A classification model for identifying possibly favored comments on YouTube

Jihong Ju 4518454
Delft University of Delft

The social video sharing platform YouTube generates not only hundreds of hours of videos every minutes but also millions of comments for these videos every day. These comments involve good quality remarks for the videos as well as irrelevant ones and spam. In order to display these comments properly, recommend good comments and filter irrelevant ones, we propose a classification method applying machine learning technique together with semantic enrichment and sentiment analysis. We judge the relevance of features extracted from semantic enrichment and sentiment analysis with p-test and predict comment possibly favored by users based on relevant features.

Categories and Subject Descriptors:

Additional Key Words and Phrases: semantic enrichment, sentiment analysis, classification

1. INTRODUCTION

Comment has become one of the most important methods for users to interact with others among online communities. People exchange their perspectives and affection by adding comments to the content of interest and others are able to 'like' the comments. The current strategy of ranking these comments for most online communities like twitter, Facebook and YouTube is counting the number of 'thumbs up' or 'likes' for the comments and sorting them in descending order. This method is simple but effective when the number of comments is small. However, the scale of the number of comments for one particular video on YouTube can easily reach tens of or even hundreds of thousands nowadays. Great amounts of comments has not receive any 'likes' not because they are not favored by users but because they are submerged in the spam comments. Identifying the relevant comments that could be possibly favored by users then becomes an inevitable problem. The primary aim of this paper is to develop a effective classification system identifying good quality comments for the YouTube videos. The supervised classification system requires labeled examples which we want them to be prepared automatically. One natural and straightforward labeling method for the comments is making use of the status that the comment was 'liked' by the other users or not. If a comment was liked by someone else, it indicates that this comment was supported by another user because it is the one in good quality.

In this article, we present a semantic and sentiment analysis based model to classify possibly favored comments from irrelevant and spam on the social video sharing platform YouTube. Our model combines and extends entity recognition and sentiment classification to describe the characteristic of the comments so that we are able to classify good quality comments using machine learning techniques. Before we dive into the practical field of works, the following research questions need to be asked and answers to these questions could help us to formulate the problem we are going to solve and come up with feasible solutions.

(1) How can we model or describe the semantics in comments?
(2) Is there a connection between semantics and comment ratings?
(3) Is there a connection between sentiment and comment ratings?
(4) Can we predict the community feedback for comments with supervised learning?

These are the questions led to this project that we discuss about in the rest of this report. The organization of the report is as follows: Section 2 summaries the related works in this area. Semantic enrichment and sentiment analysis are briefly introduced in Section 3 and 4 respectively. A classification model based on semantic enrichment and sentiment analysis is proposed in Section 5. The experimental configuration and results are discussed in Section 6. Conclusion is given at the end of the report.

2. RELATED WORK

Ahmad Ammari and his colleagues have developed a classification model for the spam comments under YouTube videos based on relevance judgment. [1] They applied the Bag-of-Words strategy to analyze the relevance of comment by comparing the occurrence and frequency of highly relevant words with the related vocabulary collections. Stefan Siersdorfer et al. studied the influence of sentiment expressions in comments on the comment ratings with more than 6 million comments on 67,000 YouTube videos. [2] They analyzed dependencies between comments, views, comment ratings and topic categories and develop a supervised classification system, identifying the relevant comments that could be possibly favored by users. Bo Pang et al. (2005) introduced a sentiment analysis methods with the help of machine learning techniques, Naive Bayes Classifier, using movie reviews as data. We, in this paper, kept his sentiment analysis methods and attempt to find the correlation between sentiment and ratings of comments. Mike Thelwall and Pardeep Sud presented a descriptive statistics about YouTube comments in terms of ages, sexuality, regions and so on and so forth.

