is a self-selection in the reviewers. Even reviews with positive scores contain useful negative opinions: after all a less-than-perfect score often indicates that the car may have a few shortcomings, despite a relatively high score.

For this reason we ignore the document-level scores and annotated a randomly selected sample of 3,000 sentences for sentiment. Each sentence was viewed in isolation and classified as "positive", "negative" or "other". The "other" category was applied to sentences with no discernible sentiment, as well as to sentences that expressed both positive and negative sentiment and sentences with sentiment that cannot be deduced without taking context and/or world knowledge into account.

The annotated data was split: 2,500 sentences were used for the initial phase of training the sentiment classifier (Section 3.3); 500 sentences were used as a gold standard for evaluation. We measured pair-wise inter-annotator agreement on a separate randomly selected sample of 100 sentences using Cohen's Kappa score.[4] The three annotators had pair-wise agreement scores of 70.10%, 71.78% and 79.93%.This suggests that the task of sentiment classification is feasible but difficult even for people.

3 System Description

Pulse first extracts a taxonomy of major categories (makes) and minor categories (models) of cars by simply querying the car reviews database. The sentences are then extracted from the reviews of each make and model and processed according to the two dimensions of information we want to expose in the final visualization stage: sentiment and topic. To train the sentiment classifier, a small random selection of sentences is labeled by hand as expressing positive, "other", or negative sentiment. This small labeled set of data is used with the entirety of the unlabeled data to bootstrap a classifier (Section 3.3).

The clustering component forms clusters from the set of sentences that corresponds to a leaf node in the taxonomy (i.e. a specific model of car). The clusters are labeled with the most prominent key term. For our prototype we implemented a simple key-word-based soft clustering algorithm with tf·idf weighting and phrase identification (Section 3.2). Once the sentences for a make and model of car have been assigned to clusters and have received a sentiment score from the sentiment classifier, the visualization component (Section 3.1) displays the clusters and the keyword labels that were produced for the sentences associated with that car. The sentences in a cluster can be displayed in a separate view. For each sentence in that view, the context (the original review text from which the sentence originated) can also be displayed. Figure 1 gives an overview of the system.

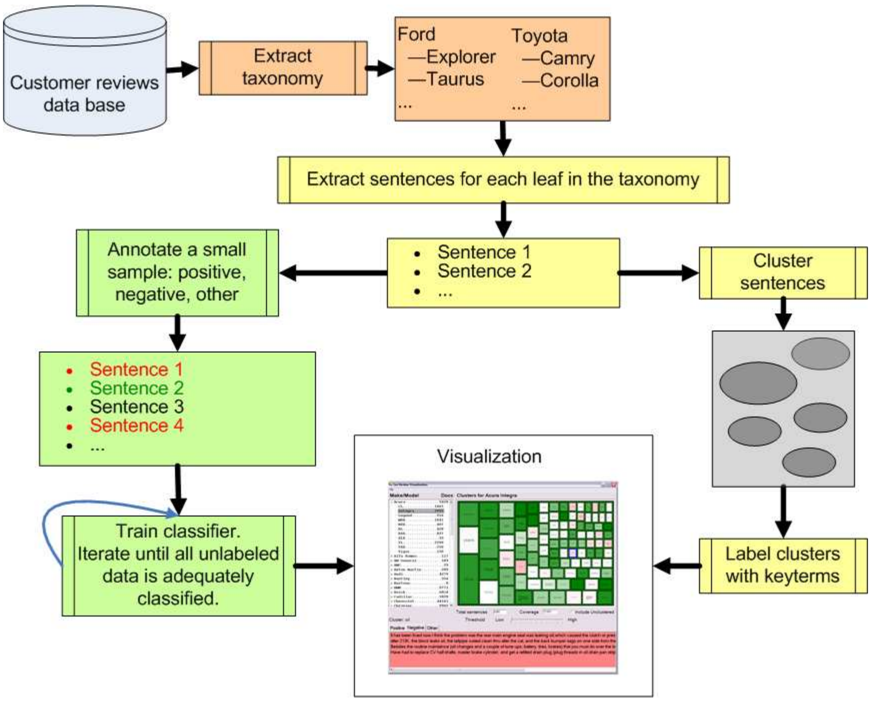

Customer reviews data base
Extract taxonomy
Ford
—Explorer
—Taurus
...
Toyota
—Camry
—Corolla
...
Extract sentences for each leaf in the taxonomy
Annotate a small sample: positive, negative, other
• Sentence 1
• Sentence 2
• ...
Cluster sentences
• Sentence 1
• Sentence 2
• Sentence 3
• Sentence 4
• ...
Visualization
Train classifier. Iterate until all unlabeled data is adequately classified.
Label clusters with keyterms

