Naïve Multi-label classification of YouTube comments using comparative opinion mining

Asad Ullah Rafiq Khan*, Madiha Khan, Mohammad Badruddin Khan

Mohammad Ali Jinnah University, Karachi Pakistan
Jinnah University for Women, Karachi Pakistan
Al-Imam Muhammad ibn Saud Islamic University, Riyadh Saudi Arabia

Abstract

With the turning wheel of time, the influence of the social networking websites on the people has significantly increased. People are now connecting with each other in cyber space and show their sentiments in the form of comments in different social networking websites such as Twitter, Facebook and Google Plus. YouTube is considered as a king in the field of video sharing. It is a largest video sharing repository, where people come and share their thoughts regarding video in the form of comments. If we are able to find useful information through comment, then these unstructured comments can be useful for different purposes. Sentiment analysis is the one way to find out the feeling of people and in the case of YouTube, we can understand the behaviour and response of individuals after seeing particular video. There are situations in which opinion shared by user has comparative content. The user sees the video of comparison of two options or products and shares his/her preference based on some reasoning. Comparative opinion mining leads to situation where number of options defines the number of labels. In this paper, we have used Naïve Bayes machine learning algorithm to perform multilabel classification to find out the sentiments of the commenters for different options. In order to reduce the computational requirements, we adopted a naïve assumption that words around keywords related to particular option are enough to understand the sentiments of user. The developed classifier based on naïve assumption demonstrated slightly lower performance with the benefit of requirement of less computational power.

* Corresponding author. Tel.: +92 332 3540172
E-mail address: asad.uk2009@gmail.com
1. Introduction

There are numerous videos sharing websites such as, Vimeo, YouTube, Dailymotion and so on, but YouTube is the website where largest number of videos are shared among the billions of users across the whole world. People can share their thoughts, ideas and their mood by sharing videos, responding through posting comments. In 2013, YouTube was garnering nearly 20 percent of all global traffic on the Internet. Its popularity is just one clue that YouTube is the best video sharing site out there.

There are lot of knowledge shared among people by video sharing and people also shared their thoughts and feelings about videos in the form of comments. These comments can be relevant and irrelevant. Relevant comments contain opinion about the video of the commentator, which is useful in the analysis of sentiments. Basically, opinion mining is concerned about “How people think about particular thing, person or idea” that’s why it is usually believed that these comments can provide useful information in various businesses to take decision according to individuals’ sentiments.

There are mainly two types of comments, which are relevant and irrelevant comments. Relevant comments are of various types. These are some types of relevant comments:
1. Declarative comment
2. Comparative comment
3. Direct opinion comment
4. Comments contain more information

We focused on the videos, in which all of these comments are available, but we choose to work on comparative comments. Comparative sentiment analysis is one of the challenging works, which is done by number of researchers. Basically in the comparative sentiment analysis we have to deal with the multi-aspect comments. Commentator usually compares more than one things, people or idea on the basis of some aspect. We did not try to find aspects on the basis of which the user shares its positive, negative or neutral comment for particular option. In this sense, the paper describes aspect-less comparative opinion mining. Even with this simplification, comparative opinion mining is much more complex than working on sentiment analysis of declarative comments.

In this research paper, we have worked on Iphone vs Android video, which consisted of over 8000 comments. The comments were of different types. We manually filtered comments and only used comparative comment in our research. Our dataset in this research is about 400 comments which are almost 5% of the original dataset. Due to dataset comprising of comparative comments between two products, we have 6 classes of two labels instead of 3 classes because three classes are related to android and three are for iOS, that’s why there are various challenges in handling comments and analysis sentiments of the users.