3. SEMANTIC ENRICHMENT

In order to study the relevance of comments, the plain text extracted from comments is far from sufficient to provide necessary descriptions. Semantic enrichment is required to help convey the meaning of the plain text for the comments. Many different methodologies for semantic enrichment have been proposed in the past three decades. The main idea of these methodologies is to add a layer of topical metadata to content so that machines can make sense of it and build connections to it. The topical metadata can be any kinds of commonly published contents such as people, events, blog posts, reviews and tags in the web pages. For example, semantic markup uses machine-readable data like *microdata*, *microformats*

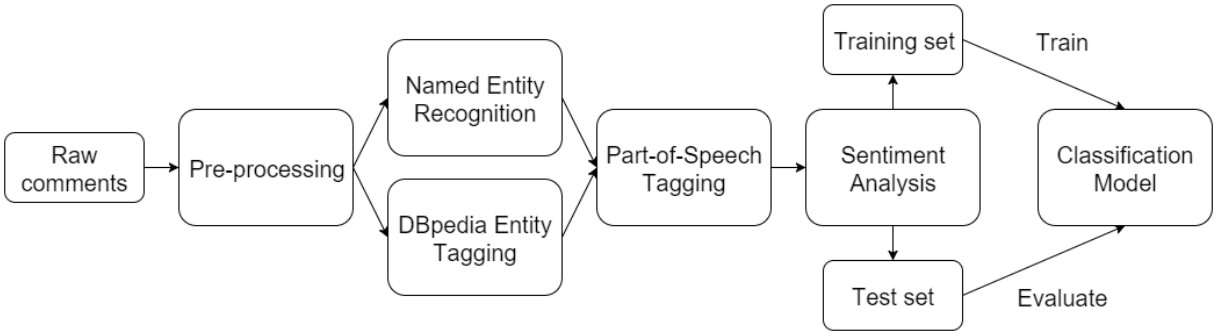

Fig. 1. Process Flow

and *RDFa* to reinforce the semantics, or meaning, of the web information. Other resources like wikipedia and WordNet can also be linked to the entities in text to enrich the semantics of the sentences, paragraphs or articles.

For example, in sentence

> The Graham Norton Show in my opinion is the best host show.

the entity *The Graham Norton Show* was mapped to its corresponding wikipedia item by looking up the DBpedia dataset. [6]

An alternative method for semantic enrichment is *Named Entity Recognition (NER)*. Named entities are phrases that contain people, organizations and locations. Named entity recognition is basically a classification problem. Classifiers like decision tree is implemented to learn from training examples and predict the class of the other words. The state-of-art named entity recognition system like Stanford NER can easily reach a relatively high accuracy. [5] Named entities provide basic information in a sentence in general. We want to study whether the number of named entity in comments influences the comment ratings or not.

An example of Named Entity Recognition using Stanford English 3 classes classifier is as follows:

> It has been 30 years and *Princess Leia* is still there.

where Prince Leia was tagged as *People*.

Part of Speech (PoS) tagging, also known as grammatical tagging, is the process of marking up the grammatical function (part of speech) of a word or phrase in its context. For example, the sentence

> This video is just amazing.

becomes

> DT/This NN/video VBZ/is RB/just JJ/amazing ./.

after part-of-speech tagging, where *DT* stands for *Determiner*, *NN* represents *Singular noun*, *VBZ* denotes *Verb, 3rd ps. sing. present*, *RB* is short for *Adverb*, *JJ* is *Adjective* and . is apparently period.[7] The English Part-of-Speech tagging is basically based on *The Brown Corpus* developed at Brown University by Henry Kuera and W. Nelson Francis, consists of about 1,000,000 words of running English prose text.

However, each word in English may have various PoS markers depending on the contexts. For instance, the word "love" has meanings and roles in sentence "I love cat" (verb) and "That is a love story" (noun). Many methods, like hidden Markov Model, unsupervised learning and so on and so forth, have been developed to help distinguish the different roles of words and phrases in various contexts.