Fig. 1. Overview of the Pulse System Architecture

3.1 The Visualization Component

The visualization component needs to display the two dimensions of information, i.e. topic and sentiment, simultaneously. Another requirement is that it allow the user to easily access the specifics of a given topic. Pulse uses a Tree Map visualization [5] to display clusters and their associated sentiment. Each cluster is rendered as one box in the Tree Map. The size of the box indicates the number of sentences in the cluster, and the color indicates the average sentiment of the sentences in the box. The color ranges from red to green, with red indicating negative clusters and green indicating positive ones. Clusters containing an equal mixture of positive and negative sentiment or containing mostly sentences classified as belonging to the “other” category are colored white. Each box is also labeled with the key word for that particular cluster.

The Tree Map visualization allows the identification of the following information about the sentences associated with a given make/model at a glance:

- the overall sentiment associated with the make/model (indicated by the relative area in the entire Tree Map colored red or green)
- the most common topics that customers mention in the reviews for the make/model as indicated by the larger boxes
- the most positive and the most negative topics, indicated by the darkest shades of green and red in the cluster boxes.

Figure 2 shows a screenshot of the visualization in the cluster view. The taxonomy of makes and models (i.e. major and minor category) is displayed in the left pane, the Tree Map to the right of it, and the sentences in the tabbed display at the bottom.

The user has selected the Volkswagen Golf. The two biggest clusters appear in the boxes at the left of the Tree Map: “drive”, and “vw, service”. The user has chosen to inspect the “vw, service” cluster by clicking on it and viewing the negative sentences in the tabbed display at the bottom of the screen. The threshold slider has been set approximately three quarters of the way along, restricting the display to only sentences with high class probability. This has the effect of increasing precision at the expense of recall. Clicking on a sentence in the tabbed display brings up a window (not shown) that displays the entire review in which the selected sentence occurred, with each sentence colored according to sentiment.

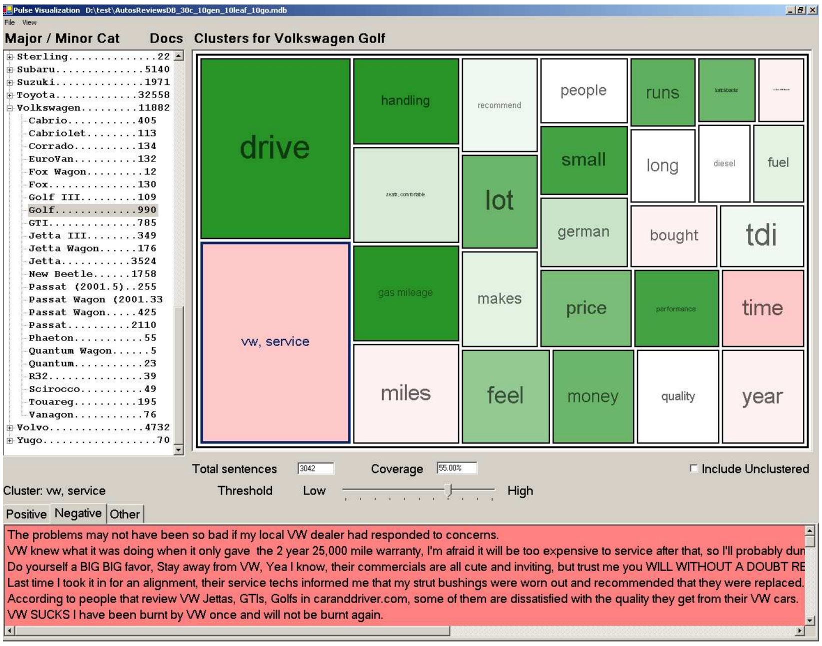
By choosing a menu option, the user can view a summary of the clusters in the form of simple “Top five” lists, where for a given make/model the top five terms overall, the top five positive terms and the top five negative terms are displayed. The top five display is very simple, and is not shown in the interests of brevity.

