Bigram Collocation Extraction of Mobile Game Comments

Jing Zhang, Haiqi Chen, Shoubin Dong

Communication and Computer Network Lab of Guangdong Province,

School of Computer Science and Engineering, South China University of Technology

Guangzhou, China

zhjing@scut.edu.cn

Keywords: Mobile game comments, 2-dimensional attribute lexicon, comment collocation, sentiment tendency analysis

Abstract: Mining user opinions about game attributes from comment data becomes key to both the player and the game developers. As it's impractical to take the user score as category information for the training set, a solution based on Co-training and Pu-training is proposed to gain the training corpus for the classifier. A concept of 2-dimensional attribute lexicon and a cluster-based construction solution are proposed. A dependency tree-based comment collocation template extraction method is proposed with the training corpus and 2D attribute lexicon as knowledge base. Experiment shows comment collocations as "picture–fabulous" and "teammate–really lame", which leads to satisfactory accuracy and recall.

1. Introduction

Mobile games consumers would share comments via websites. Sentiment tendency recognition, product attribute extraction and solutions to user demand analysis based on user-generated contents are proposed in the opinion mining field.

Stochastic methods are proposed for sentiment tendency recognition to mine the positive, negative or neutral attitudes of the users [1]. Zhu et al [2] utilizes synonyms and antonyms of adjectives from HowNet to judge the tendency of a word. Wang [3] judge the sentiment tendency on word or sentence level.

Product attribute extraction is an important task for demand analysis based on user sentiment data. From cooccurrence rules between product attributes and attribute comments, Zhu [2] proposes an interactive method for product attribute extraction to produce a series of nouns and noun phrases from association rules. The nouns and noun phrases are then pruned into a product attribute list. Tang [4] utilizes supervised learning to extract product attributes from online comments.

Mobile game comments are processed from multiple aspects to extract attributes. Co-training is used for collocation extraction. Pu-learning is adopted for the sparse effective labels in the data set. The purpose of our research is to recognize useful comments from multiple views about the games.

Mobile game comments are far more complex than user scores. The features of game comments are analyzed in section 2. A mobile game comment corpus labeling method is proposed in section 3 based on co-training and pu-learning, which generates training corpus from game comments. A game comment sentiment analysis model based on NB-LR model is proposed and applied to experiments in section 5. A comment collocation extraction method is proposed in section 6 based on dependency syntax tree.

2. Features of mobile game comments

230 million pieces of comments are taken as mobile game comment corpus in the research. Analysis of the corpus leads to discovery of features of mobile game comments, as follows: 1) The text length varies between 25~50 characters, so we take game comment analysis as short text analysis. 2) The user scores are not reliable, so labeling of training corpus needs to be solved. 3)

  DOI: 10.25236/icemit.2019.025

Comment tendency analysis only is not enough to know user experience.

The corpus contains 150 million pieces of user-scored (1~5) comments and is about 1/3 of the whole corpus. Usually a score of 1 or 2 are taken as negative, and 4 or 5 as positive. Based on analysis of the corpus, we know that user scores do not always correspond with the comment sentiments. A random sample of 5000 pieces is shown in Table 1.

Sentiment tendency analysis expresses the overall attitude of users, but user comments usually have multiple aspects. Comment collocation is the main topic of our research. From granularity of analysis, comment collocation is more in-detail than sentiment tendency analysis, which involves recognition of user comments on various attributes of games, and user sentiment tendency towards the attributes. A creative division of attribute dimensions into 2 levels is proposed in the paper.

3. Label the training corpus

For the different results of user scoring and human labeling, further processing of the training corpus is made in the following 5 steps.

Step 1: Human labeling of small samples. Extract some comments from the unlabled corpus (UnlabelCorpus) and make human labeling.

Step 2: Extract negative comment labeled data (NegCorpus) from the comments to enlarge the negative labeled sample:

Rule 1. train a logistic regression classifier with the corpus obtained in step 1. Classify all the comments with user score of 1 or 2, and choose the negative comments with probability above the threshold as negative data.

Rule 2. Traverse all comments, extract comments with adversatives such as “but”,” “though” etc., and divide the comment text into two short sentences. Apply rule 1 again to the two short sentences. Choose the negative comments with probability above the threshold as negative label data.

Table 1 User score and sentiment label statistics

Human Label	**User Score**		
	1-2	***4-5***	***Sum***
Positive	76	2166	2242
Negative	1510	847	2357
Sum	1586	2013	4599

Here, Rule 1 associates user scores with classifier prediction value, and adjustment of the threshold value leads to the choice of negative comments. Reliable negative labeling data is obtained in this way.

Rule 2 deals with negative comments with adversative relations. Both positive and negative attitudes co-occur in such comments, and the classifier in rule 1 can judge the two short sentences and take the negative short sentence as negative labeled data, e.g. “funny, but no pets and many monsters”.