4. SENTIMENT ANALYSIS

Sentiment analysis, determining whether a piece of text is positive, neutral or negative, derives the attitude of the author about the content. For example, the fans of a video on YouTube may show their positive opinions in their comments unconsciously whilst the adversaries tend to present their negative attitudes via comments. The positive and negative sentiment compose the polarized sentiment, and neutral is then categorized as non-polarized one. CG Lord and his colleagues found that people who hold strong opinions are likely to accept "confirming" evidence and reject "disconfirming" evidence (1979). [8] Users holding positive attitude about the video tends to accept positive statements about the video whereas users with negative are likely to accept negative statements. People who don't hold opinions in advance are likely to keep their mind open for all kinds of opinions. Therefore, the polarized attitudes are more likely to gain agreement with other users than the non-polarized counterpart. Sentiment analysis helps mining the polarity of the attitude from comments contributor so that we are able to study the effort of sentiment polarity on the comment ratings.

The fundamental technique for opinion mining is classification because sentiment analysis can be formulated as an opinion polarity classification problem. Given labeled textual examples, the classification model is able to find the possibility of an unlabeled piece of writing being positive or negative. In practice, some models also contain neutral class. Bayes model, maximum entropy classification and many other models have been studied to apply on sentiment analysis. [9; 10] The performance of the classification does not only depend on the model being chosen but also the training examples being collected. First of all, the topic and context of training examples should be similar with the unlabeled texts. Models trained with movie reviews are not necessarily as accurate on tweets as on movie reviews. Second, The training set should be sufficiently large and diverse in order to avoid overfitting. If the training set is small or not, a large bias of classification result will arise. The classification model will then not be able to predict accurately with target content. Finally, the plain text itself is not able to "teach" the model properly. The key concepts or sentimental phrases should be extracted from the sentences so that the frequency of these concepts or phrases (features) can be "learned" by the model. One common method is to use a so-called *standard bag-of-feature framework*. Let $(f_1, f_2, ...f_m)$ be m features that might be occurred in the textual document. Each document is then represented as a feature vector $(n_1(d), n_2(d), ..., n_m(d))$. The classification models take fea-

ture vectors as input and predict the polarity and corresponding probability. [9]

Again, we use the sentence *"This video is just amazing."* as an example, where the sentiment of this sentence is classified as positive with probability $neutral : 0.2$, $pos : 0.6$, $neg : 0.4$. To obtain the result above, Naive Bayes Classifier was trained with movie reviews. [11] The classification model was accessed via sentiment analysis API from the website *text-processing.com* developed by Jacob with *Python*'s natural language processing toolkit library *nltk*.

5. THE PROPOSED CLASSIFICATION MODEL

In order to develop a well-formulated classification model, the input patterns that the model can learn is required to be clear and well-defined. We have discussed methods of semantic enrichment and sentiment analysis in Section 3 and Section 4 respectively. These methods are able to help extract quantitative features, which are compatible with learning methods, from raw comments. We are going to discuss how can we extract features based on techniques introduced above in this section (Section 5.3 and Section 5.4). The relevance of these features with comment ratings will be judged in the next section. We also describe the development of a classification model with relevant features in Section 5.5.

5.1 Methodology

Figure 1 shows a flowchart representing the methodology of building a classification model from raw comments based on semantic enrichment and sentiment analysis. Totally 1525 comments from ten videos within the topic of talk show and animations were collected. The step-by-step process is described as follows:

Step 1 Pre-process the collected comments to ensure that they are shown in clean and clear style.
Step 2 Apply semantic enrichment methods on pre-processed comments.
Step 3 Implement sentiment analysis on pre-processed comments.
Step 4 Extract relevant features from Step 2 and Step 3 and build the dataset using these features.
Step 5 Split the whole dataset into two parts, training set for training the classification model and test set for evaluating the model.

5.2 Pre-Processing

Data cleaning is one of the most important processes when we are dealing with content from online communities because people usually do not care much about spelling and prefer to use slang when appropriate. Multiple formats in comments, such as expressions and html links, are also need to be handled properly. The cleaning details in our project basically contains:

(1) Escaping HTML characters
(2) Removal of Expressions
(3) Decoding data translating complex symbols to simple ones
(4) Apostrophe Lookup
(5) Removal of Stop-words
(6) Removal of Punctuations
(7) Split Attached Words
(8) Slangs Lookup
(9) Standardizing misspellings
(10) Stemming words