3.2 Clustering Algorithm

We experimented with several different clustering algorithms for finding salient patterns in the sentences:

- a k-means clustering algorithm using tf·idf vectors, as described in [6],
- an EM implementation of soft, non-hierarchical clustering[7],
- a hierarchical, entropy-based clustering algorithm[8], and
- an algorithm that used character n-gram feature vectors.

None of the approaches we tried produced clusters that we found satisfactory. Each algorithm was designed for a different task. The first two were designed for clustering documents which are much larger units of text than sentences. The third and fourth approaches were designed for clustering units of text that are much smaller than sentences, namely words and Internet search queries. We therefore formulated the following simple algorithm, which performs well.



```
⊞ Sterling..............22
⊞ Subaru..............5140
⊞ Suzuki..............1971
⊞ Toyota.............32558
⊟ Volkswagen.........11882
    Cabrio...........405
    Cabriolet........113
    Corrado..........134
    EuroVan..........132
    Fox Wagon.........12
    Fox..............130
    Golf III.........109
    Golf.............990
    GTI..............785
    Jetta III........349
    Jetta Wagon......176
    Jetta...........3524
    New Beetle......1758
    Passat (2001.5)..255
    Passat Wagon (2001.33
    Passat Wagon.....425
    Passat..........2110
    Phaeton...........55
    Quantum Wagon......5
    Quantum...........23
    R32...............39
    Scirocco..........49
    Touareg..........195
    Vanagon...........76
⊞ Volvo...............4732
⊞ Yugo..................70
```

Total sentences 3042 Coverage 55.00% □ Include Unclustered

Cluster: vw, service

The problems may not have been so bad if my local VW dealer had responded to concerns.
VW knew what it was doing when it only gave the 2 year 25,000 mile warranty, I'm afraid it will be too expensive to service after that, so I'll probably dun
Do yourself a BIG BIG favor, Stay away from VW, Yea I know, their commercials are all cute and inviting, but trust me you WILL WITHOUT A DOUBT RE
Last time I took it in for an alignment, their service techs informed me that my strut bushings were worn out and recommended that they were replaced.
According to people that review VW Jettas, GTIs, Golfs in caranddriver.com, some of them are dissatisfied with the quality they get from their VW cars.
VW SUCKS I have been burnt by VW once and will not be burnt again.

Fig. 2. Screenshot of the Pulse user interface showing the taxonomy of makes and models, the Tree Map with labeled clusters and sentiment coloring, and individual sentences from one cluster.

The input to the clustering algorithm is the set of sentences S for which clusters are to be extracted, a stop-list W_{Stop} of words around which clusters ought not to be created, and (optionally) a "go list" W_{Go} of words known to be salient in the domain.

1. The sentences, as well as the stop and go lists, are stemmed using the Porter stemmer. [9]
2. Occurence counts C_W are collected for each stem not appearing in W_{Stop}.
3. The total count for stems occuring in W_{Go} is multiplied by a configurable parameter λ_1.
4. The total count for stems with a high tf·idf (calculated over the whole data set) is multiplied by a configurable parameter λ_2.
5. The total count for stems with a high tf·idf (calculated over the data in the given leaf node of the taxonomy) is multiplied by a configurable parameter λ_3.
6. The list of counts is sorted by size.
7. To create a set of N clusters, one cluster is created for each of the most frequent N stems, with all of the sentences containing the stem forming the

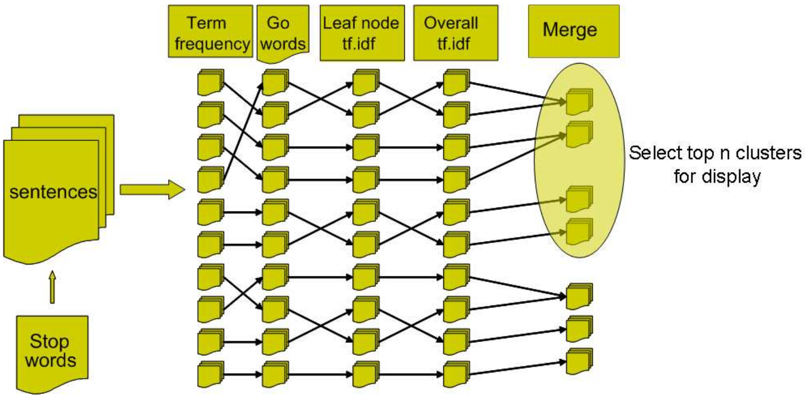


Fig. 3. Diagram of the clustering algorithm

cluster. The clusters are labeled with the corresponding stem St [4] An optional additional constraint is to require a minimum number M of sentences in each cluster.