Step 3: PU-Learning [5] is applied to construct the positive comment sample to enlarge the positive comment label sample.

Traverse all comments a second time after the negative samples are acquired in step 2, thus to acquire positive samples. The PU-learning has two stages. The first stage of 1-DNF is to acquire small scale of positive samples, and the second stage of PEBL [6] is to expand the positive samples.

Stage 1: 1-DNF algorithm segments the UnlabelCorpus and NegCorpus into words, and calculates the occurrence probability of each word in the NegCorpus and UnlabelCorpus. Rank the words in decreasing probability order, and take the first 2000 as feature words of the NegCorpus.

Traverse the UnlabelCorpus, select all comments with a user-score of 5 and do not include any of the 2000 feature words as positive comment data (PosCorpus).

This strategy aims to exclude comments with user-score of 5 out of the negative corpus, and take the comments of score 5 and without negative words as positive data.

Stage 2: Train a bi-gram classifier using the PosCorpus and NegCorpus as the training set. The

classifier is then used to label the UnlabelCorpus. PosCorpus is enlarged with the labeled positive comments. This stage is processed iteratively until no more positive comments are labeled.

Step 4: Acquire more positive and negative comments with Co-training to expand the comment samples.

Small amounts of positive and negative comments are acquired in step 2 and step 3, which are then used as labeled data to train the two classifiers by iterative Co-training. The UnlabelCorpus data are classified and the N most confident samples are added to PosCorpus and NegCorpus separately.

The Co-training algorithm is used to train 3 classifiers for UnlabelCorpus. The 3 classifiers are logistic regression model with word-bag as features (BowLRModel), logistic regression model with sentence vector as features (Snt2vecLRModel), and naïve bayesian classifier based on polynomial model (NaiveBayesModel). The pseudocode is shown in Fig.1.

Step 5: Random sampling verification. A large scale positive and negative sample is acquired after the processing above. 3 random sampling processes are made to extract 2000 comments each time from the PosCorpus and NegCorpus. The precision of the automatic labeling is human-verified.

```
= CoTraining(PosCorpus，NegCorpus，UnlabelCorpus):
    Set T1, T2, and T3 as the thresholds of the three models of BowLRModel,
      Snt2vecLRModel and NaiveBayesModel.
  Set K as the maximum iteration number.
  Initialize i=1; Remain=UnlabelCorpus
  While i<K and the Remain set is not empty, iterating the following steps:
      Using the PosCorpus and NegCorpus as the training set to train the 3 models
      Randomly choose M samples from Remain as UnlabelSample;
      Using the 3 models to classify the samples in the Remain set:
          = predict(UnlabelSample,BowLRModel) if the prediction probability is
            greater than T1;
          = predict(UnlabelSample,Snt2vecLRModel) if the prediction probability is
            greater than T2;
          = predict(UnlabelSample,NaiveBayesModel) if the prediction probability is
            greater than T3;
        Move the labeled samples from the Remain set to PosCorpus and NegCorpus
            according to the label result.
```

Fig. 1. The pseudocode of Co-Training

4. Comment sentiment analysis with NB-LR

PosCorpus and NegCorpus are acquired as labeled comment corpus, and divide into two classes based on NB-LR [7], which integrates naïve bayesian model and logistic regression for classification. The classification steps are as follows: segment all comment samples into words, and denote the text in unigram-bigram language model. For example, the sentence “this game is Ceezy, too many ads, sick”, the word sequence is acquired as follows:“this, game, Ceezy, ads, too many, sick, this_game, game_Ceezy, too many_ads”.

After the processing above, comments are vectorized with the word-bag model into the liblinear format. Liblinear is used to train the logistic regression model with L2 normalization. The comment is classified as positive with a probability above 0.55, negative below 0.45, and neutral between the two values.

5. Extract comment collocation

Sentiment tendency analysis reflects overall attitude of users. But the comments usually have multiple aspects. Text analysis is made on finer granularity for further information. Collocations are extracted from texts, and used to recognize the users’ sentiment tendency towards the various aspects of the games. Comment collocation extraction provides 5-tuples of <1-dimension, 2-dimension, feature word, sentiment word, sentiment tendency>, and gets comments on the games from various dimensions. For example, a comment “the match ranking is sick”, can have <game play design, grading, ranking, sick, negative> as comment on “ranking” of the game. This is a finer processing of

sentiment analysis and can lead to better understanding of the problems in the game operation.

5.1 Definition of dimension features and generation of sentiment lexicon

Dimension is used to denote the attributes of user comments, which can be divided into 1-dimension and 2-dimension. Dimension attributes are acquired as follows: Step 1. Pre-processing. Acquire all sentences with sentiment words, segment and part-of-speech tag the sentence.

