Public opinion analysis of player comments for the PV of Genshin Impact

Simei Li
Vrije Universiteit Amsterdam
s8.li@student.vu.nl

Yiran Li
Vrije Universiteit Amsterdam
y46.li@student.vu.nl

Kairui Wang
Vrije Universiteit Amsterdam
k4.wang@student.vu.nl

Tian Xia
Vrije Universiteit Amsterdam
t.xia@student.vu.nl

1 INTRODUCTION

Our project is aimed at Genshin Impact, an open-world, action RPG, analyzing the public opinion of the comments under each trailer video published on the video platform (Youtube, bilibili..) by its official account.

Through our processing process, we will output the most interested concerns of audience for each video, the most discussed topics, and the emotional tendency of the comments. This can be a general public opinion analysis method. Even if the user is not familiar with the game, he can also quickly learn the game terminology, player's concerns and emotional feedback from the results. This kind of analysis may be useful in game business analysis job, game operation job, etc.

Moreover, we also applied the same method to the Chinese video platform to compare whether players in different language communities have similar concerns and emotional feedback.

2 RUN CODE INSTRUCTIONS

Git: https://github.com/97Simei/wdps27_final.git
See README.md

3 GOAL

3.1 Text Mining

In text mining part, we do Named Entity Recognization (NER) for comments and count the number of occurrences of each entity. We select the top 20 entity words as the manifestation of players' interests. For each entity, we do entity linking to a wiki page so that anyone can know what the entity is. In addition, we try to find the top described words to each entity.

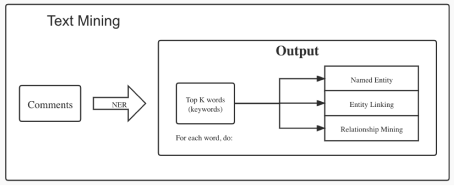

Figure 1: Text Mining

3.2 Sentiment Analysis

In sentiment analysis part, we score each comment sentimentality and label them into 5 categories based on the score: <Positive, Semi-Positive, Normal, Semi-Negative, Negative> Results are showed in pie chart. We also compare the results of several videos horizontally (emotional trend) to reflect the changes in players' emotion over time.

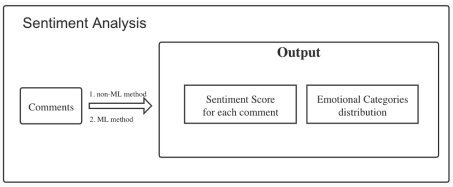

Figure 2: Sentiment Analysis

3.3 Text Mining + Sentiment Analysis

Filtering the comments by each entity word and performing sentiment analysis, the result can be regarded as the player's feedback on a certain character/activity/storyline.

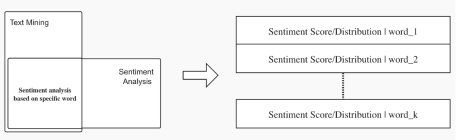

Figure 3: Text Mining + Sentiment Analysis

4 PIPELINE

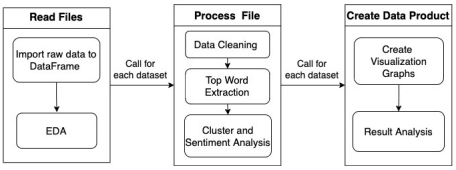

Figure 4: Pipeline

4.1 Data Extraction

4.1.1 YouTube Dataset. To get the YouTube Video's comment data, we reproduced the script from the Ahmed Shahriar Sakib' github. The script requires the *pandas* and *request* modules to support the data extraction. The script will finally dump the YouTube video comments to a CSV form. The user can also easily be sorted by popularity or timestamp when downloading the data. The data set contains 8 columns. For videos, we collect all the comments of the latest 6 trailer videos(2.3, 2.2, 2.1, 2.0, 1.6, 1.5), representing 6 different game versions in 270 days.

