YouTube Comment Classifier

mandaku2, rethina2, vraghvn2

Abstract

Youtube is the most visited and popular video sharing site, where interesting videos are uploaded by people and corporations. People also express their opinions about a video through comments using their google account. Since watching an entire video is time consuming, people decide whether to watch it or not based on the comments. For popular videos, large number of comments are present, most of them being spam and obscene comments. Youtube comment classifier aims to classify the comments into several categories like inappropriate comments, spam and others so that people can read the relevant comments only. We utilize a short text classification library called LibShortText and try different preprocessing approaches to find which methods work the best for classification.

Introduction

Youtube, the most popular video sharing site, is viewed by millions of people for daily basis. Users view the site for different purposes like entertainment, knowledge, expressing opinions etc. In a short span of time, unlikely videos get highly popular and secure a large number of hits from users of different parts of the world. The site also allows a user to comment on a particular video through his/her Google account. Many a time, the user's comments express his approval or dislike for the video uploaded. Generally, a person decides on the 'watchability' of the video by analyzing the number of likes, the number of hits and the number of positive comments obtained. Though the number of likes and the number of hits provide solid metrics on how good a video is, it is not possible to get subjective opinions on the video from those metrics. For example, a video may be bad overall, but it may be contain scenes that are excellent for a short duration in the middle. These kind of subjective positive and negative opinions can be obtained only if the user scours through the comments.

In the current system, popular videos have a plethora of comments and it is cumbersome to scour through all of them. This process if further worsened by the fact that spammers and unethical users use popular videos as a platform to further their goals of posting obscene, inappropriate, irrelevant, spam comments etc. Though options are given to mark a comment as spam and report abuse, the user is generally victimized when he does this. Also, detection of these types of comments by Youtube is a time consuming process and these comments are most prevalent in popular videos, which have the tendency to spread quickly throughout the world. Since the comments are being viewed by a global audience, comments that tend to hostile towards a particular community, race, religion, nation etc. tend to cause misconceptions and mistrust towards that group. Though Youtube takes necessary steps to curb such actions, many of such comments are missed out from blockage because of the number of likes it receives from the same person who posted it by logging into different profiles. There is a need for making the scouring of Youtube comments a more easier process. Our system, Youtube comment classifier aims to achieve this goal by categorizing comments into different categories and providing a simplistic way for the user to scour through the comments. A Google Chrome extension is developed, which when clicked, categorizes the comments for the particular url using classifiers in the backend, which are trained to find the appropriate categories.

Related Work

Many approaches in short text classification counters the problem of sparseness of short texts by enriching original short text with an additional set of auxiliary data, consisting of semantically related long texts. This is achieved either by querying search engines using the short text data or gaining additional knowledge from sources like Wikipedia or Wordnet. While these approaches produce promising results, they are inherently noisy operations and there is a good chance that the auxiliary data is semantically unrelated to the short text data. Clustering Short Texts using Wikipedia [1] discusses how to the clustering accuracy of short texts is improved by enriching the representation of short texts with additional features from Wikipedia. This paper provides an interesting solution to the information overload problem that is frequent in popular feed sources.

A General Bio-inspired Method to Improve the Short-Text Clustering Task[2] proposes methods to improve the clustering of short text documents by using a general bio-inspired method based on the AntTree approach. By taking in the results obtained from arbitrary clustering algorithms as input and refining them in various stages, interesting improvement in the result is obtained. PAntSA, which is a partial version of the hierarchical AntSA (AntTree-Silhouette-Attraction) method, is proposed as an improvement technique for short text clustering algorithms in this paper.

Short Text Classification in Twitter to Improve Information Filtering[3] offers interesting insights into how exploiting domain specific features can considerable improve the classification accuracy of short texts over the bag-of-words approach, even without meta-information like Wikipedia. It considers Twitter as the domain and aims to categorize incoming tweets into the following categories: News, Events, Opinions, Deals, and Private Messages. Authorship of a tweet is incorporated as an important feature, and this plays an important role in the classification of tweets. It is based on the assumption that users in Twitter generally adhere to a particular tweeting pattern

