Classification of Usefulness in User-submitted Content Using Supervised Learning Algorithms

Ellis Weng, Tze Jian Chear
{ew82, tc262}@cornell.edu

Abstract

In this paper, we consider the problem of classifying online user-submitted content by usefulness. Rather than classifying comments based on the submitters' sentiments or the overall helpfulness as perceived by the community, the comments are classified as either useful or useless based how much the comment contributes to a discussion. Using comments from an online community, we find that machine learning techniques (Naive Bayes, Support Vector Machines, Logistic Regression, and Bayesian networks) can help identify useful comments. In using machine learning techniques, we obtain improvement over the baseline and accuracies that approach that of human annotators.

1. Introduction

Identifying the usefulness of user-contributed content is a broad task and is useful for many applications. For instance, identifying the most important comments in a domain can further an individual's understanding of particular topics, policies, ideas at first glance without having to sift through a large amount of comments. Also, moderators can use this classification to ensure the quality of contributions and to reduce the amount of flaming or spam. There are many useful online communities that can benefit from this type of research, and making a classifying for all these domains would be too broad, so our research will focus on a particular domain: online feedback to proposed regulations. U.S. Regulatory agencies are interested in and are required to get feedback from the public on proposed regulatory policies. The rule-writers must go through a great number of comments and identify comments that are important, novel, and substantive. Our research will help rule-writers find comments that are useful to them in the rule writing process: comments that provide evidence and factual assertions and novel ideas.

2. Related Work

Text classification is a well-known problem and has many applications. Spam classification is a perhaps the best example of text classification: Given an email can we determine if it is spam or ham. There are other classification problems in the field of natural language processing that are more related to our field of research. For example, there problems of genre tagging, author publisher tagging, native-language background. In our research we will adopt some of the ideas and concepts behind these problems, but these classifications are not entirely relevant to our task of identifying usefulness in user-submitted comments. There are a couple fields of research that are related to our classification problem.

2.1 Sentiment Analysis

The first field of research is sentiment analysis: The goal of sentiment analysis is to gain an understanding of what is expressed in text. The sentiment analysis problem that is the most closely to our classification task is semantic orientation classification (classifying opinions as either positive or negative). The goal of this problem is to try to find if an opinion has a positive or negative annotation, which can be further broken down into classification on a word phrase, sentence, or document level. There are machine learning algorithms that can help predict the sentiment of an opinion. In recent years, research has found machine learning algorithms that help with sentiment classification that have an accuracy between 75-92% depending on the data-set used (B. Pang, L. Lee; Kwon, Hovy, Zhou). We cannot directly apply this method to our problems; understanding whether someone favors or opposes a policy does not directly imply the usefulness of a comment because regulation writers are not only interested in the mere position of a person.

2.2 Claims and Justifications

The second field of research that relates to our study is classifying claims and finding justifications. This type of research deals with finding reason and justification behind an opinion holder's claims. There is very little research in this field than compared to that of sentiment analysis. In "Identifying Types of Claims in Online Customer Reviews", comments were categorized comments in online reviews as either a qualified claim or a bold claim (Shilpa Arora, Mahesh Joshi, and Carolyn Ros´e). An example of a qualified claim is "this camera is small enough to easily fit in a coat pocket"; whereas a bold claim is "this is a small camera." In using machine learning techniques, this study obtains accuracies of 68-72% which was only slightly above the original baseline of the majority (the baselines were 53% for one test set and 69% in another). Another similar study, "Automatic Identification of Pro and Con Reasons in Online Reviews", uses machine learning techniques to automatically detect pros and cons within online reviews. The accuracy in the end of this research was not too high, ranging from 63-77%. These studies show that these types of classification problems are rather difficult and not well-developed. However, understanding the ideas behind these papers will help us in devising a method to classify our comments as useful. In addition to looking for the claims and justifications, other major criteria that rule makers are looking for, which also determines usefulness in comments, are new ideas, relevant questions, and suggestions.

2.3 Helpfulness in Online Reviews

The final field of research is helpfulness classification in online product reviews. Some common examples of online review data-sets are the reviews on Amazon, Yelp, and TripAdvisor. Most research in this field deals with creating systems to return a set of reviews that are more helpful to other users, not necessarily classifying helpful reviews. Similar to classifying claims and justifications, this field of research is under-developed. Many of the systems proposed return comments that are on average 2-15% more useful than simple baselines (O'Mahony, M. P., and Smyth). However, this type of research is most similar to our problem and can provide insights as to what is useful in a comment and what affects the perceived usefulness of a comment. For example, there are many social factors affecting the perceived helpfulness of a comment--Even if the review is plagiarized, the content of the comment can be identical, the perceived helpfulness can be different if the review was posted by different users (C. Danescu-Niculescu-Mizil, G. Kossinets, J. Kleinberg, L. Lee). It is important to realize that this classification problem is dealing with subjective measures that we cannot always account for.

