Peer Review in Online Forums: Classifying Feedback-Sentiment

Greg Harris
Department of Computer Science
University of Southern California
Los Angeles, California
gfharris@usc.edu

Anand Panangadan and Viktor K. Prasanna
Ming-Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, California
{anandvp, prasanna}@usc.edu

Abstract

Replies posted in technical online forums often contain feedback to the author of the parent comment in the form of agreement, doubt, gratitude, contradiction, etc. We call this feedback-sentiment. *Inference of feedback-sentiment has application in expert finding, fact validation, and answer validation. To study feedback-sentiment, we use nearly 25 million comments from a popular discussion forum (Slashdot.org), spanning over 10 years. We propose and test a heuristic that feedback-sentiment most commonly appears in the first sentence of a forum reply. We introduce a novel interactive decision tree system that allows us to train a classifier using principles from active learning. We classify individual reply sentences as positive, negative, or neutral, and then test the accuracy of our classifier against labels provided by human annotators (using Amazon's Mechanical Turk). We show how our classifier outperforms three general-purpose sentiment classifiers for the task of finding feedback-sentiment.*

1 Introduction

Online technical discussion forums contain a wealth of information, intermingled with misinformation. Ideally, misinformation gets questioned by other forum members, leading to a self-correcting community. This is a form of peer review, making the replies a valuable part of the conversation.

We define *feedback-sentiment* as the sentiment in a forum reply directed either toward its parent comment, or toward the author of the parent comment. Feedback-sentiment can have a positive or negative polarity. It can give clues as to the perceived quality or accuracy of information in a parent comment. This, in turn, gives an indication of the trustworthiness of the author of the parent comment as a source of information. Positive feedback-sentiment can validate facts, while negative feedback-sentiment can indicate contradiction.

Feedback-sentiment can be manifest in many ways, including:

- (dis)agreeing with a comment/author
- showing appreciation
- insulting the author
- questioning/expressing doubt
- listing a counterexample

Feedback-sentiment is distinct from general sentiment, which is the overall attitude expressed in a document, or the affective state of the author while writing the document. While feedback-sentiment is more narrowly defined than general sentiment, it is more broadly defined than the subject of related work: attitude [6], agreement [9], and antagonism [13]. Feedback-sentiment can appear in the same sentence as sentiment directed elsewhere. An example can be seen in this sentence:

> "Yeah, they've got the worst customer service ever."

This has *positive* feedback-sentiment, since it shows agreement with the parent post. The overall negative feeling toward the customer service is irrelevant.

In this work, we propose an interactive method to train a feedback-sentiment classifier that will work on forum replies. We use an interactive decision tree that assists the user in navigating the most commonly used response patterns. The primary benefit of our method is that it does not require the user to begin with labeled training data, which is the chief drawback of supervised learning techniques. We present heuristics that are applicable to feedback-sentiment classification that enable the user to rapidly visualize and label only those portions of the data most useful for increasing precision and recall. We applied our interactive decision tree approach to classify feedback-sentiment in posts made to the Slashdot online discussion board. Note that our classifier directly classifies individual sentences as positive, negative, or neutral, without the intermediate step of deciding whether the sentence is subjective. We evaluate our trained classifier by comparing it to human annotation performed using Amazon's Mechanical Turk service. We were not able to find other classifiers trained for feedback-sentiment, so we compare ours against three more general-purpose sentiment classifier baselines.

Feedback for an online comment can indicate the truth or value of the comment, which reflects on the level of expertise of the author. This makes feedback a potentially valuable tool for the task of expert finding. Smirnova and Balog [16] surveyed the expert finding systems presented in the TREC Enterprise Search track. They reported that the systems generally fell into two categories: candidate-based, and document-based. Both categories only determine expertise by considering the text authored by each person. We believe a feedback-sentiment classifier could augment these systems by providing a view of peer opinions. Kumar and Ahmad [10] also had the idea of including the sentiment of replies when looking for experts in blog posts. Rather than looking directly for feedback-sentiment, however, they looked to see if the general sentiment polarity of the comments matched that of the blog post.

The contributions of our work are defining and studying feedback-sentiment and finding two heuristics that help with feedback-sentiment classification. We also developed an interactive decision tree for text that works with the heuristics, facilitating classifier training in a short amount of time and with no prior labeled training data. The resulting classifier outperforms general-purpose sentiment classifiers for the task of finding feedback-sentiment.