Step 2. Calculate the probability of nouns in the sentence, extract nouns with frequency above the threshold, and get a word frequency lexicon D.

Step 3. For each word w in D, train with word2vec to acquire the word vector, put every word with similarity above 0.45 with the noun w into D. Iterate maximum times.

Step 4. K-means cluster the words in D. Set the number of cluster centers and human filter the center words. Process the words and define 1-dimension and 2-dimension classification system, as shown in Table 2. The sentiment lexicon is generated by combing the 1-dimension attributes and the 2-dimension attributes.

5.2 Extract collocation template with dependency tree

Dependency parsing is used to analyze the structural dependency relation between the words, i.e., subject-predicate relationship, coordinating relationship etc. Collocation extraction based on dependency parsing is as follows:

Step 1. Make dependency parsing of the sentences with FudanNLP, and construct dependency path template. For example, “This game is not really interesting.” can have the dependency tree.

Step 2: Convert the directed acyclic graph into a syntax graph. As the directed acyclic graph is not convenient for path traversal to find relationship between sentiment and attribute words. A syntax graph as in Fig.2 is constructed by the following process: add a path from son-node to father-node for each dependency relation on the syntax tree, and label as “inverse relation”.

Step 3: Traverse all the m sentiment words in the sentence, and acquire all the n nouns routable from the sentiment word, and construct <noun, sentiment word> bi-gram. For data sparseness of the corpus, construct bigrams with labels of dependency syntax. Then count the frequency of all bigrams, rank and extract top 60 collocations.

Table 2 The 2-Dimension Attributes Of User Comments

1-DIMENSION ATTRIBUTES	2-DIMENSION ATTRIBUTES
APPEARANCE	CHARACTER, PAINT, MUSIC, THEME
PLAY DESIGN	TASK, MAP, PETS, SKILL, WEAPON, TOOL, ACCOUNT, SETTING
NETWORK PROBLEMS	CLIENT, DATA, SERVER, EXCEPTION, LOGGING, VERSION, DEVICE PROBLEM, NETWORK DELAY
OPERATION PROBLEMS	SERVICE, FREEZE, ADVERTISEMENT, FORUM, LOTTERY, DISCOUNT
…	

Some of the templates acquired from bigram count have low precision, so human labeling of hundreds of pieces of data are made. Dependency path template list is acquired by precision and used in later calculations. Top 10 templates with the highest precision are listed in table 3.

5.3 Template-based comment collocation extraction and polarity judgement

Based on dependency templates mentioned above, Three steps are made to extract collocation and

to judge the sentiment polarity of collocations.

Step 1: Extract sentiment words and nouns. If the dependency bigram is included in dependency path template, extract the pair of noun and predicate.

Step 2: Judge the dimension of the attribute word. After extracting the collocation of <sentiment word, noun>, look up the 1-dimension and 2-dimension attributes from the attribute lexicon corresponding with the noun. If it doesn't exist, discard the collocation.

Step 3: Judge polarity of the comment collocation. Polarity of the comment collocation has the following two conditions:

1) Sentiment words have polarity characteristics, while attribute words do not have clear polarity. For example, in "game-interesting", the attribute word "game" does not have a clear polarity, while "interesting" has a positive polarity.

2) Sentiment words have an unstable polarity, while attribute words have implicit polarity characteristics. For example, the word "many" has unstable polarity, which is positive in "reward-many", while negative in "ads-many". The sentiment analysis model in section III is used to predict sentiment tendency of the comment collocation.

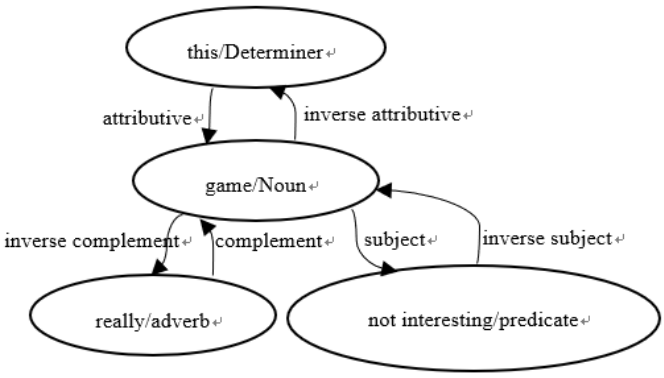

Fig. 2. The syntax graph of "this game is not really interesting"

Table 3 TOP 5 Bigram Collocations

Bigram collocations	Precision
[word predicate] complement [verb] word structure [structure auxiliary] reverse punctuation [noun]	92.82%
[form predicate] adverbial [form predicate] inverse subject [noun] inverse subject [noun]	92.78%
[word predicate] word structure [structure auxiliary] complement [verb] inverse object [noun]	89.50%
… …	86.83%

Step 4: SVM-based comment collocation filtering. Accuracy of collocation extraction with dependency path template depends on the syntax path. Low accuracy of the syntax path may lead to low performance in the next step. A SVM classifier is used for two-class modeling of collocation extraction to filter inaccurate comment collocations.