4.1.2 Train Dataset. A handmade crawler to bilibili and Google-Play Store through given website api.

4.2 Data Cleaning

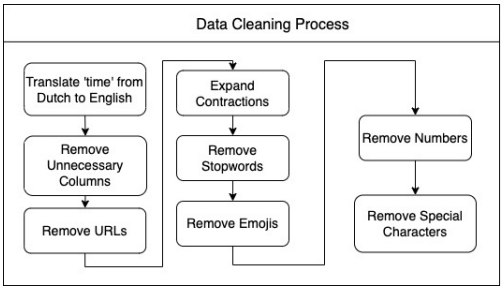

Figure 5: Data Cleaning Process

Data Cleaning is an essential step which could provide cleaned and meaningful data sources to the top word extraction pipeline.The main steps of the data cleaning process is shown in figure2. For the YouTube data set, every 'time' value should be translated from Dutch to English. Next, unnecessary columns "cid",'photo' and 'channel' will be dropped from our data frame. Some comments may include URLs and special characters such as "@user" will also be dropped. After expending the contractions, stop words like "the" will be dropped. Finally, we also dropped the emojis and numbers.

4.3 Named Entity Recognization

The project used the spacy and sentence-transformer packages to do the NER process. During the experiment, the former package has a better recognization result on the PERSON,ORGANIZATION and LOCATION result. But the former process has a longer processing time. We finally used the spacy to done with our NER, because the "popular" entity will always ranked high with its high frequency

4.4 Entity Linking

We link the Genshin Impact Fandom Wiki to the named entities we've selected. Fandom is a repository of entertainment culture. By examining if the web page is incorrect, we may determine whether this entity is affiliated to Genshin Impact. Because the usage of the same name during game development will be avoided, the word disambiguation will not be involved.

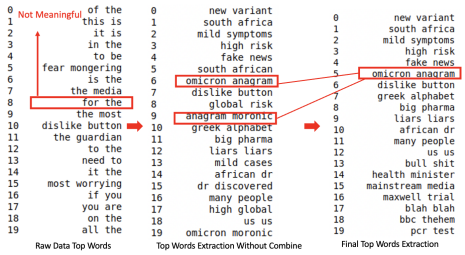

Figure 6: Result Compare

4.5 Top words Extraction

In this project, we have adopted two methods to extract top words from the document. First method calculated the frequency of the words and phrases,and rank the word or phrases based on the frequency. Secondly, we selected the top words based on the similarity and the word frequency.

4.5.1 Extract Top Words Based on Word Frequency. After the process 4.2, though data is already cleaner than the raw data, there still remains the pronoun and subject which is not very meaningful to represent the comments' main idea. As a result, we import the spacy package and use the inside pipeline "en_core_web_sm". After the Part of Speach (POS) process, we can easily extract the noun from each comment's string. The new POS extract result will become the corpus input of the "getTopMentionByFrequency" function. At first we use the "CountVectorizer" to create the bag of words data structure, next based on the vector array, we could use "sum" function to calculate the frequency of the words/triples. We could decide the size of phrase manually, in this paper we set the size as 2. After doing several experiments, it is a big problem that there are always similar bigrams in the result. For example "A,B" and "B,A" actually should be the same bigrams, they only have the different order of the word, the symantic meaning between the bigrams are same. To solve the problem, we compared similarities between bigrams, and combined the bigrams,which means that we added the frequency of the similar bigrams. Fig5 shows the difference before combining the bigrams and after combining the bigrams. As we can see in the Fig5, most of the duplicate results are eliminated. For example, "omicron anagram" and "anagram moronic" is combined into "omicron anagram".

4.5.2 Extract Top Words Based on Similarity. In the second method to extract top words, we used the algorithms called Maximal Sum Similarity. The distance between pairs of data is defined as the pairs of data for which the distance between them is maximized. In the paper, what we intend to do is to maximize the candidate top words' similarity to the document, in the meanwhile, minimizing the similarity between the candidate top words. In our project, we calculated the distance using *cosine_similarity()* function. So the first step should be rank the top words candidate based on the cosine similarity, and the next step will be calculate the combination of words that are the least similar to each other.

4.6 Sentiment Analysis

4.6.1 Data cleaning and reorganization. Remove special punctuation (e.g. emoji) and non-English symbol. According to whether quoting the original lines in the PV video, the comments are separated, and only the player's own text is selected. Remove the comments that are too long or too short. Remove duplicate comments.