2 Related Work

General-purpose sentiment analysis is not a precise way to study feedback-sentiment. Some special-purpose work has been done, however, that investigates sub-areas of feedback-sentiment:

Hassan et al. [5] studied subgroups in online communities using graphs with both positive and negative edges. The signed edges represent attitudes of community members toward one-another. The groundwork for finding the edges was laid in their earlier work classifying attitude [6]. Here, they use supervised learning to first determine whether a sentence contains attitude, then they determine the polarity. Since they are looking only for attitude directed at other members, they consider only sentences that contain a second-person pronoun (you, your, etc.). This differs from our research, because we seek to identify not only the attitude toward the author of a comment, but also any feedback relating to the comment itself. Therefore, filtering on second-person pronouns discards too many sentences that contain feedback-sentiment. Looking ahead, none of the most frequent response sentences found in Table 2 contain second-person pronouns, yet they all contain feedback-sentiment.

Danescu-Niculescu-Mizil et al. [1] studied politeness in online interactions; since politeness indicates respect, this is a form of feedback. Sood et al. [18] studied malicious and insulting behavior on social news websites. Musat et al. [13] studied antagonism in online communities, which they defined as direct sentiment towards the authors of previous comments. They used a general-purpose sentiment lexicon to find polarized words that refer to a second-person pronoun. Their work only considers negative sentiment and the model was not evaluated with ground truth.

Feedback-sentiment also encompasses Agreement/Disagreement. Most research on agreement detection has been done using the ICSI meeting recording corpus, introduced by Janin et al. [9]. This corpus contains about 72 hours of audio recorded during the course of 75 meetings. It contains word-level transcriptions and numerous details about the speakers. Hillard et al. [7] used the ICSI corpus to create a classifier that recognized agreement and disagreement utterances. Their model used unigrams and bigrams along with prosodic cues, such as pause, fundamental frequency, and duration. Galley et al. [2] and Hahn et al. [4] built on this work. Germesin and Wilson [3] studied agreement detection using the AMI meeting corpus, which is has more annotated meetings and includes the targets of agreements.

Our approach to classification is based on interactive decision trees. Interactive decision tree systems are generally used in applications where a domain expert wants to control the construction of the classifier. A key benefit interactive approaches provide is helping the user visualize the data. Liu and Salvendy [12] evaluated several decision tree visual representations, including outline views, node-link diagrams, tree-maps, tree rings, and icicle plots. All these representations and systems are designed for numeric features, rather than text. They visualize data using graphics, while we use tables of short sentences. Also, as supervised learning techniques, they still require labels for the training data. Our system starts with no labeled data, guiding the user in identifying the most important phrases.

3 Dataset

We used data from Slashdot[1] (tagline: "News for nerds, stuff that matters"), one of the oldest and most popular discussion boards on the World Wide Web. The Slashdot home page lists summaries of news stories, often about technology, along with a link to the original story. Readers post comments in the site's threaded discussion forum.

We collected all news summaries (100 thousand) and

[1] http://slashdot.org

comments (25 million) available in Slashdot's archive spanning June 26, 2002, through August 31, 2013. Nearly 5 million of the comments initiated new discussion threads, with the rest being posted in reply to one of these.

4 Feedback-Sentiment Characteristics

Below we report characteristics of the dataset related to feedback-sentiment that influenced our approach to classification.

4.1 First-Sentence Heuristic

We found that authors frequently express feedback-sentiment at the beginning of the reply. We observed the same pattern in other online threaded discussion boards, as well. This observation forms the basis of the heuristic that enables rapid visualization and classification using interactive decision trees:

> The first sentence in a reply contains the sentiment of the author toward the parent comment/author.

Following this heuristic, a sentiment-feedback classifier should prioritize the sentiment-bearing phrases in the first sentence over those that appear in later sentences.

In Slashdot, replies contain the same subject as the parent comment, only with "Re:" prepended (by default). Users can change this default subject and begin their reply in the subject line itself. We treat this edited subject line as the true first sentence of the reply. We also remove any quoted text from the reply before determining the first sentence.