Although there has been much research in text classification and sentiment classification in online reviews, there is little research in the field of classification based on usefulness or helpfulness in user-submitted content. Even with previous research in helpfulness classification in online reviews, our problem is still different. All of this research is in the domain of product reviews; what information is useful for users to make a decision before purchasing an item. Most of the time, providing informational and anecdotal evidence regarding the product is enough to classify a review as helpful or not. Useful contribution in the electronic rule-making domain is more complicated. Although there is related work, our problem is still inherently different from previous applications of machine learning.

3. Useful Contribution in User Comments

In other fields of research on helpfulness, the scores for helpfulness are usually provided by other users in community. In prior research in this field, a review is perceived to be helpful if 75% of the other users ranked this comment as useful, or if the review is above 4 stars (O'Mahony, M. P.; Cunningham, P.; and Smyth). However, the definition of useful in our problem is not so well-defined.

We chose to work with online feedback in a government regulation community. What constitutes as useful in this domain is different from what constitutes as useful in order domains. For example, in product reviews a useful or helpful comment might be “This camera works great!” Other potential buyers of this product might find this useful in making a purchasing decision. On the other hand in providing feedback for rules and regulations a similar comment, “This regulation would be great”, would not be useful. This comment will not be useful to the rule-maker because the regulation is already proposed, and there is no additional material in this comment that contributes to the discussion of the rule.

The definition of a useful comment in this sense is particularly difficult to deal with. What is a “useful comment” to a rule maker? In order to code these comments as useful or useless (high-quality or low-quality), we had a discussion with the Cornell eRulemaking Initiative (CERI) in order to determine what constitutes as a high-quality comment. We determined that a high-quality comment should include one or more of the following attributes: useful facts, relevant questions, reason behind their thought, personal experiences or examples, suggestions for improvements of rules, and new ideas that are plausible. Of course, there is no concrete formula for coding even after defining these criteria. For example, a comment can be exhibit all these features but be slightly off-topic and goes against civil discourse, which ultimately does not contribute anything to the discussion.

4. Finding Useful Comments

In order to find the usefulness of comments in the e-Rulemaking domain, we first had to create a data set. First, we had to arrange meetings with CERI in order to discuss the definition of usefulness. After the criteria for a useful comment were determined as shown in A.1, we had to train annotators to manually annotate the comments on RegulationRoom.org. After the initial labeling, we resolved disagreements with the data and used this data set to train our machine learning models.

4.1 Coding

The comments used in this study were the comments on Airline Passenger Rights on RegulationRoom.org. The annotators, law students in an eRulemaking course, who coded this data had more expertise in online e-Rulemaking than an average individuals. The comments were assigned to twelve different law students; every comment was assigned to as least two coders. There were a total of 910 comments. This amount of comments is reasonable compared to similar classification problems: in “An Assessment of Machine Learning Techniques for Review Recommendation” Mahony showed a relationship between accuracy and training set size; increasing the training set from 100 to 1000 reviews showed an improved of about 10% in accuracy, whereas increasing the traning set from 1000 to 14,000, shows an increase of around 5% in accuracy (O’Mahony, M. P.; Cunningham, P.; and Smyth, B. 2009).

To assess the agreement between two coders in coding the same comment, we calculated the accuracy of coding. The inter-annotator coding agreement between these students was reasonable. The average inter-annotator annotation agreement between 6 groups was 75.73%. We then computed the Cohen’s Kappa coefficient that is used for assessing rater agreement as a chance-corrected measure (Cohen, J): Kappa= P(A)−P(E)/1− P(E). The Kappa value for this coding task was 24.4%, which indicates a slight to fair agreement between coders. This is far lower than the initial 75.73% accuracy that we determined because the probability of coding agreement by chance is 66%, so by chance coders were more likely to agree if they selected “useful” as a tag. Out of the 910 comments, 66% are coded as useful.

After the initial coding, the annotators were asked to resolve the conflicts by discussing why they coded a comment as either useful or useless. The final label for a comment was determined after consulting the coding guidelines provided.