The processes above were implemented with related Python libraries, like HTMLParser and nltk, if there are any. The apostrophe corpus were constructed based on Cambridge dictionary and the slang list were constructed based on summaries of popular slang online. [12]

5.3 Semantic Enrichment

The semantic enrichment methods involved in this project contain DBpedia annotations lookup, named entity recognition and Part-of-Speech tagging. They were all implemented with corresponding libraries and APIs. For instance, the python library *pyspotlight* provides API for DBpedia Spotlight which help map words in text to their corresponding Wikipedia items. Similarly, the library *nltk* provides method to connection to the popular named entity recognizer *StanfordNER*. *nltk* itself contains method implementing Part-of-Speech tagging with acceptable accuracy.

After applying these methods, we are able to extract quantitative features from enriched comments. Totally five different features, varying from number of DBpedia annotations, number of named entities, number of different PoS tags to number of words, were extracted. Compared with word frequency vectors in the Bag-of-Words strategy, these features can be easily judged whether they are relevant to the labels or not.

5.4 Sentiment Analysis

Usually sentiment analysis performs better if the polarity classification model is trained with specifically designed corpus. For example, the corpus for sentiment analysis on tweets should be constructed with tweets as well. We are going to apply sentiment analysis on YouTube comments in our project so that the corpus should also consist of YouTube comments. However, it is difficult to construct a well-formed YouTube comments corpus since YouTube limits the number of available comments for each video for individual developers. Thus, in our project, a Naive Bayes model trained by Kamal and Badruddin with movie reviews were used.[11] This model was accessed via the *nltk* library and API provided from the website *text-processing.com*. The outcome of the sentiment analysis is the sentiment polarity of the comments, positive, neutral or negative and the corresponding probabilities for each class. The probabilities are quantitative so that they can be used as the input of classifiers. The probability for negative and probability for positive sum up to 1 so that we can keep positive probability only for training the model.

5.5 Model Development

By applying semantic enrichment and sentiment analysis on pre-processed comments, we have collected six features containing four from semantic enrichment and two from sentiment analysis. Together with number of replies, which can be obtained directly by querying YouTube API, we have construct the data with seven features and 1525 labeled examples. The relevance of the features were judged using p-test and details are presented in Section 6 and only the relevant features are kept while the irrelevant feature are discarded. The problem is then formulated as a binary classification problem with two classes describing whether the comments were liked by the others or not. Class 1 represents comments being liked whilst class 0 stands for comments that have not been liked yet.

To develop a classification model, the dataset should be split into two sets. One is used for training the model whilst the other one

is used to evaluate the performance. In our project, the training set consists of 1056 out of 1525 examples in the dataset and the rest of the examples compose the test set. Two popular classifiers in the field of text analysis were implemented as the classification model. Ten fold cross-validation was included while training the model in order to ensure that the model does not overfit on test set.

6. EXPERIMENTAL RESULTS

Table I. Summary of the correlations between comment ratings and features extracted from comments

Feature	Pearson Correlation Coef.	p-Value
Number of DBpedia Annotations	0.243	6.597e-22
Number of Named Entities	0.140	4.019e-08
Number of different PoS Tags	0.176	3.823e-12
Number of Words	-0.005	0.854
Number of Replies	0.443	1.321e-56
Sentiment polarity	0.0309	0.294
Positive Prob.	0.022	0.381
Neutral Prob.	-0.016	0.525
Negative Prob.	-0.022	0.381

The threshold value α for p-test is 0.01. Prob. represent probability of corresponding sentiment where negative probability equals to 1 minus positive probability.

In order to judge the relevance of the features described in Section 5, we computed Pearson correlation coefficients with the class and implemented p-test to against Null Hypothesis as well. Class is set to be 1 when the comment has been liked and 0 when comment has not. The null hypothesis supposes that there is no correlation between the two features for the given dataset. If the p-value is less than or equal to the chosen significance level (also known as threshold α), it suggests that the observation are inconsistent with the null hypothesis which means the feature is correlated with output. The corresponding results are summarized in Table 6. As shown in Table 6, the correlation between number of replies and the class are highly confident with largest Pearson correlation coefficient and lowest p-value. Number of words, on the other hand, is completely uncorrelated with the class with p-value larger than the threshold, convincing the null hypothesis. We can easily derives that number of DBpedia annotations, number of named entities, number of different PoS tags are correlated with whether the comment has been liked or not for this dataset. In other words, we are able to predict the community feedback of the comments based on learning patterns of these features on this dataset.