We have used machine learning technique for sentiment analysis approach for the same task, so that we can understand the opinion of the community. we performed following tasks:
- Gathering of data (gathering comments)
- Removal of noisy and irrelevant data.
- Manual assignment of sentiments to the comments in order to make training corpus.
- Development and evaluation of classification model

In the upcoming sections, we will discuss related research, followed by comments gathering process and then we describe process of removing inconsistencies from comments, and after that problem with annotations of comments will be discussed in detail. Experiments, discussions, conclusion and future research will be discussed afterwards.

2. Literature review

Many researches have been conducted in the field of sentiment analysis, as it has been of immense research interest in recent times due to the powerful prediction trends it can provide based on social opinion and concerns. The influence of the sentiment analysis is in the various disciplines such as stock prediction, news association, product analysis, social issues, and social communication trends.

[1] studied sentiment analysis in the field of products and movies reviews. The advantage is that they already have a clearly specified topic, and it is often (reasonably) assumed that the sentiments expressed in the reviews have to do with the topic. [2] proposed the Stock Sonar model and they used it to predict stock trends on the basis of a
hybrid approach that integrated semantic events obtained from several stock related RSS feeds, sentiment dictionaries and applying comparative sentiment patterns to associate sentiment scores. [3] focused on sentiment analysis on social issues. They conducted a statistical study on the variances between sentiment analysis of products and social issues and then they tried to prove and showed the importance of verb in expressing opinions regarding the social issues. [4] proposed learning sentiment-specific word embedding dubbed sentiment. They addressed this issue by encoding sentiment information of texts (e.g., sentences and words) together with contexts of words in sentiment embedding. [5] [6] focus in the field of transportation and [5] proposed the traffic sentiment analysis (TSA) as a new tool to tackle this problem, which provides a new perspective for modern intelligent transportation system.

[7] has done a great work on customer reviews of products. Their work was based on language pattern mining and proposed technique to extract product features from Pros and Cons in a particular type of reviews. [8] worked on comparative analysis of several research methodology related to sentiment analysis and presented vast number of references related to comparative sentiment analysis. [9] worked on comparative dataset and they extracted the adjectives from the dataset and filter on the basis of positive and negative comments and remaining comments are dropped by author and based on their methodology they got about 90% result. [10] used 14 different comparative articles from different sources as a dataset and they tag sentences as a subjective or objective and they analysed resultant confusion matrix and they also tagged compound sentence in this research.

3. Methodology

In this section we will discuss our methodology step by step.

A. DATA COLLECTION

YouTube provide application program interface (API) to fetch data related to videos, and users, such as, user profile, video, comments thread and so on. We wrote java program to fetch all the comments of “IPHONE VS ANDROID” video and used YouTube API and GSON library to parse JSON. We gathered nearly 8000 comments then we filtered all the comments and stored only comparative comments in our MySQL database. To do this, we read manually 500 comments and find keywords related to android and iOS (List of keywords is presented in Table 1) and then we wrote JAVA program that fetch only those comments in which keywords related to android and iOS were present at same time and we got 528 comments out of 8000 comments. After reading them manually we found that around 390 comments were relevant for our research which is about 5% of total comments of this video.

B. CLASS ASSIGNMENT

Assigning class manually was one of the time-consuming and tricky tasks. For each comment we assigned classes for two labels, one label was related to sentiments for android and other was related to sentiments for iOS. For each label, assignment of any of three classes for single comment lead to 6 total classes.

In another experiment we concatenate class related to android with class related to iOS and we got eight classes. Ideally we should have nine classes but we did not have data of neu neg (neutral of android joined with negative of iOS) class. In the joined classes, first class was related to android and other one was for iOS.

In the following lines, we are giving example of one comment for each class combination to show the complexity of problem.

**POS POS class:** I love my Android devices and I love my iOS devices. I love my MacBook Pro with OSX Mountain Lion and I love doing my 3D modeling on it after rebooting into Windows. OSX almost never fails and my rendering software runs flawlessly with my 3D mouse in Windows 7.

**POS NEG class:** I think android is better because you can update more easily than ios 6. Clearly there are some phones that are not updated good video man.

