Hyun-Woong & Woo Je, 2015

Volume 1 Issue 1, pp. 108-119

Year of Publication: 2015

DOI- https://dx.doi.org/10.20319/mijst.2016.s11.108119

This paper can be cited as: Hyun-Woong, C., & Woo Je, K. (2015). Development of a K-NN Model to Predict the Polarity of Korean Game Review Comments. MATTER: International Journal of Science and Technology, 1(1), 108-119.

This work is licensed under the Creative Commons Attribution-Non Commercial 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

DEVELOPMENT OF A K-NN MODEL TO PREDICT THE POLARITY OF KOREAN GAME REVIEW COMMENTS

Cho, Hyun-Woong
Dept. of SW Analysis & Design, Seoul National University of Science and Technology, Seoul, Republic of Korea
choman88@seoultech.ac.kr

Kim, Woo Je
Dept. of SW Analysis & Design, Seoul National University of Science and Technology, Seoul, Republic of Korea
wjkim@seoultech.ac.kr

Abstract

The purpose of this paper is to develop the machine learning model to evaluate game software in quantitative scale using opinion mining with the review comments which are provided by users in Korean. To do this, we first decompose the review comments into a lot of meaningful morphemes, and second construct a dictionary for opinion mining. Third, we develop a k-NN model to predict the polarity of review comment. Finally, we predict the polarity for each review comment which is included in validation data set by the model. The experimental results of the developed model are performed by the model which is implemented by JAVA and R language.

Keywords

Opinion mining, k-NN, Polarity, Game Review Comment

1. Introduction

Game software industry is growing continuously from mid-1990s. The game software 'Space War' was first made by MIT in 1961, and many kinds of game software's have appeared until now.

The 'Valve' company which was founded by Gabe Newell made Steam that was a platform for selling lots of game software's. Through Steam, a lot of game software shave been posted and sold to user. The number of users of Stream reached about 75 million peoples at the January of 2014.

Game users want to get information on specific game software before buying it. Normally there are three ways to get the information. The first way is to get the information provided by the company which has developed the game software. The second way is to get the metacritic score that is scored at some famous sites for game software evaluation. The third way is to get information from review comments which are posted in Steam by many users.

Steambb (http://www.steambb.com) is a website where a lot of users give review comments in Korean for the game software's which are sold in Steam. There are so a lot of review comments that it is difficult for users to read and understand all review comments. However, if we can summarize the review comments, then it may be useful and easy for users to decide to buy game software or for developers to get feedback for the game software which is developed by them. Actually there have been many studies to summarize the review comments, but there are few studies for analyzing review comments of game soft wares.

Opinion mining is one of techniques to summarize the review comments. In this paper, we will develop a machine learning model to evaluate game software in quantitative scale using opinion mining with the review comments which are provided by users in Korean. We will first decompose the review comment into many meaningful morphemes, and second construct a dictionary for opinion mining. Finally we will develop a k-NN model to estimate the polarity of the review comment by opinion mining.

2. Related Works

In (Kim et al., 2009), the authors analyzed the characteristics of internet language and figured out intended meaning of the review comments. They classified the review comments

according to polarity. Also, they automatically collected a corpus by positive and negative words, and they analyzed polarity of the review comments. This paper showed that its model had good performance on estimating polarity for positive review comments but not for negative review comments. It was because that the number of positive review comments were so greater than that of negative review comments.

In (Jang et al., 2015), the authors divided the review comments into 5 groups: very positive, positive, neutral, negative, and very negative. They analyzed the review comments by the dictionary that they made, and calculated the probability with which the review comment would be included in each group using Dempster-Shafer theory.

In (Shin & Kim, 2010), the authors extracted the characteristics of a particular product. Also, they built a dictionary by tagging POS (Part of Speech) to dictionary on ratings and review comments.

In (Song & Lee, 2011), the authors found that there were a lot of spacing errors and spelling errors in internet language. So they did not use morpheme analyzer. To overcome the problem, they developed some patterns between consonants in the review comments. Then they analyzed the review comments with the patterns.

Many studies on opinion mining have been done but there are few studies on opinion mining with the review comments for the game software. Therefore in this paper we will develop a machine learning models to predict the polarity with the review comments for game softwares, and show the performance of the developed model by experimental results.

