APPLYING OPINION MINING IN E-LEARNING

Thao N.T. Le
Ton Duc Thang University, Ho Chi Minh City

ABSTRACT

Recently, we have witnessed an explosive growth of e-learning. More and more learning contents have been published and shared over the Internet. "Putting the Learner *at the* Center" *is the next-generation innovation emphasizes enabling learning in* e-learning *environments. Therefore, how to know learner's opinion on the course-wares, the teachers, the charge or something else of the e-learning system to aim to progress an efficient e-learning process becomes a critical issue.*

This paper proposes applying the OFMM (Opinion Features Mining Model) to determine the personal learner's opinions from learner's reviews.

Keyword: *opinion mining, feature-based summary, e-learning.*

I. INTRODUCTION

The fast development of Internet and Web technologies has strongly changed the way that people express their opinions. Especially in e-learning, learners usually post their reviews of courses to the websites as a common way to discuss or describe their personal opinions. As e-learning is being more and more popular, the web that contains a wealth of learner's reviews becomes resources with valuable information for e-learning system. Ideally, there is a system would mine the opinions of learners on features of e-learning courses automatically. This system is performed in two tasks:

1. Mining the features of course that learners have expressed opinions on (called opinion features).

2. For each feature, identifying whether the opinion from each reviewer is positive or negative.

There are three main review formats on the Web. Different review formats may need different techniques to perform the above tasks.

Format (1) - Pros and Cons: The reviewer is asked to describe Pros and Cons separately.

Format (2) - Pros, Cons and detailed review: The reviewer is asked to describe Pros and Cons separately and also write a detailed review.

Format (3) - Free format: The reviewer can write freely, i.e., no separation of Pros and Cons.

For formats (1) and (2), opinion orientations (positive or negative) of features are known because Pros and Cons are separated and thus there is no need to identify them. Only course features that have been commented on by learners need to be identified. For format (3), we need to identify both e-learning course features and opinion orientations. Reviewers typically use full sentences.

However, for format (2), Pros and Cons tend to be very brief. For example, under Cons, one may write: "heavy, bad picture quality, battery life too short", which are elaborated in the detailed review.

In this paper, we only focus the first task of the feature-based opinion mining system. That is, we aim to mine the features of e-learning course from learner's review in format (3). Towards this goal, we propose the **OFMM** (**Opinion Features Mining Model**) with divide and rule strategy. Features are divided into two subsets: frequent features set and infrequent features set. This division makes mining opinion/e-learning course features with appropriate techniques more effective.

> **GREAT Course**, Jun 12, 2007
>
> Reviewer: **Nick S.** from Missouri
>
> The courses have given me an opportunity to expand my knowledge in several technical areas. Examples are the UNIX operating system and Oracle database. A specific goal of mine was reached, aided by completion of the appropriate on-line courses.
>
> This month I took the Microsoft test and received my certification as a Microsoft Office User Specialist Expert for MS Access. You could say that makes me a MOS E. Humor aside, this certification validates my understanding of this application and adds to the arsenal of tools I can use in support of customers.

Figure 1. *A sample learner's review in format (3)*

2. RELATED WORK

In recent years, there are many researches focus on studying solution to opinion feature mining problems.

Zhuang et al. [10] used WordNet "IMDB" labeled training data to produce feature and opinion keyword list. And use the keyword list to find out feature terms and opinion terms. Besides, determine each opinion word is positive or negative. Then they used pattern mining to find feature-opinion pattern. And they used these patterns to produce review summary.

Kusumura et al. [11] generalized 13 type feature and 37 feature-value pair. Then, they used a supervised method to find frequent patterns. After that, they used the patterns to find the feedbacks contain real feeling and eliminate courtesy description.

Riloff, E. and Wiebe, J. [12] presents a bootstrapping process that learns linguistically rich extraction patterns for subjective (opinionated) expressions.

Wilson et al. [16] used a wide range of features to classifying the strength of opinions and other types of subjectivity and classifying the subjectivity of deeply nested clauses.

Hu and Liu [1] propose several methods to analyze customer reviews of free format. They perform the same tasks of identifying product features on which customers have expressed their opinions and determining whether the opinions are positive or negative.

Popescu and Etzioni [3] depend on compute PMI scores between review's phrase and discriminators associated with product to extract features and their feature extraction results is higher than Hu and Liu's work, besides, they also test the semantic orientation precision and recall.

Our research based on deeply Liu and Hu's work to improve some techniques and propose a model – the OFMM (Opinion Feature Mining Model) with divide and rule strategy. We apply this model to identifying course features on which learners have expressed their opinions in e-learning system.