4.2 Features

Feature selection is a core task in this project. Not only generating the features, but determining what they are is also a difficult task. In deciding which features to use, we relied on the category guidelines for coding in A.1 because by definition a useful comment should exhibit these features. Our next approach was to model our features on previous research that deals with similar classification problems.

Modeling each of the categories in A.1 is a very difficult task. Trying to identify these qualities in a comment is a separate project in itself. There is recent research that tries to accomplish analogous tasks. For example in “Identifying and Classifying Subjective Claims”, the subjective claim is identified in e-Rulemaking comments (Kwon, Hovy, Zhou). Identifying the main claim of a comment is analogous to finding our “suggestion” and “new idea” category in our coding scheme. The accuracy in identifying these claims is 52%. Another example of identifying one of our categories is the research of Kim and Hovy in “Automatic Identification of Pro and Con Reasons in Online Reviews”. Identifying pros and cons is analogous to identifying reasons. The accuracy for this paper is slightly higher with 66%. It is difficult to model these predefined categories.

4.2.1 Linguistic Inquiry and Word Count

The best approach that we found to try to model these pre-defined categories is in the realm of linguistic word count. According to J. W. Pennebaker and M. E. Francis, there are cognitive processes that cause people to choose certain words to use. In using Pennebaker’s technique of Linguistic Inquiry and Word Count, we manage to extract several features from our comments that are related to the categories that we predefined. For example, the “Cognitive Mechanism” category has words that related to our categories of “suggestion”, “reason”, and “new idea”. Some words in this category include “because”, “should”, “ought”, “consider”. We also used other categories such as “Positive Emotion” and “Negative Emotion” in order to try to identify opinions.

4.2.2 Unigram and Bigram

It is common to see n-grams be used as features in the natural language processing setting. A unigram (1-gram) count is simply the count that a particular word occurs in the comment. Similarly, a bigram count is a count that a particular sequence of two words appears in the comment, such as “airline passengers”. This is common practice because having a set of features that represent the actual word counts of the important words makes will help better define our comments. In our approach though, we limited our unigrams and bigrams to ones that occur more than 5 times in the corpus. Also, initially we removed all the stop words (such as “a”, “the”, “is”, “in”) from the unigrams and bigrams.

4.2.3 Readability Scores

After going through the comments, we determined that certain users have more sophisticated language than others, and it affected our perception of how useful a comment is. It makes sense that people who put more thought into their writing should have more coherent writing overall. See appendix A.2 for description of readability score definitions.

4.2.4 Punctuation

In adding punctuation to our features, we can better model the “Revelent Question” category. Also, other punctuation is useful for analysis. Usually useful comments have proper punctuation and more sophisticated punctuations such as colons, hyphens, and dashes as opposed to just periods.

4.2.5 Sentence, Word, Syllable Length

On average, the useful comments tend to be longer. Useful contributions tend to have one or more of the predefined categories, and it takes more thought and words to establish a good argument, question, reason, suggestion, etc. We believe that adding in simple measures such as sentence length will improve our performance.

5. Bayesian Networks

Because of our predefined categories and diverse features, we also tried a graphical model in order to try to model the relationships between these categories. With such diverse features, we have two models in mind.

5.4.1 Model 1 – Factual / Experience

In section A.1, we specified a set of guidelines to be used by labelers in judging the usefulness of a comment. Chief amongst those criteria are facts and experiences. Often times, a commenter must provide evidence to buttress her claim or suggestion in response to the topic; and evidence may come in the form of factual citations (such as figures, statistics) or anecdotal encounters. With these assumptions, we propose a Bayesian network that depicts the dependencies as shown in Figure 1 (Appendix). Useful/useless is modeled as a binomial distribution, while the others are each modeled as a Gaussian distribution.

5.4.2 Model 2 – Paragraph length / Writing quality

In order for a comment to be useful, more often than not it would have to include a fair amount of information, and this correlates positively with the number of words used. It is also fair to assume that the higher quality a comment's writing is, the more useful it is. We use readability indices to indicate a comment's quality. We propose the use of the scoring formulae as specified in A.2 as feature variables in our model. With these assumptions, our model is as shown in figure 2 (Appendix).

We use WinMine Toolkit to construct our models and to run our training set on the model. From our training set and with our models as constraints, WinMine infers another Bayesian network that is closest to our models. Variables may be dropped from our model because the training set does not show any significant dependency relationship. Note that comment_tag is a variable denoting a comment's usefulness. See appendix A.5 for the models.