8. Two clusters C_1 and C_2 are merged if the overlap of sentences S_{C1C2} contained in both C_1 and C_2 exceeds 50% of the set of sentences in C_1 or C_2. If the labels of C_1 and C_2 form a phrase in the sentences in S_{C1C2}, the new cluster C_{12} is labeled with that phrase, otherwise it is labeled with both labels, separated by a comma.

An overview of the clustering approach is presented in Figure 3. The initial set of clusters is determined by term frequency alone. Go words and the two tf·idf weighting schemes each re-rank the clusters, and finally some of the clusters are merged and a fixed number of clusters is selected off the top of the ranked list for display.

The stop word list consists of two components. The first is a manually specified set of function words and high frequency, semantically empty content words such as “put”. The more interesting and essential part of the stop list, however, is the set of the top N features from the sentiment classifier, according to log likelihood ratio (LLR) with the target feature [10]. By disallowing words known to be highly correlated with positive or negative sentiment we ensure that the topics represented in the clusters are orthogonal to the sentiment of the feedback. Term frequency (tf)/inverse document frequency (idf) weighting is a common technique in clustering. Terms with high tf·idf scores are terms that have a high degree of semantic focus, i.e. that tend to occur frequently in specific subsets of documents. The tf·idf weighting scheme that we employed is formulated as

[4] We experimented with N in the range 30–50. For larger values of N, the visualization became too cluttered to be useful.

$$weight(i,j) = \left\{ \begin{matrix} (1 + log(tf_{i,j})log\frac{N}{df_i} & \text{if } tf_{i,j} \geq 1 \\ 0 & \text{otherwise} \end{matrix} \right\} \quad (1)$$

where $tf_{i,j}$ is the term frequency of a word w_i, and df_i is the document frequency of w_i, i.e. the number of documents containing w_i and N is the number of leaf nodes in the taxonomy ([6]).

Since we cluster sentences, i.e. sub-document units, we are not interested in using tf·idf for weight assignment in the sentence vectors themselves. We rather want to find out which of all the terms in all the reviews for one make/model leaf node should be given increased importance when clustering sentences in that leaf node. In order to assign a per-word weight that we can use in clustering, we calculate two different per-word scores:

1. We can take df_i to be the number of reviews under a given leaf node which contain w_i. $tf_{i,j}$ is taken to be the term frequency in the reviews in that leaf node. A high score in this scenario indicates high semantic focus within the specific leaf node.
2. If df_i is defined to be the number of reviews in the whole collection which contain w_i, and $tf_{i,j}$ is the term frequency in the whole collection, a high tf·idf score indicates a term with high semantic focus in the whole domain.

These two scores allow the customization of the weighting of terms according to their leaf-node specific salience or their domain-specific salience. The more uniform a collection of data is, the more the two measures will coincide. In addition to weighting the terms for clustering according to these two scores, Pulse also allows for the use of a go word list (i.e. a domain dictionary) where such a resource is available.[5] The go word list allows us to steer the clustering toward terms that we know to be salient in the domain, while at the same time still allowing us to discover new clusters automatically that do not appear in our domain dictionary. For example, for many makes and models of car, terms like "family" and "snow", which were not in the domain-specific go list, emerged as labels for clusters.

Finally, it must be noted that not all sentences are assigned to a cluster. Unassigned sentences are assigned to a nonce cluster, which is not displayed unless the user explicitly chooses to see it. Also, because more than one cluster keyword can appear in a given sentence, that sentence may correspondingly belong to more than one cluster (soft clustering).