4.6.2 Score and Labeling. Use TextBlob to calculate the sentiment polarity(score) and subjectivity. Remove the comments with no subjectivity. Label the comments into 5 categories, <Postive, Semi-Positive, Normal, Semi-Negative, Negative>, according to its score.

4.6.3 Machine Learning method to scoring and labeling. Motivation: Textblob is only focusing on the meaning of a single word, rather than "understand" the meaning of a sentence. Sometimes a word can significantly influence the result of a sentence. Therefore, we may have better classification accuracy of emotions by training a sentiment classification model using labeled comments in the same game.

Solution: We use crawler to obtain some comments in the google play store about this game, and each comment has a labeled rating. We get about 2000 comments and break them into training and testing data. We have tried Logistic Regression, SVM and Naive Bayes model. The best accuracy of model is about 75% under the Naive Bayes classifier, which considers the connection of words in the sentence.

5 IMPROVEMENTS

In our pipeline, we also tried to cluster the top words based on the Machine Learning methods. The main idea of the clustering is to first create a TFIDF vector, using "TFIDFVectorizer" from *sklearn* packages of *Python*. After normalizing and deleting all null values of the array, we got the reduced vector array based on PCA estimation. To observe the best cluster result, we used the Elbow Method , where can we observe the cluster score of each round in Fig7. And Fig8 is the result of the clustering, Fig9 also shows the top words in each cluster.

6 RESULT

6.1 Entity and linking result

	LINK	ENTITY	FR_TOP	SM_TOP
0	https://genshin-impact.fandom.com/wiki/Albedo	albedo	['albedo albedo', 'cryo delusion', 'eula devil', 'dragonspine event', 'story quest']	['1154 albedo devil', '11737 albedo secretly
1	https://genshin-impact.fandom.com/wiki/Itto	itto	['itto itto', 'albedo eula', 'voice actor', 'playable character', 'new boss']	['hot bully 11144', 'best husbando 11678', '
2	https://genshin-impact.fandom.com/wiki/Mihoyo	mihoyo	['mihoyo albedo', 'game coincidence', 'people china', 'death flags', 'story anniversary']	['story amazing battle', '10784 just dragon
3	https://genshin-impact.fandom.com/wiki/Bennett	bennett	['bennett bennett', 'team albedo', 'amber eula', 'screen time', 'pyro archon']	['4575 dear diary', '1785 bennett dragon', '
4	https://genshin-impact.fandom.com/wiki/Dragonspine	dragonspine	['dragonspine dragonspine', 'albedo cyro', 'new event', 'itto part', 'dragon durin']	['getting flashbacks dragonspine', '2975 th
5	https://genshin-impact.fandom.com/wiki/Genshin	genshin	['genshin genshin', 'game mihoyo', 'new albedo', 'quest lines', 'cryo accident']	['3909 rifthounds whopperflowers', '2018
6	https://genshin-impact.fandom.com/wiki/Geo	geo	['geo vision', 'albedo cryo', 'itto part', 'khemia life', 'lay lines']	['itto 10351 christmas', '2396 albedo bette
7	https://genshin-impact.fandom.com/wiki/Gorou	gorou	['gorou gorou', 'itto part', 'albedo notes', 'hangout event', 'voice actor']	['best boy 6949', 'sleeping gorou 8215', 'tr
8	https://genshin-impact.fandom.com/wiki/Klee	klee	['albedo klee', 'big brother', 'new dragonspine', 'alice beginning', 'solitary confinement']	['3447 hurt albedo', 'klee destroy 9246', 'a
9	https://genshin-impact.fandom.com/wiki/Cryo	cryo	['albedo cryo', 'eula vision', 'archon mihoyo', 'albedos unknown', 'traveler multiple']	['2566 theorise albedo', 'cryo 3161 fun', 'w
10	https://genshin-impact.fandom.com/wiki/Eula	eula	['eula albedo', 'amber bennett', 'banner rosaria', 'star char', 'story line']	['2455 gangsta till', 'eula fighting whos', '3
11	https://genshin-impact.fandom.com/wiki/Signora	signora	['signora albedo', 'khemia life', 'delusion cryo', 'three people', 'fatui research']	['harbinger 2911 beat', 'maybe stole albed'
12	https://genshin-impact.fandom.com/wiki/Hu_Tao	hu tao	['hu tao', 'itto albedo', 'weapon banner', 'eula cryo', 'fates primos']	['12915 albedo rerun', 'mihoyololyall gotta
13	https://genshin-impact.fandom.com/wiki/Timmie	timmie	['timmie banner', 'birds dragonspine', 'snowman brain', 'albedo friends', 'pigeons fowl']	['birds play dragonspine', 'dragonspine bir