4.2 Changing Response Patterns

Some of the frequent response patterns on Slashdot are recent additions to the public lexicon. For example, common sentiment-bearing acronyms, such as "LOL!" and "WTF?" only became popular with the advent of instant messaging and mobile texting. The phrase "[citation needed]" has become popular since its introduction as a meta-tag in Wikipedia[2] in 2006. The complete sentence "This." has seen a recent surge in usage in Slashdot, as shown in Table 1. This implies that a feedback-sentiment classifier will have to be retrained periodically to accommodate changing speech patterns.

Year	"citation needed"	"This."
2002	0	18
2003	0	34
2004	0	39
2005	0	28
2006	34	14
2007	321	41
2008	1,241	138
2009	2,511	448
2010	2,433	825
2011	2,102	1,460
2012	1,925	1,830
to 8/31/2013	1,091	1,337

Table 1. Frequency of replies that contain "citation needed" and frequency of variations of "This." as a complete first sentence in Slashdot replies.

4.3 Forum-Specific Characteristics

Slashdot has some unique characteristics (Lampe and Resnick [11] give a good overview). The randomized, distributed moderation system leaves many people in the position of having experience as a moderator, yet without any remaining moderation points. This leads to replies containing instructions for other moderators. These replies are variations of phrases like "+1" and "MOD PARENT DOWN," which contain feedback-sentiment, but are not generally found in other forums.

Slashdot also shows characteristics that are common among public online forums, but may be uncommon in internal corporate discussion boards which provide less anonymity. In particular, we found a relatively high level of profanity. We also found that the use of proper grammar, spelling, capitalization, and punctuation varied widely between comments. This implies that a feedback-sentiment classifier should be trained on data from the specific forum in which it will be used.

5 Interactive Decision Tree

The observations described in Section 4 indicate that response patterns change over time, and also that a single feedback-sentiment classifier will not be well-suited for all forums. Ideally, a classifier should be trained for each individual forum, and updated periodically. Classifier training should be quick and easily performed, even for large datasets. We did not explore supervised learning techniques, due to the effort required to label significant num-

[2] http://www.wikipedia.org

bers of messages. Avenues such as Amazon's Mechanical Turk[3] could be used for large-scale labeling of training data; however, this method is not applicable to private or corporate discussion boards (since sensitive conversations and intellectual property must be tightly controlled).

We propose using an interactive binary decision tree, modified to work with text, for rapidly classifying feedback-sentiment. The root node of the decision tree represents all the first-sentences from the training data. The user then splits the node by providing a regular expression pattern. Complex regular expressions are not needed. Each pattern is a word, phrase, or punctuation mark. First sentences that contain the pattern are separated into one child node, and those that don't contain the pattern are sent to a different child node. Each of those child nodes is further split until the user is satisfied. The final, un-split leaf nodes contain first-sentences that are similar and are easily labeled as positive, negative, or neutral. A portion of the decision tree that was applied to the Slashdot data is shown in Figure 1.

Interactive decision trees must provide a way for the user to visualize the contents of each node. That is another reason we use only the first-sentences, and not the full reply. Individual sentences are short and can be displayed in a table that is easily skimmed by the user.

Settles [15] describes Active Learning as:

> "...the active learner aims to achieve high accuracy using as few labeled instances as possible..."

The interactive decision tree approach to feedback-sentiment classification follows this principle. The most frequent first-sentences in each leaf node are at least inspected by the user, if not fully annotated. If the user is satisfied with the precision of the most frequent sentences, then the whole leaf is deemed satisfactory. In this way, the user can label the leaves with high confidence and move on to parts of the tree that need further refining.

Greedy Heuristic

Some response phrases are very common, leading to first-sentences that are duplicated many times in the training data. We found that only 2.5% of the distinct first-sentences in the Slashdot data appeared with a frequency greater than one, yet they made up 14% percent of the total. Some of the most common full first-sentences are shown with their counts in Table 2.

Fitting the decision tree to the exact sentences found in Table 2 would account for only a small fraction of the possible sentences. However, we observe that the words and patterns used in the most frequent first-sentences reflect those found throughout the training data, leading to the Greedy Heuristic:

Count	Sentence	Count	Sentence
33,678	No.	11,184	Nope.
26,143	Yes.	10,364	Wrong.
23,334	Exactly.	10,128	What?
20,718	Really?	9,691	Yeah.
19,239	I agree.	8,857	?
18,122	Agreed.	8,734	I disagree.
12,070	Huh?	8,481	Not really.
11,993	Indeed.	8,383	Right.
11,812	Wow.	7,984	Yep.
11,577	Why?	7,318	True.