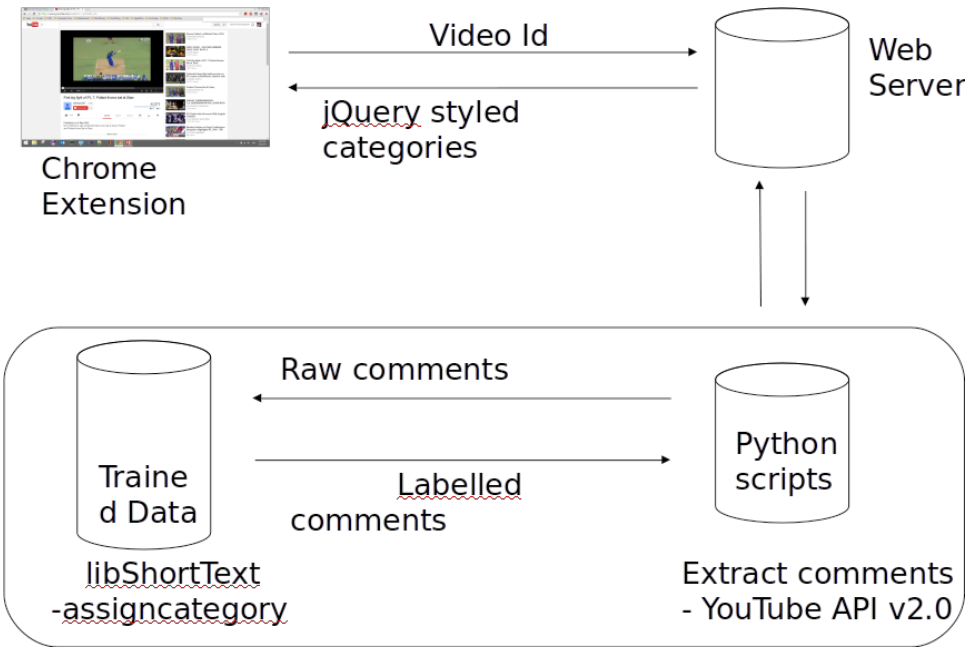
Though text enrichment of the sparse text from external sources may produce good results in other domains, we concluded it will not work for the application of detecting obscene and spam comments, owing to their fundamental difference from normal text. Also, we decided against the use of authorship as a feature in our machine learning approach as, unlike in twitter, users did not conform to a specific commenting trend in youtube. It was found that users reacted differently to different videos. A person who posted obscene comments in one video was found to be positive and useful comments about another video. LibShortText, a short text classification library, which has been developed keeping in mind the unique properties of short text, has been employed to achieve the goal of classification. Also, various text preprocessing approaches were tried out to determine which worked best for our application.

We developed a Google Chrome extension that assists user in scouring through comments and avoiding irrelevant Youtube comments by dividing the large corpus of comments for a video into several different categories.

Problem Definition

The overall objective of the project is to efficiently classify the short comments present in a video into one of the following categories:Obscene comments, irrelevant or spam comments and others. Sub problems include collecting huge training data for training models that efficiently categorize comments into the above mentioned categories, development of Google chrome extension to provide control of the classification of comments to the user and testing the efficiency of the classifiers. The training of classifiers is done using LibShortText, the short text classification library.

Product Architecture



As diagrammatically explained, the following actions take place once the extension button is clicked:

* Sending of video ID corresponding to the Youtube page viewed to the web server hosted
* The server, with the help of python scripts, extracts the Youtube comments for the corresponding video ID
* The raw comments are assigned dummy labels and are given as test data to the model files generated after training
* Once the classification is done, the comments with assigned labels under different categories are sent back to the server, which sends to the browser as jQuery styled comments, which are then displayed in a neat interface.

Data Collection

A total of 47,432 comments were collected from video using python scripts. Care was taken that they belong to videos of different genres and sufficient comments belonging to all the above mentioned categories were extracted. It was made sure that the data set collected was representative so that classifiers performs well with unseen test data.