However, the features extracted from sentiment analysis, on the other side, cannot escape from p-test. Both the polarity labels and probabilities present low correlation with the class and p-value far above the threshold. Even though we can see relatively different patterns in the polarity distribution for the liked comments and non-liked comments for the training set–negative sentiment occupies higher percentage for the comments have been liked compared with the comments have not been liked–as shown in Figure 2, it is still not very convincing. This failure of proving correlation can be explained by the improper sentiment classification corpus which contains movie reviews only. Movie reviews are naturally much longer than video comments so that the bag-of-features strategy cannot get obtain information for prediction. A single word in the comments can influence the sentiment of the comments significantly. Furthermore, movie reviews and video comments have different contexts and corpus. Words convey sentiment in comments may be undetectable by using classifiers trained with movie reviews corpus. Another possible reason for that might be the small scale of the dataset.

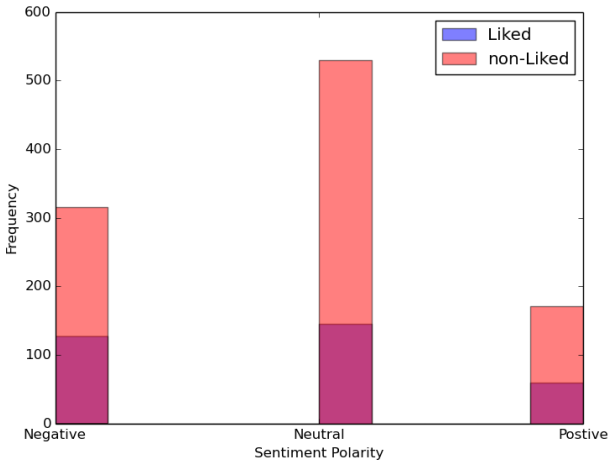

Fig. 2. The sentiment polarity distribution for comments have been liked and comments have not. The purple bars are mixed with blue and orange.

Stefan Siersdorfer et al. constructed the dataset containing millions of comments for 67000 videos in various categories whereas our dataset contains only thousands of comments for 10 videos in the two categories, talk show and animation, only. The number of categories was limited to two because we want to limit the variations for the comments so that a small training set can still show relatively good performance. The method can then be generalized to broader categories of videos. We can imagine that comments for political videos are likely to be sensitive to sentiment. The correlation between sentiment and comment ratings tends to be weak for comments under entertainment videos compared with political ones. Therefore, the results shown for our dataset is most likely to be biased.

Stefan Siersdorfer et al. has shown that sentiment in comments is related to ratings, whether positive rated or negative rated. Based on their conclusion, we suppose that the sentiment in video comments is also correlated with whether the comments are liked or not as long as the sentiment classifier is trained with YouTube comments corpus and the scale of the dataset the large enough while the video categories are diverse. Therefore, we included the sentiment polarity probabilities as the features for the classification so that, we are able to extend our model if the proper corpus are developed and the corresponding sentiment analysis result is obtained.

Table II. Summary of Classification Performance

Evaluation Measures	Artificial Neural Networks		Random Forest	
	565 examples	1056 examples	565 examples	1056 examples
CV ROC Area	0.641	0.732	0.874	0.775
Test ROC Area	0.668	0.702	0.673	0.704
Class 0 F-Measure	0.806	0.89	0.807	0.827
Class 1 F-Measure	0.417	0.405	0.47	0.459

Class 1 represents comments being liked whilst Class 0 stands for comments not being liked. ROC Area denotes the area under Receiver Operating Characteristic (ROC) curve which describes the true positive rate against false positive rate. All correct classification will have ROC area as 1 whereas random guess will have 0.5. Cross-validation (CV) evaluates the performance of the model on training set and Test evaluates the performance on test set. F-Measure is a weighted sum of precision and recall which provides relatively accurate performance measure even if the dataset is unbalanced.