Table 2. Some of the most common full first-sentences found in the Slashdot training dataset.

> First-sentences that appear most frequently contain the same sentiment-bearing words and patterns that most frequently appear in the larger set of distinct first-sentences.

For example, if the exact sentence "I agree." appears more often than the exact sentence "I concur." then we would expect longer, less frequent sentence variations that contain "agree" to also appear more often than those that contain "concur."

Applying the Greedy Heuristic enables the user of the interactive decision tree to prioritize which patterns to encode in the regular expressions that splits the nodes. In our visualization of the data in each node, the most frequent first-sentences are displayed sorted in descending order, similar to Table 2. Intuitively, the most frequent sentences tend to be short, which makes the table-based visualization easy to examine and the sentences easy to label. In this way, the Greedy Heuristic provides guidance that will keep users working along the same path as they split tree nodes.

The interactive decision tree is a helpful tool for exploring the dataset. This is particularly important as the user tries to determine the best level of generalization to encode in each regular expression. For example, the regular expression "I agree" is too specific to capture such sentences as, "I definitely agree." Likewise, the regular expression "agree" is very general, and will capture such sentences as, "I disagree." or "They formed an agreement." The easiest approach is to choose the more general regular expression and then refine that branch with further splits, if needed. For instance, after splitting on "agree," the user would inspect the results to see how well it worked. If needed, another split could separate out sentences that contain "disagree" or "don't agree".

The Greedy Heuristic helps the user know when to stop training the decision tree. Our rule of thumb is to stop split-

[3] http://www.mturk.com

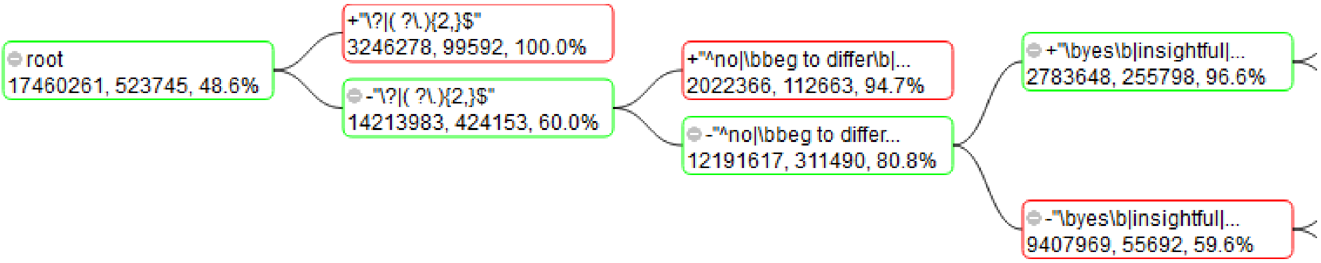

Figure 1. Interactive Decision Tree: Red nodes are negative, and green nodes are positive. The top row of text in each node contains the regular expression that led to its creation. A plus sign in front means the regular expression was found, and a minus sign means it was not found. The bottom row of text contains the total number of sentences in that node, the total number of labeled sentences, and the percentage of labeled sentences that match the majority class. Clicking on a node will display the most frequent sentences in the node, along with links for labeling.

ting a node once the most frequent 100 sentences all have the same polarity. However, there is generally a single remaining leaf node that contains the unclassified sentences. These sentences will be marked as "neutral" by the classifier.

6 Results

6.1 Experiment Design

We divided the data according to the date the news story first appeared on Slashdot's home page. Training data covers July 1, 2002 to December 31, 2011. Development data covers January 1, 2012 to June 30, 2012. Test data includes 5,000 randomly sampled replies from news stories between July 1, 2012 and December 31, 2012. Ten of those replies were removed, however, because they contained nothing but quoted text. Experiments were conducted using both the full replies and only the first-sentences (as defined in 4.1) in the test dataset. We separated the test and training data according to time instead of using random sampling in order to avoid look-ahead bias. As response patterns change over time, we didn't want patterns from the future unfairly influencing the training.

Amazon's Mechanical Turk provides a convenient way to label test data. We used the default template they provide for sentiment analysis tasks. Three separate workers labeled the full text of each of the 4,990 test replies on a 5-point scale from -2 to 2. Amazon provided a weighted average score for each "HIT", combining the three sentiment scores into just one. We used the sign of their weighted average score as the class label (positive, negative, or neutral). We refer to this source of labels as "mturk-fulltext." We repeated this full process with just the first sentence of each reply. This source of labels is called "mturk-first-sent." Separately, we also labeled each of the 4,990 first-sentences ourselves (called "authors-first-sent").