**POS NEU class:** I hate when Apple users complain about the screen size of Android phones because these are the same people who use their iPad to do things that their iPhone does better. Like playing card games music and taking pictures. U Apple faithful use 10 screen because ur 3.5 now 4 is to small and my 4.8 is too big Smh. Silly sheeple.

**NEG POS class:** iPhone 5 : Limited Jailbroken iPhone 5 : Fully customizable better than any android phones.

**NEG NEG class:** You can see how fucking bad the Nexus and iPhone < 3 are.
NEG NEU class: Honestly this guy is biased. He drops the notification center for Android and it shows all his isht in one bulk too stating he has 30 messages. Its still either all or nothing as he said to get rid of those messages. What a loser. Is there ever going to be a non-biased video out there for both Apple and Google!

NEU POS class: I think that you are a android lover because iOS is defenatly much better in so many ways.

NEU NEU class: there is no winner someone like ios someone android it is up to customer.

C. DIFFICULTIES WITH ASSIGNING ANNOTATIONS

In the step of preparing data, we faced numerous challenges in annotating comments as it was difficult to understand the polarity of some comments due to ambiguity, spelling errors, missing punctuation in the sentences and grammatical errors. We gave two labels for each comment. One with respect to android and other one with respect to iOS.

In the following lines, we will describe each problem one by one, which we faced in annotating the comments.

Handling problems with symbols and short forms

Given below are the two comments:

1. ANDROID gt ios and in some aspects os gt Android
2. Hardware: iPhone 5 > Galaxy Nexus OS: iPhone 5 < Galaxy Nexus

Both of the above comments contain either mathematical symbol(<, >) or short forms(gt for greater). In order to annotate these comments, we first gave consistent forms to ensure uniformity.

Hence above comments were changed to following form.

1. Android is greater than ios and in some aspects os is greater than Android.
2. Hardware: iPhone 5 is greater than Galaxy Nexus OS: iPhone 5 is lesser than Galaxy Nexus

The uniformity is necessary because of following reasons:

1. We cannot filter part of speech of mathematical symbols and informal short forms like gt and ls and determining correct part of speech is vital for segment of our experiments.
2. WEKA and MEKA tools limitations for some symbols

There are two types of ambiguities in the comments which are described in following lines.

Ambiguity in comments – Type I:

Some comments are ambiguous in the sense that it is difficult to decide whether comments are positive or not and therefore it is difficult to annotate such type of comments.

1. galaxy nexus + iphone 5 = nexus 5
2. I love iOS and hate android but my fiancé loves android and hated iOS but is starting to come around

In the first comment it is difficult to understand that whether commentator is directed towards android or ios. Different interpretations of comment are possible. It can mean that nexus 5 is the copy of iphone 5 or another interpretation can be that nexus 5 has more feature than iphone 5 as it contains all the features of iphone 5 and galaxy nexus.

Second comment is a combination of two declarative sentences which are combined together with the help of conjunction “but”. In the first part of comment, the polarity of the comment is positive towards iOS and negative towards android and in the second part, it seems that comment is positive towards android and negative towards iOS. It was difficult to understand the polarity of first comment, therefore we have excluded it from our dataset, whereas for the second comment we assign positive class for both labels describing sentiment for android and iOS.

Ambiguity in comments – Type II

There are some comments, which are written in the way such that they are apparently ambiguous because of punctuation, grammatical and typo mistakes.

1. Hardware: iPhone 5 is greater than Galaxy Nexus OS: iPhone 5 is less than Galaxy Nexus.
2. Android is greater than ios and in some aspects os is greater than Android.