3. Overall Procedure

Overall procedure to develop a machine learning model to predictthe polarity group with the review comments for game softwares is shown in Fig. 1.

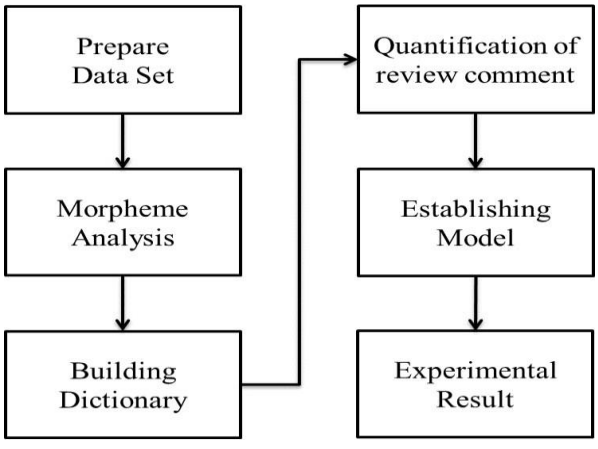

Figure 1: *Overall procedure*

First, we collected 17,182 review comments on Steambb from December 2012 to May 2015, and divided the review comments into learning data set (12,028, 70%) and validation data set (5,154, 30%). Each review comment had a rating score that was given by each user who wrote the review comment. Also, we divided the learning data set into three polarity groups (negative, neutral, positive). If its rating score is 1 or 2, it is included in the set of negative review comments. If the rating score is 3, it is the set of neutral review comments. If the rating score is 4 or 5, it is the set of positive review comments.

Second, the review comments were analyzed by using morpheme analyzer of which name was Hannanum (http://www.kldp.net/projects/hannanum) invented by KAIST (Korea Advanced Insititute of Science of Technology). Hannanum provides three kinds of POS (Part of Speech) tagger: POS 09, POS 22, & POS 69.

Third, we built a dictionary with the data sets which were provided by Hannanum. After the dictionary was built, we calculated the function value for each word included in the dictionary by TF-IDF (Term Frequency & Inverse Document Frequency) function.

Fourth, we developed a machine learning model to predict the polarity of review comment and let the model for the opinion mining learned with the TF-IDF function values and its rating score for the review comment.

Finally, we estimated the rating score for validation data set by the model and evaluated performance by comparing the value obtained by our model and the rating score provided by Steambb. Also, we improved the model by tuning some parameters.

4. Our Method

4.1 Morpheme analysis

To elicit morphemes from review comment, we used Hannanum morpheme analyzer. The Hannanum can decompose a sentence into several words and POS tags. It provides POS tag information that has three kinds of POS (09, 22, & 69). The POS 09 means that each word elicited from the review comment is classified into nine parts of speech.

Table 4.1: *Example of Morpheme Analysis*

	Morpheme
Review comment	완벽하다는 말은 이런 게임에 쓸 수 있는 단어 같습니다.
POS 09	완벽/N+하/X+다는/E+말/N+은/J+이런/M+게임/N+에/J+쓰/P+ㄹ/E+수/N+있/P+는/E+단어/N+같/P+습니다/E+./S
POS 22	완벽/NC+하/XS+다/EF+는/ET+말/NC+은/JX+이런/MM+게임/NC+에/JC+쓰/PV+ㄹ/ET+수/NB+있/PX+는/ET+단어/NC+같/PA+습니다/EF+./SF
POS 69	완벽/ncps+하/xsms+다/ef+는/etm+말/ncn+은/jxc+이런/mmd+게임/ncn+에/jca+쓰/pvg+ㄹ/etm+수/nbn+있/px+는/etm+단어/ncn+같/paa+습니다/ef+./sf

The nine parts of speech are as follows: S=syntax, F=foreign, N=noun, P= predicate, M=modifier, I=independent word, J=particle (This tag does not exist in English but in Korean), E=ending, and X=affix.

So we had a morpheme set by accumulating all of words which were elicited from all of the review comments by the Hannanum.

4.2 Dictionary building

After we applied the Hannanum solution to the review comments, we could have the first version of POS tagged dictionaries by POS 09, 22 & 69. However, there were a lot of morphemes that had no sentimental meaning in the first version of POS tagged dictionary. So we had to refine the first version of the POS tagged dictionary, and established the second version of POS tagged dictionary by eliminating morphemes which had no sentimental meaning.