3. OPINION FEATURES MINING MODEL (OFFM)

3.1. Problem definition [1]

Let $R = \{r_1, r_2, \ldots, r_n\}$ be a set of learner's reviews on a object O. In free format, each review r_i consists of sequence of sentences $r_i = <s_{i1}, s_{i2}, \ldots, s_{im}>$.

Definition (object): An object $O: (T,A)$ is an entity which has a set of components T and a set of attributes A . Each components has its own set of sub-components and attributes.

In general, the opinions can be expressed on anything, e.g., a course, a product, an individual, an organization, an event, a topic, etc. We use the general term "object" to denote the e-learning course that has been commented on.

Example: Object is a SQL course. It has a set of component, e.g., tutorials, lectures, assignment, technical support, class tools, content, teacher, etc., and a set of attributes, e.g., course title, credits, syllabus, timetable, cost, etc. The lecturer component also has its set of attributes, e.g., pedagogical methods, enthusiasm, preparation, practical skills, expertise, etc.

Object O has a limited set of features $F = \{ f_1, f_2, \ldots, f_m\}$. F is considered as the "minimalist" features set of the object O.

Definition (feature): A feature f in a review r_i is an attribute/component of object O that has been commented on in r_i.

Example: Object is a SQL course. One can express an opinion on the object (feature is course), e.g., "I don't like this course", or on its attribute (feature is syllabus), e.g., "the syllabus of this course is so poor, not meet the requirements in practice". Likewise, one can also express an opinion on one of object's component (feature is tutorial) e.g., "The tutorials are extremely well produced " or on the attribute of the component (feature is enthusiasm), e.g., "I appreciate the enthusiasm of this teacher".

Definition (explicit and implicit feature): If a feature f appears in a review r_i, it is called explicit feature in r_i. If f does not appear in r_i but is implied, it is called implicit feature in r_i.

Example: "SQL course" is an explicit feature in sentence "The SQL course is easy to follow", "layout" is an implicit feature in sentence "It's hard to find lessons in the content page".

Each feature f_i in F can be expressed through a limited set of synonyms W_i.

Let $W = \{ W_1, W_2, \ldots, W_m\}$ where $W_i = \{ w_1, w_2, \ldots, w_p \}$.

In a learner's review d_i, customers i will comment on the object O through a set of feature S_i, where $S_i \subseteq F$.

$\forall$ feature $f_j \in S_i$ which is commented on by learner i in the review d_i, f_j can be expressed through words/word phrases $w_k \in W_j$.

Our task: Given set D with the set F is unknown, should explore the set W. Means we need find out all the features of course are learners mentioned in the reviews in free format.

3.2. The Proposed Model

Our work is based on published works to improve and propose a model – the OFMM (Opinion Feature Mining Model) with divide and rule strategy. Features are divided into two subsets: frequent features set and infrequent features set. This model includes 4 main steps:

Text pre-processing on review database

Frequent feature mining

Opinion keywords extraction

Infrequent feature mining

The model overview is listed on following figure.

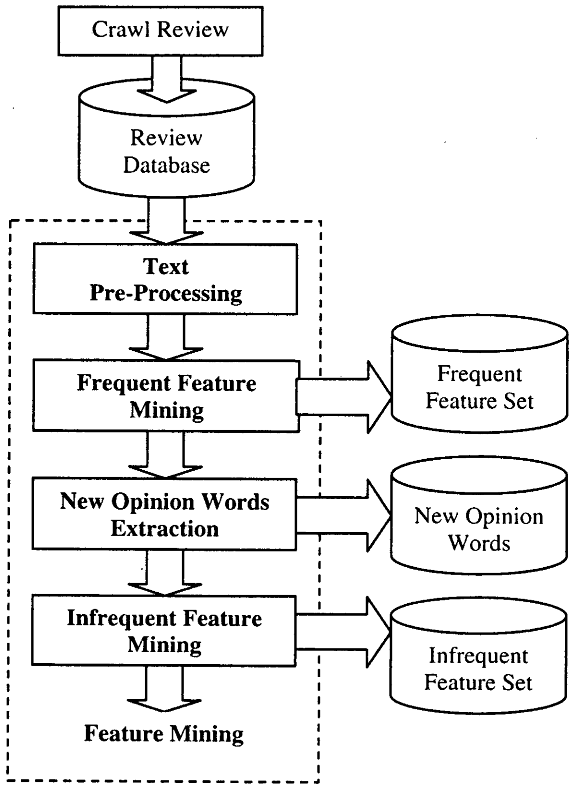

Figure 2. *Model architecture of OFMM*

3.2.1. Text pre-processing

The online student feedbacks are stored in raw text database. In the first step, we applied natural language processing techniques in order to standardize data for next steps. Text pre-processing includes three sub steps:

Part-of-speech tagging (POS)

Identifying and removing stop words

Word stemming

We use the NLProcessor[3] linguistic parser to generate the POS tag of each word (whether the word is a noun, verb, adjective, etc). Each feedback is saved in database with its POS tags. The result of this step is a XML file. After tagging all the items, we would like to identify and removing stop words to decrease information disturbance in feature mining process. We use an available stop word list to compare and delete stop words in sentences. This stop word list is order of dictionary. We also perform word stemming technique with Porter Stemming[4] algorithm.

3.2.2. Frequent feature mining

Frequent features are course features which many learner are interested in. So, frequent features are important and need be identified first.

Initializing list L_1 contains nouns extracted from the data source;

minsup=0.01;

// n is the number of sentences in database

for *i=0, ..., L_1. count-1* ***do***

  if *Count(L_1[i]) $\geq$ minsup*n* ***then***

    Add L_1[i] $\rightarrow$ frequent feature set L_1

  end if

end for

k=2;

while *$W_{k-1} \neq \varnothing$* ***do***

  Generate candidate set L_k;

  for *i=0, ..., Lk . count-1* ***do***

    if *Count(L_k [i]) $\geq$ minsup*n* ***then***

      Add L_k [i] $\rightarrow$ feature set W_k;

    end if

  end for

end while

$W \leftarrow W_1 \cup W_2 \cup W_3$

$W \leftarrow$ Pruning (W);

Figure 3. *Frequent features mining algorithm*

Frequent feature generation

The method we used is based on Liu et al [2]. The concept of their work is that they thought the frequent noun/noun phrases in the feedback are important features which people

[3] http://www.infogistics.com/

[4] http://tartarus.org/~martin/PorterStemmer/

care about. They used association rule mining technique to find these features. However, they did not use associate rule, they just found the frequent itemsets.

The algorithm we used is Apriori [7] to find all frequent itemsets. And we just keep the noun/noun phrase because of the target feature is almost noun/noun phrase. In our context, an itemset is a set of words or a phrase that occurs together in some sentence. Association rule mining is stated as follows:

$I = \{i_1, i_1, \ldots, i_n\}$ be a set of items

D is a set of transactions (the dataset).

Each transaction consists of a subset of items in *I*. An *association rule* is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ holds in *D* with confidence *c* if *c*% of transactions in *D* that support *X* also support *Y*. The rule has support *s* in *D* if *s*% of transactions in *D* contain $X \cup Y$. The problem of mining association rules is to generate all association rules in *D* that have support and confidence greater than the user specified minimum support and minimum confidence.

It finds all frequent itemsets in the transaction set and consider the word sequence. We set minimum support as 1% (i.e. an itemset as frequent if it appears more than 1%).

After using Apriori algorithm, we generate a frequent feature set, we call that is candidate feature set, but not all the frequent features are real features,. There are also some redundant features and ones we are not interested in. Candidate feature set is divide into two groups. The first is nouns, which we call *single features* and the second is noun phrases, which we call *feature phrases.* We should reduce candidate feature set by appropriate pruning methods for each group.

Surrounding pruning

This method checks features that contain at least two words, which we call *feature phrases*, and remove those that are likely to be meaningless. Although we consider the sequence of the word in the sentence, because of that we remove stop words, there are many words in the original sentence are removal. Hence, an itemset has two words like "course, course outline" may appear in candidate features, but it is not the feature which we are interested in. The distance of this kind of feature in original sentence is usually too far. The distance is means how many words are there between the words of the candidate feature. Thus, we could prune this kind of feature by set a word distance threshold. If we find this kind of feature, and the word distance is more than threshold, we will prune it. For example, there is a frequent feature phrase "course, course outline". There are some sentence contain the phrase: S1: "The course is good, but the course outline is built not sufficiently careful." S2: "I like this course very much, and also the course outline ."

The word distance of S1 and S2 are 4 and 5, respectively. Thus, based on the rule we mentioned before, this phrase will prune. But if we can find at least two sentences that the word distances are small than threshold, then we keep this feature. In this work, we set the threshold as 3.

Redundancy pruning

This method checks features that contain only one word, which we call *single features.* In some case, a single frequent noun feature may be a redundant feature. However, the superset of this frequent noun feature may is an interesting feature. For example, a feature phrase "front desk" is a meaningful feature of a course of hotel management. But we may also identify the term "front" and "desk" as candidate features. However, these two features are not interesting features if they appear alone.