Figure 7: Entity and linking result

6.2 Histogram of named entity

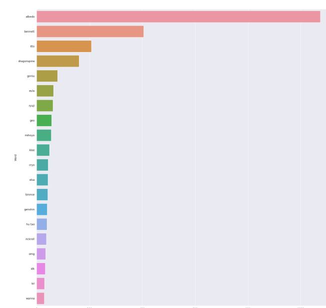

Figure 8: Histogram of named entity

6.3 Pie graph of sentiment categories

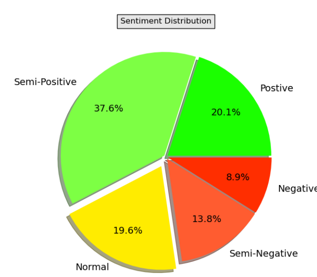

Figure 9: Pie graph of sentiment categories

6.4 Line graph of trend

6.5 CHI ENG Comparison

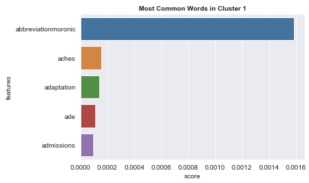

Figure 14: Cluster0 Result

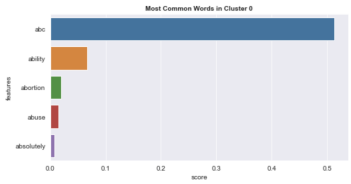

Figure 15: Cluster1 Result

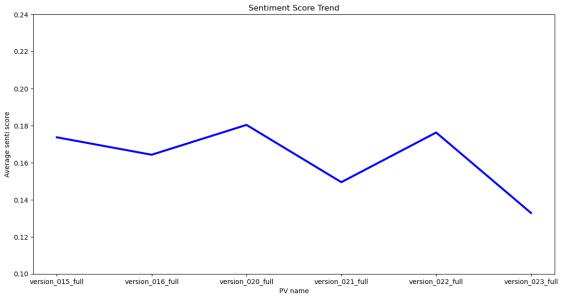

Figure 10: Line graph of trend

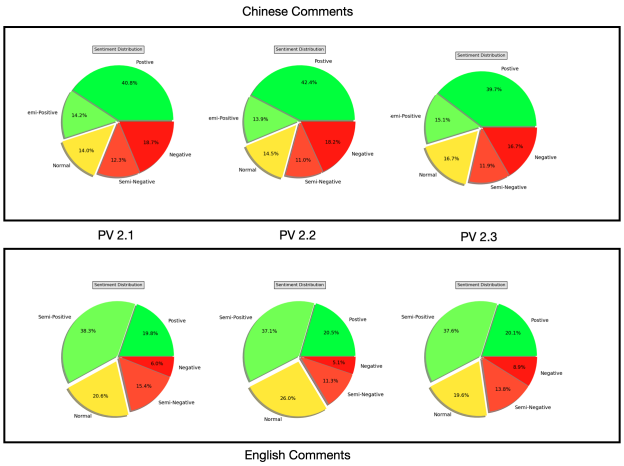

Figure 11: Comparison

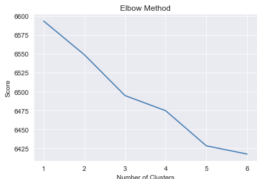

Figure 12: Cluster Score

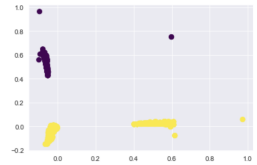

Figure 13: 2 Cluster Result

7 APPENDIX