6. Results

The data was split into two equal sets, the first 455 comments were used as a training set, and the next 455 comments were used as the test set. These sets had a similar distribution of useful and useless comments.

We tried many combinations of different training sets, feature sets, and algorithms. A chart of all the different runs can be found in the appendix. After much experimentation, Support Vector Machines and Naïve Bayes prove to be the best approach. Logistic Regression came as a close third with the highest accuracy of 69.01%. The Bayesian network approach was hard to compare to these algorithms because of the log score statistics, a separate discussion will be presented for this algorithm in 6.3.

The highest accuracies obtained are provided in the following chart:

Features Used	# Of Features	SVM	NB
Unigram, LIWC	1204	71.43%	70.77%

Unigram, Bigram, POS, LIWC	3178	67.03%	**71.21%**
Unigram, Custom LIWC	1157	**71.87%**	70.99%

6.1 General Behavior of Machine Learning Algorithms

In general, the support vector machines had a consistent behavior with all the different feature sets, the accuracies ranged from 65.93% - 71%. This model was superior to the other models. The Naïve Bayes model was a close second given the correct statistics. There was more variance in this model from as low as 52.25% - 71.21%. Logistic Regression was highly inefficient to train and was very sensitive to over fitting. Eliminating many features in order to use logistic regression was helpful: Logistic regression performed the best when there were only 50 features, at 69.01%.

6.2 Determining Features to Use

First we ran all the algorithms with all the different features that we proposed. We noticed several things; the scores were close to our baseline of the majority of 66% and the logistic regression model took a significant amount of time to train. We then started to experiment with different to see the effects on the models.

6.2.1 Grade Level, Punctuations, Sentence Length

These features were simple measures and after removing them, we realized that these were good features and were few enough so that there was no over fitting in our models. As a result, for all of the results presented in our appendix, we included using the features in all of our runs.

6.2.2 Unigrams

At first we eliminated all unigrams that were considered stop words, or functional words, such as “the”, “a”, “to”, “is”. We thought that these words were too common in language to display any significance. However, when we added these features, we obtained much higher accuracy rates. The stop words did not increase the number of features much so, which prevents over-fitting of the data. Also, sometimes the number of stop words is the only telling characteristic of a comment. There was an exception though—logistic regression. Logistic regression was very sensitive to over fitting, so when we removed the unigram features, we had a significantly lower amount of features, and the performance was better.

6.2.3 Bigrams and Part of Speech

In general, bigrams and part of speech did not help our performance by much. These are common features including in many natural language processing algorithms, but in our case, there was too much over-fitting. Every time we included these features in our test runs, the number of features more than doubled, and most of the time the algorithms suffered as a result.

6.2.4 Linguistic Inquiry and Word Count

LIWC features proved to be helpful in our classification problem. Removing these features lowered our accuracy. These features were particularly useful in helping to classify useless comments; there were fewer errors in classifying useless comments as useful. Initially, we included all of the features provided with LIWC. However, we thought that some of these categories might have been not so useful in our model, so we only used the important categories that matched our model: “I”, “You”, “Cognitive Mechanisms”, “Positive Emotion”, “Negative Emotion”, “Swear”, “Perception”.

6.3 Cross Validation

Initially we were not running cross-validation tests in our training models. After running a ten-fold cross validation on our training set, we discovered that our cross-validation error was significantly lower than our testing error. This implied that our data was not even distributed across the test and training set. In all of our confusion matrices, we found that most of our errors occurred when we classified useless comments as useful. This was not a problem in our cross-validation confusion matrices in our cross-validation, which implies that there were not enough useless comments in our original training set. After discovering this, we randomly distributed the comments into the training and test set instead of taking the first half of the comments as the training data. By doing this, we significantly improved our performance—the results shown in the beginning of section 6 show our new training set, while the results in our appendix represent our old training set before cross-validation.

6.4 Bayesian Network Analysis

We compare the *log score accuracy** of each model in the following table:

Model	Log score
Factual/Experience	-0.624089
Paragraph length/Writing Quality	-0.798142
WinMine Inferred	-0.989617

**Log score is defined as the average of the log posterior probability of the value for each output variable, given the values of all other variables. Thus, the closer this score is to zero, the higher the joint probabilities are. We see that our first model is the best amongst the three by this measure.*

We justify this result by noting that in judging a comment (which is a response to a regulation issue), how a commenter feels about an issue is important in establishing empathy amongst readers to keep a discussion going, and therefore ‘Feel’ is considered useful. ‘Number’ was dropped by WinMine from the model because most commenters do not quote statistics result in buttressing their response. ‘Insight’ introduces new ideas into a discussion, and is thus considered useful.