3.3 Sentiment Analysis

As mentioned in the introduction, machine-learned approaches to sentiment analysis are a topic that has received considerable attention from researchers over the past few years. A number of different approaches have been applied

[5] For the autos domain, W_{Go} was created by extracting entry keywords from a freely-available online automotive dictionary.

to the problem. The annotated movie review data set made publicly available by Pang and Lee [11, 12] has become a benchmark for many studies. The data consists of 2000 movie reviews, evenly split between positive and negative instances. The task is to determine which are positive and which are negative. Classification accuracies approaching 90% for this binary classification task are cited [11–13]. Features for sentiment classification typically consist of simple unigram (term) presence. However, the following characteristics of the car reviews data set rendered techniques previously cited in the literature unsuited to our task:

1. Since we are aiming at sentence-level classification, we are dealing with much shorter textual units than the full movie reviews, which range from a few sentences to several paragraphs.
2. The car reviews are not annotated at the sentence level. Since one of the main purposes of Pulse is to avoid the cost associated with manual examination of data, we would like to be able make do with as little annotated data as possible.
3. The Movie Review data set is carefully selected to be balanced, and to contain only extremes, i.e. only very strong recommendations/disrecommendations. The car review data, on the other hand, are strongly imbalanced, with positive reviews predominating.
4. While the movie reviews are generally well-written, the car review sentences are frequently ungrammatical, fragmentary and idiosyncratic. They contain numerous misspellings, acronyms, and a more telegraphic style.

We ignored the recommendation scores at the review (document) level for two reasons. First, since we focus our classification on individual sentences, we cannot make the assumption that in a review all sentences express the same sentiment. If a reviewer decides to give 8 out of 10 stars, for example, the review is likely to contain a number of positive remarks about the car, with a few negative remarks–after all the reviewer had a reason to not assign a 10-out-of-10 score. Secondly, we wanted to investigate the feasibility of our approach in the absence of labeled data, which makes Pulse a much more generally applicable tool in other domains where customer feedback without any recommendations is common.

Because the sentences in the car reviews database are not annotated, we decided to implement a classification strategy that requires as little labeled data as possible. We implemented a modified version of Nigam et al.'s algorithm for training a Naive Bayes classifier using Expectation Maximization (EM) and bootstrapping from a small set of labeled data to a large set of unlabeled data [14]. The classification task in our domain is a three-way distinction between positive, negative, and "other". The latter category includes sentences with no discernible sentiment (a sentiment-neutral description of a model, for example), sentences with balanced sentiment (where both a positive and a negative opinion are expressed within the same sentence), and sentences with a sentiment that can only be detected by taking the review context and/or world knowledge into account. This bootstrapping allowed us to make use of the large amount of unlabeled data in the car reviews database, almost 900,000 sentences. The algorithm requires two data sets as input, one labeled (D_L), the other unlabeled (D_U).

1. An initial naive Bayes classifier with parameters θ is trained on the documents in D_L.
2. This initial classifier is used to estimate a probability distribution over all classes for each of the documents in D_U. (E-Step)
3. The labeled and unlabeled data are then used to estimate parameters for a new classifier. (M-Step)

Steps 2 and 3 are repeated until convergence is achieved when the difference in the joint probability of the data and the parameters falls below the configurable threshhold ϵ between iterations. We also implemented two additional modifications described by [14]:

1. A free parameter, δ, was used to vary the weight given to the unlabeled documents.
2. Mixtures were used to model each class.

In order to prepare the data for classification, we normalized each sentence using some simple filters. All words were converted to lower-case, and numbers were collapsed to a single token[6]. For each sentence, we produced a sparse binary feature vector, with one feature for each word or punctuation mark. Our labeled data were the hand-annotated sentences described in section 2. 2500 of these were used to train the classifier D_L, and the remaining 500 were reserved as a test set. The classifier was trained and then evaluated on the test set. The data set shows a clear skew towards positive reviews: in the annotated data set, positive sentences comprise 62.33% of the data, sentences of type “other” comprise 23.27%, and negative sentences 14.4%. Because of this skew toward a positive label in the data set, overall accuracy numbers are not very illuminating–naively classifying every sentence as positive will result in a 62.33% accuracy. Instead we evaluate the classifier by considering the precision vs. recall graph for the negative and “other” classes, which are the classes with the fewest occurrences in the training data. We achieved some of the best results on the negative and “other” classes by using a δ of 1.0. Figure 4 shows that the classifier is able to achieve reasonable precision on the negative and “other” classes at the expense of recall. In domains with very large amounts of free-form customer feedback (typically so large that complete human analysis would not even be attempted) low recall is acceptable. The “ other” category is clearly the hardest to identify, which is not surprising given its very heterogeneous nature. Recall on the positive class is nearly constant across precision values, ranging from 0.95 to 0.97.