For each word in the second version of dictionaries, we counted the frequency that the word appears in all of review comments.

POS 09

Word	POS	Frequency
판타지판	N	1
문명	N	6
라는	E	99
다렉	N	1
에서	J	484
구매했습니	N	2
..	S	472
여러가지	N	24
미치	P	22
못하	P	178
ㅂ니다	E	1178
대부분	N	23
4X	N	1
게임	N	2426
그러	P	135
겠지만	E	71
인터페이스	N	43
불편해서	N	2
시간	N	126
었네	E	50

POS 22

Word	POS	Frequency
판타지판	NC	1
문명	NC	6
다렉	NC	1
에서	JC	573
구매했습니	NC	2
..	SF	526
여러	NN	36
가지	NB	85
미치	PV	22
못하	PX	166
ㅂ니다	EF	1237
대부분	NC	23
4X	NC	1
게임	NC	2426
그러	PV	135
지만	EC	568
인터페이스	NC	43
불편해서	NC	2
시간	NB	154
유닛	NC	37

POS 69

Word	POS	Frequency
판타지	ncn	14
문명	ncn	6
다렉	ncn	1
에서	jca	595
구매했습니	ncn	2
여러	nnc	36
가지	nbu	85
미치	pvg	22
못하	px	166
ㅂ니다	ef	1237
대부분	ncn	23
4X	ncn	1
게임	ncn	2693
그러	pvd	135
지만	ecs	568
인터페이스	ncpa	43
불편	ncn	69
해서	ncn	57
시간	nbu	154
유닛	ncn	47

Figure 2: *Part of POS tagged dictionary*

After building the refined dictionary, we calculated the TF-IDF function value for each morpheme in the dictionary by the following equation.

$$TF - IDF\ (w, p) = r \times tf\ (w) \times log\ (\frac{N}{df\ (w)+1}) \qquad (1)$$

$$r = (\frac{the\ number\ of\ total\ review\ comments}{the\ number\ of\ relevant\ review\ comments}) \qquad (2)$$

Eq. 1 is the formula to calculate TF-IDF function value for each morpheme. However, the number of review comments by each polarity group (negative, neutral, positive) of the review comment are different. The TF-IDF function value can be affected by the number of review comments for each polarity group of review comment. So we needed to equalize the size of each polarity group of review comment, and introduced a weight factor '*r*' to TF-IDF function. The '*r*' means the ratio between total review comments and the relevant review comments. In Eq. 1, the '*tf(w)*' means frequency of word '*w*' which is shown in the relevant polarity group(*p*), and the '*df(w)*' means frequency of word '*w*' which is shown in other polarity groups. However, if the 'df *(w)*' is zero, the TF-IDF function value becomes infinite. To prevent it, we add one to '*df (w)*'. The '*N*' is total number of review comments.

For example, let us obtain the TF-IDF function value that the word ‘fun’ is included in the negative group. Then the ‘*tf(w)*’ can be calculated by frequency of word ‘fun’ on negative group, and the ‘*df(w)*’ can be calculated by frequency of word ‘fun’ on neutral group and positive group.

Therefore we can build the dictionary which has TF-IDF function value as the following Table 4.2.

Table 4.2: *Part of dictionary with TF-IDF function value*

Polarity Group	Word	POS	Frequency	TF-IDF
negative	구매했습니다만	N	10.3152	44.38499
negative	불편해서	N	10.3152	33.42851
negative	색감	N	20.6304	51.31039
negative	뭐랄까	N	30.9456	77.67911
negative	안되	P	577.6512	846.1155
negative	더욱	M	72.2064	126.7627
negative	어렵	P	747.852	744.4225
negative	매우	M	541.548	657.7399
negative	한글화	N	195.9888	296.0548
negative	안되서	N	46.4184	116.3694
negative	패치까지	N	10.3152	44.38499
negative	추천드	N	56.7336	123.953
negative	태블릿	N	10.3152	44.38499
negative	별로	M	464.184	740.7465
negative	나쁘	P	335.244	501.6523
negative	연관	N	25.788	73.612
negative	분위기	N	304.2984	383.736
negative	상당히	M	417.7656	483.0717
negative	긴장감	N	103.152	195.3867