7. Discussion

The results produced by machine learning techniques show an improvement in comparison to the baseline of 66% accuracy. Having an improvement to 72% was an achievement because the human agreement for this annotation task was 75%. The low annotator agreement implies that this is a difficult task and that we should not expect to surpass the human agreement accuracy (Kwon, Hovy, Zhou). In terms of relative performance, the Support Vector Machine model did the best in terms of consistency and overall accuracy. Naïve Bayes also performed relatively well with the correct features, and Logistic Regression performed well when there were few number of features. We also contributed an original data set of annotated comments for e-Rulemaking. There was much work in creating this data set in terms of discussing what is useful for rule makers and in training annotators to get higher levels of inner-annotator agreement.

In further research, the first task should be to increase the size of the data set. Creating a data set this size was reasonable for a term project because there were no additional quality comments on RegulationRoom.org in Airline Passengers, and it was too difficult to explore other domains while finding more qualified individuals to annotate more comments. At the end of the project, we ran cross-validation on the entire training set on a SVM model, and the accuracy for this model was significantly higher than any of our models. After running this model 10% of the cross-validation error was reduced if we were to use the entire data set as a training set. This implies that more data would significantly improve our performance. In further research, finding ways to model the predefined categories should be explored. Finding ways to identify these categories was not within the scope of this course project. There is much new research that shows promise in exploring these fields, such as identifying the types of claims in online reviews and using machine learning techniques for review recommendation (Shilpa Arora, Mahesh Joshi, and Carolyn Ros´e. 2009; O'Mahony, M. P.; Cunningham, P.; and Smyth, B. 2009).

References

Anindya Ghose and Panagiotis G. Ipeirotis. Designing novel review ranking systems: predicting the helpfulness and impact of reviews. In ICEC, pages 303–310, 2007.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of EMNLP, 2002. Introduced polarity dataset v0.9.

Cohen, J. A Coefficient of Agreement for Nominal Scales. Education and Psychological Measurement. 43(6):37—46. 1960.

Cristian Danescu-Niculescu-Mizil, Gueorgi Kossinets, Jon Kleinberg, Lillian Lee. How opinions are received by online communities: A case study on Amazon.com helpfulness votes. Proceedings of WWW. 2009

C.-F. Hsu, E. Khabiri, J. Caverlee, Ranking comments on the social web, in: Proceedings of the 2009 IEEE International Conference on Social Computing (SocialCom-09), Vancouver, Canada, 2009, pp. 90–97

Chickering, David Maxwell. The WinMine Toolkit. Microsoft. 2002.

J. W. Pennebaker and M. E. Francis, Linguistic inquiry and word count: LIWC 2001, Erlbaum Publishers, Mahwah: NJ, 2001.

Kim, S-M. and E.H. Hovy. 2006. Automatic Identification of Pro and Con Reasons in Online Reviews. Poster. Companion Proceedings of the conference of the ACL. Sydney, Australia.

Kwon, N., S. Shulman, and E.H. Hovy. 2006. Multidimensional Text Analysis for eRulemaking. Proceedings of the National Conference on Digital Government. San Diego, CA.

Namhee Kwon, Eduard Hovy & Stuart Shulman. Identifying and Classifying Subjective Claims. May 2007.

O'Mahony, M. P., and Smyth, B. 2009. A classification- based review recommender. Knowledge-Based Systems doi:10.1016/j.knosys.2009.11.004.

O'Mahony, M. P.; Cunningham, P.; and Smyth, B. 2009. An assessment of machine learning techniques for review recommendation. In Proceedings of the 20th Irish Conference on Artificial Intelligence and Cognitive Science.

Rose, C. P., Wang, Y.C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., Fischer, F. (In Press). Analyzing Collaborative Learning Processes Automatically: Exploiting the Advances of Computational Linguistics in Computer-Supported Collaborative Learning , International Journal of Computer Supported Collaborative Learning

Shilpa Arora, Mahesh Joshi, and Carolyn Ros´e. 2009. Identifying Types of Claims in Online Customer Reviews. In Proceedings of NAACL 2009.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many rel- evant features. In almost all of