[6] We leave it for future research to also employ automatic spelling correction. We expect this to be useful in the car review domain, where misspellings are rather abundant (the word “transmission”, for example, is spelled in 29 different ways in this data set).

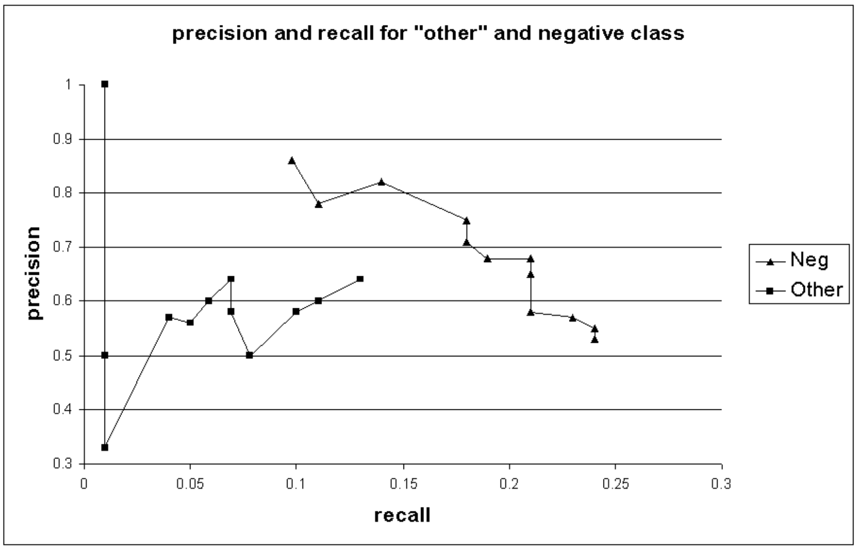


Fig. 4. Precision vs. Recall for Negative and OtherClass

4 Conclusion

Much has been written about the individual fields of clustering and sentiment analysis on their own. Combined, however, and paired with an appropriate visualization they provide a powerful tool for exploring customer feedback. In future work we intend to apply this combination of techniques to the analysis of a range of data, including blogs, newsgroups, email and different customer feedback sites. We are currently working with various end-users who are interested in using a practical tool for performing data analysis. The end-user feedback that we have received to date suggests the need for improved text normalization to handle tokenization issues, and the use of a speller tool to identify and normalize spelling variants and misspellings. Finally, our research will continue to focus on the identification of sentiment vocabulary and sentiment orientation with minimal customization cost for a new domain. We have begun experimenting with a variation of a technique for bootstrapping from seed words with known orientation [1, 2] with promising initial results [15]. As opposed to the approach described here, the new approach only requires the user to identify a small (about ten item) seed word list with known strong and frequent sentiment terms and their orientation. The only additional task for the user would be to verify and edit an extended seed word list that the tool will automatically produce. Once this extended list has been verified, a sentiment classifier can be produced without further labeling of data.

References

1. Turney, P.D.: Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: Proceedings of ACL 2002. (2002) 417–424
2. Turney, P.D., Littman, M.L.: Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical Report ERC-1094 (NRC 44929), National Research Council of Canada (2002)
3. Microsoft_Corporation: Msn autos (http://autos.msn.com/default.aspx) (2005)
4. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological measurements **20** (1960) 37–46
5. Smith, M.A., Fiore, A.T.: Visualization components for persistent conversations. In: CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press (2001) 136–143
6. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, Massachusetts (1999)
7. Meila, M., Heckerman, D.: An experimental comparison of several clustering and initialization methods. Technical report, Microsoft Research (1998)
8. Goodman, J.: A bit of progress in language modeling. Technical report, Microsoft Research (2000)
9. Porter, M.: An algorithm for suffix stripping. Program **14** (1980) 130–137
10. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics **19** (1993) 61–74
11. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of EMNLP 2002, EMNLP (2002) 79–86
12. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of ACL 2004, ACL (2004) 217–278
13. Bai, X., Padman, R., Airoldi, E.: Sentiment extraction from unstructured text using tabu search enhanced markov blanket. In: Proceedings of the International Workshop on Mining for and from the Semantic Web. (2004) 24–35
14. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using em. Machine Learning **39(2/3)** (2000) 103–134
15. Gamon, M., Aue, A.: Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. In: Proceedings of the ACL 2005 Workshop on Feature Engineering for Machine Learning in NLP, ACL (to appear)