Short text preprocessing

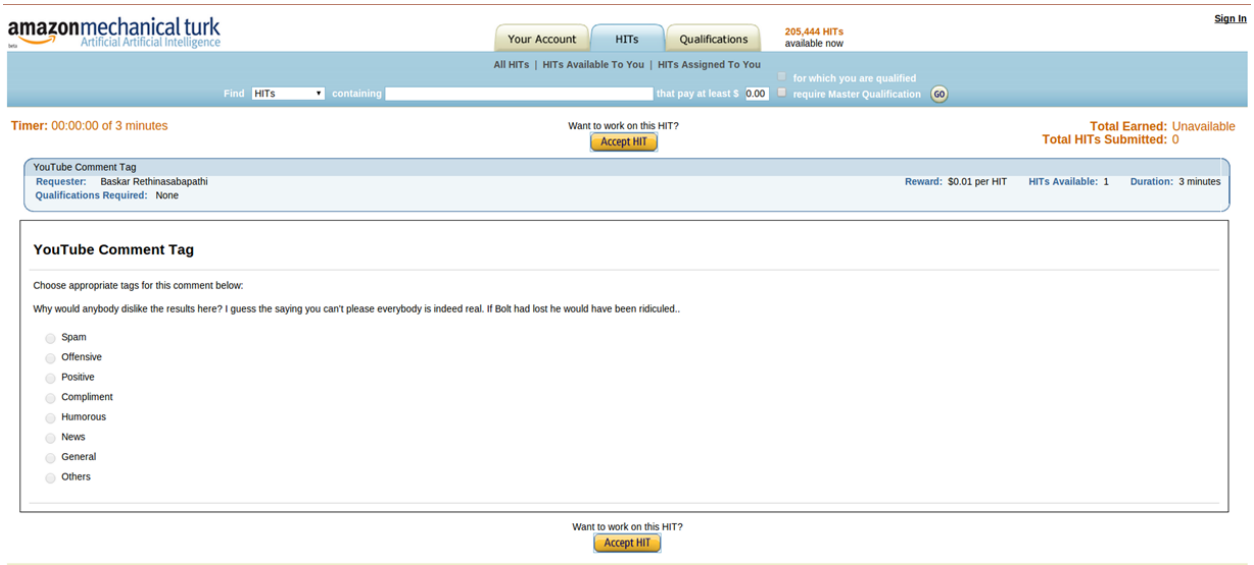
It was questionable whether normal text preprocessing techniques like stemming, stop word removal and formation of unigram/bigram word features would be applicable to short text classification.

Stemming is the process of reducing the inflected words to their stem. Stop word removal is the process of removing the commonly occurring words like is, the etc. Arunachalam *et al.* argue that efficient short text classifiers are obtained by not employing stemming and stopword removal and by using bigram word features in their paper Product Title Classification versus Text Classification[4]. We decided to to try different combinations of the preprocessing techniques and find out which one works best. The results of the various approaches have been discussed in the results section of this report.

Training of classifiers

Labelling using Mechanical Turk

The comments crawled across different domains like Politics,Sports,Music,Animals,Movies etc were cleaned for machine detectable spam and then a random batch of 10000 comments were selected to be annotated by online workforce. We chose Amazon Mechanical Turk as the workforce platform since it had the largest number of workers who would quickly finish the task. We divided the comments into ten batches of 1000 comments each and each batch was given to 100 workers to work on, putting a cap of maximum of 10 comments by each worker. The random comment is posted along with a range of options to be labelled as radio buttons. This forced the worker to select only the best fit option and hence helped in improving the accuracy of the classifier later. This is the screenshot of a typical HIT as workers would see in Mechanical Turk.



We posted half of the work in the sandbox so that we need not pay for them, but the batch progressed slowly compared to the actual platform where we paid 1 cent per HIT. At the end of the cycle we made use of mturk python API to programmatically download each individual user's answers and annotated using an online annotation toolkit. This was then used as the primary training data for the LibShortText classifier.

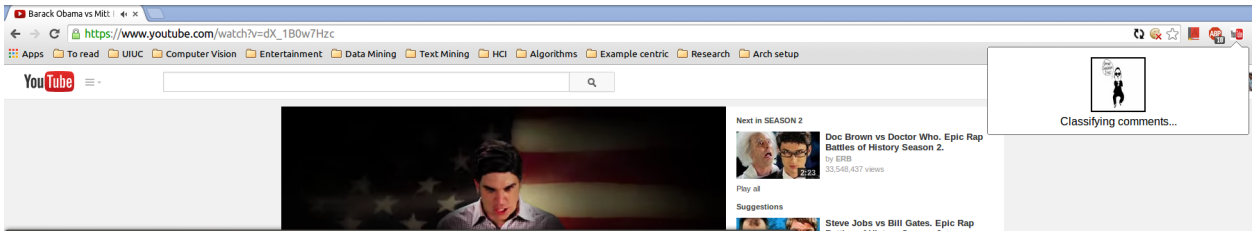
LibShortText

LibShortText is an open source library for short text classification and analysis. The fast training and testing is built on top of the linear classifier, LIBLINEAR. The preprocessed data is converted into LIBSVM format and then the training is carried out. Training and testing on the pre processed and feature generated comments using *text-train.py* and *text-predict.py*. Training was performed using the labelled 47000 comments for all the three categories on computer with the following configuration: 1.60GHz, 4GB RAM. Three model files for the three categories are generated, which can be used for classification.