6.2 Baselines

We have not found another system trained for the task of feedback-sentiment classification. In the absence of annotated data, we cannot train and evaluate state-of-the-art supervised learning algorithms. Instead, we compare against common solutions used by practitioners: publicly available general-purpose sentiment classifiers.

Many sentiment analysis approaches use a bag-of-words representation, where the valence of individual words is aggregated to find the overall document sentiment [14]. To implement this approach as our first baseline, we use the list of positive and negative opinion bearing words provided by Hu and Liu [8]. We assign a polarity to each sentence according to whether it has more positive or negative words. Ties and sentences without positive or negative words are classified as neutral. We refer to this baseline as "lex-first-sent" when applied to just the first sentence, and "lex-fulltext" when applied to the full text of the reply.

Our second baseline is the OpinionFinder 2.0 polarity classifier [20].[4] This system includes a preliminary subjectivity classification, followed by the identification of contextual polarity. Again, we ran the classifier on both first-sentences and on the full text of each reply, which we refer to as "opfin-first-sent" and "opfin-fulltext."

Our third baseline is the Recursive Neural Tensor Network (RNTN) sentiment classifier by Socher et al. [17]. At the time of this writing, it is the most accurate sentiment classifier for movie reviews. Their system was trained on the Stanford Sentiment Treebank dataset.[5] The words and

[4] http://mpqa.cs.pitt.edu/opinionfinder/opinionfinder_2/

[5] http://www-nlp.stanford.edu/sentiment/treebank.html

phrases in the treebank were annotated for sentiment individually by Mechanical Turk workers. So, even though the training text comes from movie reviews, we believe the system to be of wider use.

The creators provide a convenient web interface to a live demonstration.[6] Their system is designed to analyze sentiment at the sentence level, so we submitted our test set of 4,990 first-sentences for analysis. In some cases, their system separated one of our sentences into two sentences. For those, we kept only the sentiment of the first. We call these labels "RNTN-first-sent."

6.3 Decision Tree Classifier

We ran our trained classifier on the test set of 4,990 first-sentences. These labels are denoted "dtree-first-sent." We also compare classification accuracy by applying the single-sentence classifier to the full text of the replies. To do this, we checked (in order) the first-sentence in a reply, followed by any sentences that followed quoted text. We took the first non-neutral label we found as the label for the whole reply. After that, if the label was still neutral, we ran the classifier on each of the other sentences (in order) until we found a non-neutral label. So, only if all sentences were neutral would the overall reply be labeled neutral. This set of labels we call "dtree-fulltext." As seen in Table 3, dtree-fulltext has a much smaller neutral class proportion than dtree-first-sent (resulting in higher recall on the positive and negative classes). In fact, all models that considered the full text of the reply had smaller neutral proportions than those that only considered the first sentence.

6.4 Results

Table 4 shows the accuracy of each classifier when compared with the mturk-fulltext gold standard. Here, dtree-fulltext outperforms the other models in all three metrics. When evaluated against the authors' annotations as the gold standard (Table 5), the dtree-first-sent model is the best performer.

Sentiment analysis is a difficult task, in general. It is common for three-class sentiment models to have accuracies under 60% [19]. Even people have trouble agreeing on the sentiment of short pieces of text. Table 5 shows that the authors' first-sentence annotations agree with the first-sentence annotations from the Mechanical Turk only 59% of the time.

6.5 Validation of the Heuristics

The First-Sentence Heuristic can be validated independent of the complete classifier. To do this, we separated

Label Source	Neg	Neu	Pos
authors-first-sent	49%	34%	17%
mturk-first-sent	48%	27%	25%
mturk-fulltext	53%	28%	19%
lex-first-sent	24%	53%	23%
lex-fulltext	35%	28%	38%
opfin-first-sent	17%	70%	12%
opfin-fulltext	39%	37%	24%
RNTN-first-sent	51%	36%	14%
dtree-first-sent	35%	48%	17%
dtree-fulltext	52%	22%	25%

Table 3. Class proportions of the annotations and of the models when applied to the test data.