4.3 Quantification of review comment

After building dictionary, we quantify the review comment by each polarity group. There are three polarity groups: negative group, neutral group, and positive group. For each polarity group, we quantify the review comment by summing TF-IDF function values of the words which are shown in the review comment. So we can obtain three values for each review comment as the following Eq. 3.

$$QRC_{i} = \left(\sum ngwi, \sum ntwj, \sum ptwj,\right) \quad (3)$$

QRC_i= Quantified value for review comment i

$Ngwi$= TF-IDF function value of word j which is included in negative polarity group and shown in the review comment$i$

Ntwi = TF-IDF function value of word *j* which is included in neutral polarity group and shown in the review comment*i*

Ptwi= TF-IDF function value of word *j* which is included in positive polarity group and shown in the review comment*i*

For example, we assume that a review comment is divided into the following morphemes: '재미' and '나쁜', and its TF-IDF function value by each polarity group is shown in the dictionary as the following Fig. 3.

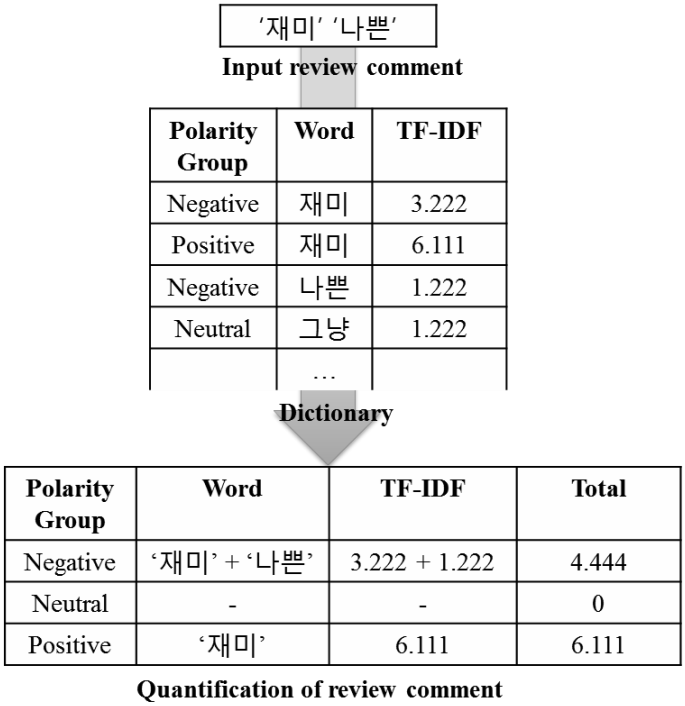

Figure 3: *Example process of quantification of review comment*

Then the word '재미' has TF-IDF function value for negative group and positive group, respectively. The TF-IDF function value of '재미' has difference on negative group and positive group. The function value is 3.222 on negative group and 6.111 on positive. Similarly, the function value of '나쁜' is 1.222 on negative group. So we can obtain *QRCi* as (4.444, 0, & 6.111).

We can quantify all review comments in the learning data set by calculating all *QRCi*'s. Therefore we will develop the machine learning model to predict the polarity group for review comments with *QRCi*'s and their rating scores.

4.4 Establishing Model to Predict Polarity Group

The review comment can be quantified as three-tuple by Eq. 3. So we can develop the machine learning model by letting the model learned with three-tuple and its rating score.

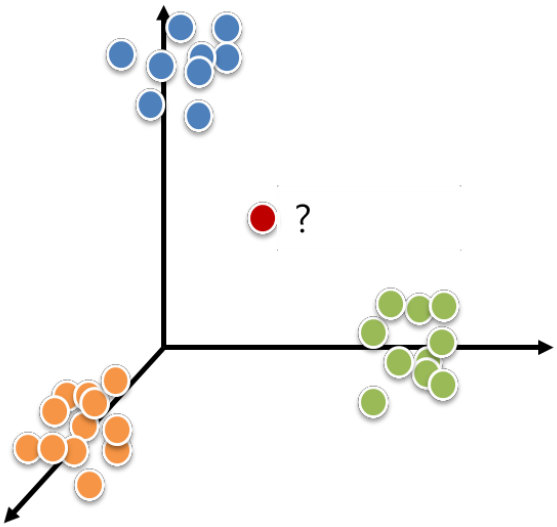

Figure 4: *Concept of machine learning model*

For the machine learning models, we developed a model based on k-NN (Nearest Neighbors). The k-NN is a classification method on machine learning, which estimates polarity group of the review comment by calculating distance between other points. The input consists of the k closest training examples in the feature space. The output is a polarity group. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors.