First comment is ambiguous because there is missing punctuation. If reader reads the comment in first instance, it is difficult to interpret the polarity. The comment is composed of two predicates which are:

1. Hardware: iPhone 5 is greater than Galaxy Nexus
2. OS: iPhone 5 is less than Galaxy Nexus

Basically due to poor usage of English language, it is difficult to understand what the intention of the commentator in the comment is.
Second comment should be
“Android is greater than ios and in some aspects OS is greater than Android.”
In the second comment, commentator forgot to write letter “I” prior to “os”, that’s why it is difficult to understand what user tried to say in the comments.

D. FINDING PART OF SPEECH AND NEIGHBOUR WORDS OF KEYWORDS FROM COMMENTS
We have used GATE plugin and its library, wordnet and Stanford libraries in order to find the part of speech from the comments. We have used these libraries and wrote java program for retrieving part of speech.

```
1: Procedure getPOS(comment : String)
2: Load GATE plug-in
3: tokenizer = instantiate Gate Tokenizer
4: regexStn = instantiate Gate creole regex sentence splitter
5: posTagging = instantiate GATE creole pos tagger
6: localDoc = instantiate Gate Factory(comment)
7: tokenizer.setDocument(localDoc);
8: regexStn.setDocument(localDoc);
9: posTagging.setDocument(localDoc);
10: ArrayList<String[]> posWithToken;
11: annotationSet = get annotation from localDoc
12: posWithToken = annotationSet.getPosWithToken()
13: return posWithToken
14: end procedure
```

```
1: procedure
getNeighboursWordsOfKeywords(comment : String)
2: iOSKeywordList ← getList(“iphone”)
3: androidKeywordList ← getList(“android”)
4: String androidKeyword=
getKeywordFromComment(comment,3, androidKeywordList)
5: String iOSKeyword=
getKeywordFromComment(comment,3, iOSKeywordList)
6: String[] keywords = {androidKeyword, iOSKeyword};
7: return keywords
8: end procedure
```

We have written java routine for getting neighbours words of the keywords that are related to the iOS and android. List of the keywords are available in table 1.

<table>
<thead>
<tr>
<th>iOS Keywords</th>
<th>Android Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>Android</td>
</tr>
<tr>
<td>Apple</td>
<td>Samsung</td>
</tr>
<tr>
<td>Siri</td>
<td>Nexus</td>
</tr>
<tr>
<td>Iphone</td>
<td>HTC</td>
</tr>
<tr>
<td></td>
<td>Froyo</td>
</tr>
<tr>
<td></td>
<td>Cupcake</td>
</tr>
<tr>
<td></td>
<td>Donut</td>
</tr>
<tr>
<td></td>
<td>Eclair</td>
</tr>
<tr>
<td></td>
<td>GingerBread</td>
</tr>
<tr>
<td></td>
<td>HoneyComb</td>
</tr>
<tr>
<td></td>
<td>Ice Cream Sandwich</td>
</tr>
<tr>
<td></td>
<td>Jelly Bean</td>
</tr>
</tbody>
</table>

Table 1: List of keywords

We used noun, adjective and verbs in our experiments and as well as words nearest to keywords of android and iOS, that’s why we retrieve it and placed them in a separate column. Hence in this stage, we prepared our dataset to be used for experiments. Our dataset consisted of following columns.
1. Comments
2. Comments filtered based on Part of Speech
3. Neighbour words around keywords
4. Label iOS (three possible value: Pos, Neg, Neu)
5. Label android(three possible value: Pos, Neg, Neu)
6. Joint label (concatenation of label android and iOS)

E. TOOLS AND STEPS USED FOR CLASSIFICATION

We used WEKA and MEKA (extension of WEKA for multi label classification), specialised softwares, to perform machine learning tasks. Following are the steps taken to develop classification model.

1. Data Processing and Class balancing

Word vector is basically creating bag of word by splitting sentence into the words. WEKA and MEKA give several options of creating tokens. We used 1-gram and 2-gram tokens for our classification process.