gold standard: mturk-fulltext

Label Source	Acc	Pos F1	Neg F1
authors-first-sent	58%	0.46	0.69
mturk-first-sent	54%	0.45	0.67
lex-first-sent	38%	0.30	0.42
lex-fulltext	41%	0.33	0.52
opfin-first-sent	35%	0.22	0.31
opfin-fulltext	43%	0.27	0.53
RNTN-first-sent	44%	0.29	0.56
dtree-first-sent	45%	0.35	0.52
dtree-fulltext	49%	0.37	0.62

Table 4. Model accuracy, positive class F1-score, and negative class F1-score, when compared against mturk-fulltext.

gold standard: authors-first-sent

Label Source	Acc	Pos F1	Neg F1
mturk-first-sent	59%	0.51	0.70
mturk-fulltext	58%	0.46	0.69
lex-first-sent	38%	0.25	0.39
lex-fulltext	36%	0.26	0.44
opfin-first-sent	37%	0.19	0.30
opfin-fulltext	40%	0.24	0.47
RNTN-first-sent	42%	0.29	0.53
dtree-first-sent	57%	0.53	0.59
dtree-fulltext	54%	0.47	0.63

Table 5. Model accuracy, positive class F1-score, and negative class F1-score, when compared against authors-first-sent.

[6] http://nlp.stanford.edu:8080/sentiment/rntnDemo.html

gold standard: mturk-fulltext

Label Source	Single-Sentence Accuracy	Multi-Sentence Accuracy
mturk-first-sent	58%	53%
authors-first-sent	58%	58%

Table 6. First-sentence annotation accuracy with the test dataset separated into single-sentence replies versus replies with multiple sentences.

the 4,990 test replies into two sets: 1,090 single-sentence replies, and 3,900 replies with multiple sentences. Table 6 shows the accuracy (versus mturk-fulltext) of each first-sentence annotation source when separated this way. If the heuristic was a poor one, meaning feedback-sentiment might appear anywhere in a reply, then the multi-sentence accuracy would be significantly worse than the single-sentence accuracy. The results show otherwise. In fact, the agreement between authors-first-sent and mturk-fulltext is unchanged (at 58%) even when the Mechanical Turk workers saw more sentences than the authors did. This table shows that looking past the first sentence of a reply is not particularly helpful when searching for feedback-sentiment.

Note that Table 6 shows once more the difficulty of identifying sentiment. For single-sentence replies, the text shown to the Mechanical Turk workers in the two batches was identical. However, instead of finding 100% agreement between mturk-first-sent and mturk-fulltext, we find only 58%. This is consistent with the level of agreement (59%, Table 5) between the authors and the Mechanical Turk workers.

The Greedy Heuristic is seen to be helpful when training with our interactive decision tree. Each time a user splits a node with a regular expression, the new nodes display the number of matching sentences. As we trained our classifier, we found that the most frequent sentences in a node did contain the most common response patterns found throughout the data. Specific sentences that appeared only a few hundred times often contained general response patterns that appeared hundreds of thousands of times throughout the training data. In this way, each of the first few nodes were important contributions to the classifier. Our finished classifier only has 21 nodes (10 regular expressions), although we did combine many class-equivalent nodes together to make it run more quickly. This relatively small set of response patterns was able to classify with high precision 57% of the sentences in the training data. The remainder were assigned the class of neutral.

Note that even in a thoroughly trained tree, Table 3 shows that approximately 30% of sentences actually are neutral and should remain in the neutral leaf node. Therefore, the misclassified (polarized) sentences in the neutral node do not make up a significant fraction of the total. The decision tree can be further refined, but the improvement in accuracy diminishes.

6.6 Classifier Features

The interactive decision tree is designed for finding the most discriminative features for feedback-sentiment classification. Each dataset will have its own prominent features, which the decision tree is expected to make plain. Many of the features we used in our classifier were common words or phrases. Other features, which we present below, were helpful in classifying millions of sentences, and may be useful in forums other than Slashdot:

First-sentences that start with the letters “no” (case-insensitive) have negative sentiment with high precision. Some examples of common words that begin first-sentences include, “No,” “Nope,” “Not,” “Nonsense,” and “Nothing.”

Certain punctuation marks such as “?!?” and “...” at the end of a first-sentence indicate negative sentiment with high precision.[7] In Slashdot, when the first-sentence is a question, the author of the reply is generally questioning the parent comment rather than genuinely seeking an answer.