Evaluations and Sample results

Extension Output Screenshots

1. When watching a YouTube video, click on our extension icon. It will crawl all the comments in the video, classify them on server side. Depending on the number of comments in the video, the classification result will be delayed.



2. View the comments classified to different categories - spam,offensive and others

Results by following different combinations of preprocessing techniques

Accuracy values for various combinations for obscene comments classifier.

Stop word removal	Stemming	Unigram/ Bigram features	Accuracy
No	No	U	91.1535
No	No	B	91.2664
No	Yes	U	90.5121
No	Yes	B	91.0663
Yes	No	U	89.296
Yes	No	B	89.5269
Yes	Yes	U	88.8598
Yes	Yes	B	88.9009

As discussed, the highest value has been obtained by not performing stemming, stop word removal and the use of bigram features and were in accordance to what the paper argued.

Five-fold Cross Validation

Five fold cross validation is the method by which the entire data set is divided into five equal parts. Each time, four sets get to be the training data and the resultant classifier is tested with the fifth data set. This process is repeated so that each part gets to be the test data once. This provides an overall sense of the accuracy of the classifier generated. The average accuracy obtained for the obscene comments classifier was 90.29%.

Test Set	1	2	3	4	5
Accuracy	90.2545	89.2128	89.8029	90.8965	91.2664

Similarly, the accuracy for the spam category was 81.43% and the accuracy of useful comments category was 91.2%.

Conclusion and Future work

Minimum deliverable goals are successfully achieved as promised. We could deliver a valid Chrome extension which does the following

- categorize comments which are less relevant
- highlight comments which are more relevant
- filter out spams, advertisements, malicious links, obscene comments, copyright violations
- prevent scrolling through numerous useless comments by clustering similar sounding comments under separate sections
- Mine comments for hidden gems like - time pointers inside videos, unknown facts and trivia mentioned

Since this extension has a severe server overload, uploading it in the Chrome Web store would require scaling up our server capacity. We plan to do this with a cloud infrastructure. If that goes well, our future work will be focussing on the following things

- An exclusive search engine for YouTube videos making use of comments
- A mobile app for searching YouTube videos based on comments
- pre-cache video links in the comments and prepare a better “Watch Next” list
- Overlay comments made on particular parts of the video (like Soundcloud) for better visibility of every comment (with a toggle)

Individual Contributions

mandaku2 - Server creation and scripting, Python Scripts for Integration with LibShortText
rethina2 - Python scripts for Mechanical Turk HIT validation, YouTube Comments mining, Chrome extension development
vraghvn2 - Text Preprocessing, Training Classifier

References:

[1]Banerjee, Somnath, Krishnan Ramanathan, and Ajay Gupta. "Clustering short texts using wikipedia." *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007.

[2]Ingaramo, Diego, Marcelo Errecalde, and Paolo Rosso. "A general bio-inspired method to improve the short-text clustering task." *Computational Linguistics and Intelligent Text Processing*. Springer Berlin Heidelberg, 2010. 661-672.

[3]Sriram, Bharath, et al. "Short text classification in twitter to improve information filtering." *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010.

[4]Yu, Hsiang-Fu, et al. *Product title classification versus text classification*. Technical report, 2012. URL http://www. csie. ntu. edu. tw/~ cjlin/papers/title. pdf, 2012.

[5] Serbanoiu, Andrei, and Traian Rebedea. "Relevance-Based Ranking of Video Comments on YouTube." *Control Systems and Computer Science (CSCS), 2013 19th International Conference on*. IEEE, 2013.

[6] Lee, Kathy, et al. "Twitter trending topic classification." *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. IEEE, 2011.

[7] Magazine, D-Lib. "Academic Libraries on Facebook: An Analysis of Users' Comments." *D-Lib Magazine* 17.11/12 (2011).

[8] Liu, Dong, et al. "Boost search relevance for tag-based social image retrieval."*Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*. IEEE, 2009.

[9] Steiner, Thomas. "A meteoroid on steroids: ranking media items stemming from multiple social networks." *Proceedings of the 22nd international conference on World Wide Web companion*. International World Wide Web Conferences Steering Committee, 2013.

[10] Marcus, Adam, et al. "Twitinfo: aggregating and visualizing microblogs for event exploration." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2011.

[11] Inches, Giacomo, and Fabio Crestani. "Overview of the International Sexual Predator Identification Competition at PAN-2012." *CLEF (Online Working Notes/Labs/Workshop)*. 2012.