2. Classification

We used naïve base classifier in our experiments, which is basically a probabilistic classifier. Experiments were performed with following different settings in MEKA:

1) Input: Full YouTube comments
2) Input: Comments with Part of speech filtering
3) Input: Words in neighbourhood of Android and iOS keywords

In all settings, number of labels were 2 with 3 classes each.

The target of experiments was to check the extent of performance deterioration for naïve assumption of usage of keywords. Due to MEKA limitations, we used WEKA to gain deeper insights of model. Experiments were performed with different settings in WEKA. The different settings can be seen in Table 3. It should be noticed that WEKA does not support multi-label classification.

3. Naïve Bayes Probabilistic classifier

It is not possible to give full probability table that resulted after application of Naïve Bayes algorithm on the word vector. However, we will give probability values of a word android in order to illustrate the nature of discoveries by statistical machine learning algorithm.

ANDROID: 7.23E-04(pos) ; 2.87E-04(neg) 3.46E-04(neu)

One can clearly see that Android when written in full capital form indicates positive sentiment for label Android.

4. Result and Discussion

Table 2 and Table 3 show the results of different experiments. With three classes, the performance of a classifier doing random assignment is theoretically 33.33%. With sophisticated comments that are difficult to be understood by human annotators, it is quite understandable that we expected our classifier performance to be low. The main thing to be noticed for MEKA experiment is that naïve assumption to take neighbourhood words of keywords (described in table1) worked quite well in comparison to situation where full comments was either utilized or Part of speech filtering was applied with retention of nouns, adjectives and verbs.

<table>
<thead>
<tr>
<th>Multi label Classification</th>
<th>Android Accuracy</th>
<th>iOS Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment</td>
<td>57%</td>
<td>52.50%</td>
</tr>
<tr>
<td>Part of Speech</td>
<td>56.40%</td>
<td>53.80%</td>
</tr>
<tr>
<td>neighbourh ood words of keywords</td>
<td>55.90%</td>
<td>52.94%</td>
</tr>
</tbody>
</table>

Table 2: MEKA - Multi-label classification result
Table 3: WEKA - Single Label and Joint Label classification result

Table 3 gives summarized results for all experimental settings that were used in WEKA environment. The reason to use WEKA is to gain deep insight of the classifier nature with single label classification. It is interesting to note that how multi-label classification has balanced [48.42%  61.18%] (accuracy in WEKA with setting of single label,
neighborhood words of keywords to [55.90% 52.94%] from MEKA. The input based on naïve assumption again was found to have comparable performance with other input schemes. Since we cannot perform multi-label classification in WEKA, we used work-around by joining the two labels. That means that we performed single-label classification with 9 classes. However, in our dataset, we had no data for one combination resulting in 8 classes. The random classifier for 8 classes is expected to give accuracy of 12.5%. However, the results were around 45%. It should be noted that classifier was unable to classify for few combinations and thus recall and precision for such combinations were zero. This is due to unbalanced nature of dataset resulting after combination of classes. The naïve assumption again gave comparable performance with respect to two other input schemes.

II. CONCLUSION

In this research work, comparative opinion mining of Youtube comments was performed on comments with contents containing comparison of Android and iOS. We described how the dataset was formulated and what the difficulties that were faced during annotation process were. Two tools were used for experiments. MEKA was used for multi-label classification. In order to gain further insights, experiments were performed in WEKA with different settings. In first setting, full comments were used for classification. In order to reduce computation, we filtered comments based on semantic resources and retained only nouns, adjectives and verbs. In another experiment setting, we used our naïve assumption that the neighborhoods of keywords are enough to explain the class of comments. In all settings, naïve bayes algorithm was used. The results in terms of different performance measures are not satisfactory but the naïve assumption regarding neighborhood words of keywords performed well as compare to other experimental setting. The finding is significant as with little comprise, the computational requirements can be drastically reduced. In order to get better results, we plan to work further on data processing side and use lexicon to make the comments more comprehensible for machine. We also plan to use other machine learning algorithms to check their performance.

REFERENCES