The performance of the developed models with classifiers artificial neural network and random forests are summarized in Table 6. We can see that the ROC area for both classifiers on the test set reach around 0.7 given only 1056 examples. The cross-validation (CV) ROC areas are listed here to ensure that the classifiers are not overfitted. The two classifiers have relatively high F-measure, which is weighted sum of precision and recall, on class 0 and relatively low F-measure on class 1. This is because the dataset is unbalanced which means that class 0 are far more than class 1 in the dataset. Even though unbalanced problem is a challenging problem in classification, a sufficiently large dataset can significantly alleviate the influence of unbalanced distribution of two classes. The increase of the evaluation measures when the size of the training set varies from 565 to 1056 convinces that a larger dataset is able to augment the performance of the classification system.

7. CONCLUSION

We have presented how can we apply semantic enrichment methods to extract descriptive features from YouTube Comments. Three out of four features extracted based on semantic enrichment, which are number of DBpedia annotations, number of named entities, number of different PoS tags, show high relevance to whether the comments are favored or not according to the result of p-tests. A Naive Bayes Model developed by Kamal and Badruddin with movie reviews was implemented to helped mining the sentiment from the YouTube comments. Even though the sentiment polarity does not show correlation with comment ratings for our simplex collections, we still look forward to connection between sentiment and comment rating if a larger dataset containing videos from various categories can be obtained. A specifically trained sentiment classifier with YouTube Comments corpus is believed to be able to improve the relevance test for the sentiment polarity. With the relevant features from semantic enrichment and polarity probabilities from sentiment analysis, we are able to build classification models showing relatively good performance considering the size of the training set. In order to improve the performance, a sufficient large and diverse collection is required.

REFERENCES

Ammari, Ahmad, Vania Dimitrova, and Dimoklis Despotakis. "Semantically enriched machine learning approach to filter YouTube comments for socially augmented user models." UMAP (2011): 71-85.

Siersdorfer, Stefan et al. How Useful Are Your Comments?: Analyzing and Predicting Youtube Comments and Comment Ratings. Proceedings of the 19th International Conference on World Wide Web. New York, NY, USA: ACM, 2010. 891900. ACM Digital Library. Web. 14 Dec. 2015. WWW 10.

Thelwall, Mike, Pardeep Sud, and Farida Vis. Commenting on YouTube Videos: From Guatemalan Rock to El Big Bang. Journal of the American Society for Information Science and Technology 63.3 (2012): 616629. Wiley Online Library. Web.

Pang, Bo, and Lillian Lee. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005. 115124. ACM Digital Library. Web. 13 Feb. 2016. ACL 05.

Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. "Incorporating non-local information into information extraction systems by gibbs sampling." Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005.

Joachim Daiber, Max Jakob, Chris Hokamp, Pablo N. Mendes Improving Efficiency and Accuracy in Multilingual Entity Extraction. Proceedings of the 9th International Conference on Semantic Systems (I-Semantics). Graz, Austria, 46 September 2013.

Roth, Dan, and Dmitry Zelenko. "Part of speech tagging using a network of linear separators." Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2. Association for Computational Linguistics, 1998.

Lord, Charles G., Lee Ross, and Mark R. Lepper. "Biased assimilation and attitude polarization: The effects of prior theories on subsequently considered evidence." Journal of personality and social psychology 37.11 (1979): 2098.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. 7986. ACM Digital Library. Web. 13 Feb. 2016. EMNLP 02.

Pang, Bo, and Lillian Lee. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004. ACM Digital Library. Web. 13 Feb. 2016. ACL 04.

Kamal, Badruddin. Application of sentimental analysis in adaptive user interfaces. Diss. BRAC University, 2011.

30 Trendy Internet Slang Words and Acronyms You Need To Know To Fit In, http://www.makeuseof.com/tag/30-trendy-internet-acronyms-slang-need-know-fit/