1) Training sample acquisition. 10,000 comment collocations are randomly selected and human labeled to serve as positive and negative samples. Positive samples (class 1) refer to accurate extractions, and negative (class 0) refer to inaccurate extractions.

2) Feature selection. 17 features are selected from attribute-sentiment word relation, context information, distance and other aspects. Then train the SVM model. Vectorize features from the training samples with libsvm opensource tool, and select optimal parameters through cross validation.

3) Predict result of comment collocation extraction. Predict the sentiment polarity of the candidate collocations <attribute word, sentiment word> acquired above.

6. Experiment and analysis

6.1 Automatic labeling of training corpus based on Co-training and Pu-learning

3052157 positive comments and 3094430 negative comments are acquired with automatic labeling of the training corpus. Average accuracy of 95.2% is acquired through 3-fold sampling verification. Train the model 5 times using the samples with a 9:1 training to test ratio. The cross verification average accuracy is 98.36% for the trained NB-LR model.

6.2 Experiment on comment sentiment polarity classification based on NB-LR modeling

5000 pieces of mobile game comments (out of the PosCorpus and NegCorpus) are randomly selected for evaluation. Polarity are human labeled. Accuracy and F-measure are evaluated for practicability of the sentiment tendency analysis modeling in the mobile game domain.

The baseline method (TF-IDF + LR) based on automatic labeled corpus is compared with the NB-LR method based on user-scored corpus, as shown in table 4.

Table 4 shows that the NB-LR model based on automatic labeled corpus outperforms the NB-LR model based on user-scored corpus, and also outperforms the baseline method. The result verifies the fact that a sentiment classifier trained on weak supervision labeled results of user-score evaluation cannot reach ideal performance. The NB-LR model trained on basis of automatic labeled corpus reaches an accuracy of 96.4%. The model satisfies engineering demand.

6.3 Experimental results of comment collocation extraction

Take the popular game “glory of lord” as an example. Important comment collocations such as “game-interesting”, “game-rubbish”, “picture-super”, “teammate-bad” etc. can be extracted from one year’s contents from a game seminar.

Table 4 Sentiment tendency Accuracy of models

Models	Accuracy	F-Score
NB-LR	96.40%	98.50%
TFIDF+LR	90.00%	91.20%
NB-LR(User-Scored)	68.42%	76.50%

Precision and recall are used to evaluate the extraction results. Test samples are 1000 comments randomly extracted and human labeled from mobile game comments. The result reaches 78.4% of precision and 82.5% of recall.

7. Conclusion

Two aspects are studied about sentiment analysis of mobile game comments: tendency analysis and collocation extraction of comments.

For comment tendency analysis, training set is acquired by automatic labeling of samples, and then to train sentiment tendency classifier with the NB-LR model. Experiment results show an effective sentiment classifier of game comments based on the trained model.

For comment collocation extraction, a sentiment lexicon is constructed, 2-dimensional attributes are defined, comment collocations are extracted on basis of dependency analysis. The collocations are filtered with a SVM model. Experiment results show effective comment collocation extraction based on the proposed algorithm.

Many human labelings are used in the work, which will be further automatized in future work for recognition of attributes and polarity analysis of game comments.

Acknowlegement

The work is sponsored by CERNET Innovation Project (NGII20160127).

References

[1] B Liu. Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies[J].2012 1-167

[2] Zhu Yanlan , Min Jin ,Zhou Yaqian ,Huang Xuanjing ,Wu Lide, Semantic Orientation Computing Based on HowNet. Journal of Chinese information processing[J], 2006, 20(1):14-20

[3] WANG Changhou, WANG Fei. Extracting sentiment words using pattern based Bootstrapping method. Computer Engineering and Applications, 2014, 50(1):127-129.

[4] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, Bing Qin. 2014. Learning Sentiment Specific Word Embedding for Twitter Sentiment Classification[J]. Proceeding of the 52th Annual Meeting of the Association for Computational Linguistics

[5] Fusilier D H, Montes-y-Gómez M, Rosso P, et al. Detecting positive and negative deceptive opinions using PU-learning[J]. Information processing & management, 2015, 51(4): 433-443.

[6] Mesnil G, Mikolov T, Ranzato M A, et al. Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews[J]. arXiv preprint arXiv:1412.5335, 2014.

[7] Ahmad A, Dey L. A k-mean clustering algorithm for mixed numeric and categorical data[J]. Data&Knowledge Engineering, 2007, 63(2):503-527.