Yelling through the use of all-caps indicates negative sentiment. For this feature we ran a separate script that gave us the most common acronyms, which we included as exceptions in an all-caps regular expression in the decision tree. Specifically, for every word of three or more letters, we calculated the fraction of times it occurred in the training data in all-caps. We discarded rare words and sorted the results. We kept all but a few of the top 300 words as acronyms that do not carry with them any sentiment. The 300th acronym appeared fully capitalized only 15% of the time, so we felt confident we had captured the bulk of the common genuine acronyms.

Profanity in the first-sentence is generally a negative indicator, while smiley emoticons are positive. Sentences that end in an exclamation mark are generally positive.

Our trained classifier, which incorporates all these features, will be made available on the authors’ website. It is a simple Python function which takes a sentence as input, and returns the feedback-sentiment: positive, negative, or neutral.

[7] Represented by the following regular expression in our trained classifier: `\?|( ?\.){2,}$`

7 Conclusion

We introduced the concept of feedback-sentiment, which is the feeling a forum member expresses in reply to the author of a comment. We showed in the First-Sentence Heuristic that the feedback-sentiment is generally expressed at the beginning of a forum reply. The modified interactive decision tree we developed uses this heuristic, along with the Greedy Heuristic, to enable a classifier to be trained without prior labeled data. At a cost of only a few hours' training time, our approach outperforms three publicly available general-purpose sentiment classifiers on feedback-sentiment classification. Our resulting classifier outperforms for both full replies and first-sentences, evaluated against both the Mechanical Turk gold standard and the authors' annotations. Future work includes preparing sufficient training data to enable state-of-the-art supervised learning for feedback-sentiment.

Acknowledgments

This work is supported by Chevron U.S.A. Inc. under the joint project, Center for Interactive Smart Oilfield Technologies (CiSoft), at the University of Southern California.

References

[1] C. Danescu-Niculescu-Mizil, M. Sudhof, D. Jurafsky, J. Leskovec, and C. Potts. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, 2013.

[2] M. Galley, K. McKeown, J. Hirschberg, and E. Shriberg. Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 669, 2004.

[3] S. Germesin and T. Wilson. Agreement detection in multiparty conversation. In *Proceedings of the 2009 international conference on Multimodal interfaces*, pages 7–14, 2009.

[4] S. Hahn, R. Ladner, and M. Ostendorf. Agreement/disagreement classification: Exploiting unlabeled data using contrast classifiers. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 53–56, 2006.

[5] A. Hassan, A. Abu-Jbara, and D. Radev. Detecting subgroups in online discussions by modeling positive and negative relations among participants. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 59–70. Association for Computational Linguistics, 2012.

[6] A. Hassan, V. Qazvinian, and D. Radev. What's with the attitude?: identifying sentences with attitude in online discussions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1245–1255. Association for Computational Linguistics, 2010.

[7] D. Hillard, M. Ostendorf, and E. Shriberg. Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003-short papers-Volume 2*, pages 34–36, 2003.

[8] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[9] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, and A. Stolcke. The ICSI meeting corpus. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages I–364, 2003.

[10] A. Kumar and N. Ahmad. ComEx miner: Expert mining in virtual communities. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 3(6), 2012.

[11] C. Lampe and P. Resnick. Slash (dot) and burn: distributed moderation in a large online conversation space. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 543–550, 2004.

[12] Y. Liu and G. Salvendy. Design and evaluation of visualization support to facilitate decision trees classification. *International journal of human-computer studies*, 65(2):95–110, 2007.

[13] C.-C. Musat, B. Faltings, and P. Rousille. Direct negative opinions in online discussions. *2013 International Conference on Social Computing (SocialCom)*, pages 142–147, 2013.

[14] B. Pang and L. Lee. Opinion mining and sentiment analysis. volume 2, pages 1–135. Now Publishers Inc., 2008.

[15] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 2010.

[16] E. Smirnova and K. Balog. A user-oriented model for expert finding. In *Advances in Information Retrieval*, pages 580–592. Springer, 2011.

[17] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, 2013.

[18] S. O. Sood, E. F. Churchill, and J. Antin. Automatic identification of personal insults on social news sites. *Journal of the American Society for Information Science and Technology*, 63(2):270–285, 2012.

[19] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan. A system for real-time twitter sentiment analysis of 2012 US presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, pages 115–120, 2012.

[20] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.