We developed a model based on k-NN. This model was learned with 12,028 review comments which was components of learning data set. Also, we evaluated the developed model with the validation data set.

The processes of morpheme analysis, building dictionary, and quantification of review comment were programmed by Java language in the model, and the machine learning model was programmed by R language.

5. Experiments and Results

After the model was learned with learning data set, we evaluated the developed model with validation data set. The model based on k-NN estimated the polarity group for every review comment in validation data set. Basically the k-NN model is to classify group based on the

distance. It classify group to refer to properties of the closest point of the input data. The best choice of *k* depends upon the data; generally, larger values of *k* reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good k can be selected by various heuristic techniques. In this experiment, the Euclidean distance and Manhattan distance are used to calculate distance, and maximum value for *k* was set to 100.

The experimental results are shown in Table 5.1. We calculated the average accuracy which means the probability that the estimated polarity group corresponds to the rating score of the review comment. This model had average 69.65% accuracy with POS 09 tagged dictionary, 69.40% accuracy with POS 22 tagged dictionary, and 69.47% accuracy with POS 69 tagged dictionary when the Euclidean distance was applied. The average accuracy was 69.53% with POS 09 tagged dictionary, 69.63% accuracy with POS 22 tagged dictionary, and 69.77% accuracy with POS 69 tagged dictionary when the Manhattan distance was applied. In Table 5.1, the Best-k means the optimum value for *k*.

Table 5.1: *Result of experiment on k-NN*

Distance	POS	Best-k	Accuracy
Euclidean	09	41	69.65%
	22	44	69.40%
	69	51	69.47%
Manhattan	09	58	69.53%
	22	74	69.63%
	69	68	69.77%

The k-NN model with POS 69 tagged dictionary and Manhattan distance had the best accuracy among all experiments.

6. Conclusion

In this paper, we developed the machine leaning model for predicting the polarity of review comments. First, we did morpheme analysis using Hannanum, and the POS tags and words were elicited from review comments. Second, we built the POS tagged dictionary which had TF-IDF function value. Third, we developed the machine learning model to predict polarity of review comment, which was a k-NN model. Finally, we predicted the polarity for each review comment which is included in validation data set by each model. We evaluated the performance of the developed modeland the k-NN model with POS 69 tagged dictionary and Manhattan distance had the best result.

For further work, it would be necessary to improve the prediction accuracy for the developed model. We found that there were not enough negative review comments which were collected from Steambb. The 1,555 negative review comments were too small to make decision comparing with the 8,020 positive review comments. Also, it needs to develop other machine learning models like a neural network based model and a SVM (Support Vector Machine) based model. So we will collect new data set from Steambb and other game sites, and develop somemachine learning models to predict the polarity of review comments. Furthermore, we will apply this research methodology to other domains such as home appliances, agro-fishery products, apparel, and personal credit, etc.

7. Acknowlegement

This research was supported by SW Master's course of hiring contract Program grant funded by the Ministry of Science, ICT and Future Planning.

REFERENCES

Kim, G., Lee, H., Yook, S., & Paik, W. (2009) Customer Preference Identification System using Natural Language Processing-based Analysis Korea Society for Information Management 16. 65-70.

Jang, K., Park, S., & Kim, W. (2015). Automatic construction of a negative/positive corpus and emotional classification using the internet emotional sign Journal of KIISE, 42(4). 512-521

Shin, J. & Kim, H. (2010) A Robust Pattern-based Feature Extraction Method for Sentiment Categorization of Korean Customer Reviews Journal of KIISE 37(12) 946-950

Song, J. & Lee, S. (2011). Automatic Construction of a Positive/Negative Feature-Predicate Dictionary for Polarity Classification of Product Reviews Journal of KIISE 38(3) 157-168.