Hence, we used pure support defined by Hu's work [1] to count this kind of feature frequency. **P-support** (pure support) of a feature *ftr* is the number of sentences that *ftr* appears in as a noun or noun phrase, and these sentences must contain no feature phrase that is a superset of *ftr*.

In the case of "front desk" with support 7, and the support of "front" and "desk" are 9 and 11, respectively. Hence, the p-support of "front" and "desk" are 2 and 4, respectively. If a feature's **p-support** is lower than the minimum p-support (we set minimum p-support as 8) and the feature is a subset of another feature phrase, this feature will prune. Hence, in the previous example, the features "front" and "desk" pruned.

3.2.3. Opinion keywords extraction

After two steps above, we have frequent features set. However, apart from features which in many learner are interested, there are other features occur in learner's reviews. It is infrequent feature. Infrequent feature is little mention in the review, but it still is considered a feature of the e-learning course.

```
Created list of opinion words O from available data source;
Inspect set T;
for i=0, ..., n do
    if S_i ⊃ w ( w ∈ W) then
      if sentence S_i contains adjective then
         t:= adjective from nearest w in S_i;
      else if  S_i contains adverb then
         t:= adverb from nearest w in S_i;
      end if
    end if
end for
for j=0, ...m do
    if Count(t_j)≥2 and t_j ∉ O then
      Add t_j → O;
    end if
end for
```

Figure 4. *New opinion words extracting algorithm*

Apriori algorithm we use in step 2 will not discover the infrequent features because blocked by threshold minsup. Therefore, we use other methods to identify and exploit infrequent features. It is based on the opinion keywords. Opinion keywords are words used to express the opinions of each individual on an object or problem.

We recommend using a list of available opinion keywords, called a K. Initially we set K is constructed by manual methods. Then, knowledge of this list will be accumulated over time.

Bing Liu et al. [2] observed that people often express their opinions of a product feature using opinion words that are located around the feature in the sentence. We can extract opinion keywords from the review database using all the remaining frequent features (after pruning). For each sentence in the review database, if it contains any frequent feature, extract the nearby *adjective*. If such an adjective is found, it is considered an opinion keyword. A nearby adjective refers to the adjacent adjective that modifies the noun/noun phrase that is a frequent feature.

We use this result and suggest adding the extract the nearby adverb within the sentence. For each sentence in the database, if the sentence contains frequent features, we extract adjectives and adverbs standing nearby. Next, we total up the number of learners using adjectives and adverbs in the reviews. If there is number of occurrence ≥ 2, adjective/adverb which is seen as an opinion keyword and will be added to the available list to enrich the collective knowledge in K set.

3.2.3. Infrequent feature mining

As we have mentioned above, infrequent feature could also form nouns or noun phrases. Based on results we observed that opinion keywords stand near features in a sentence, we find the nouns or noun phrases near the opinion keywords in K.

Examining the entire reviews database, with each sentence, if it does not contain any frequent features but containing one or more opinion keywords, we find noun or noun phrase which is the nearest opinion keywords. The noun or noun phrase is then stored in the infrequent feature set.

```
Initializing infrequent feature set V=∅
for i=0, ..., N do
    Extract set P
    /* p_k : noun/noun phrase and p_k ⊂ S_i */
    if S_i ⊃ O_k and S_i ⊄ W then
      p:=noun/noun phrase from nearest O_k
      I ← p;
    end if
  end for
  W= W ∪ V
```

Figure 5. *Infrequent features mining algorithm*

4. EXPERIMENT RESULTS

4.1. Test review data

We extract reviews of six courses from yahoo.com. We extract first 100 reviews for each of them. Afterwards, we eliminate HTML tags and extract reviews in the page. After that, we apply our system to mine features set.

We tagged all the reviews manually. We identified all the features in the reviews. If a sentence in a review shows the opinion of a user, we tag the feature in the sentence.

4.2. Evaluation measures

The evaluate method we used are precision (p) and recall (r). The formulas are as follows [2]:

$$r = \frac{\sum_{i=1}^{n} EC_i}{\sum_{i=1}^{n} C_i} \quad \text{and} \quad p = \frac{\sum_{i=1}^{n} EC_i}{\sum_{i=1}^{n} E_i}$$

$$F - score = \frac{2 \times p \times r}{p + r}$$

In these formulas, n is the total number of reviews of a course. EC_i is the number of correct extracted features from review i, C_i is the number of actual features in review i, E_i is the number of extracted features from review i. This evaluation is based on the result of every review as it is crucial to extract features correctly from every review.

4.3. Results

Table 1. *Recall and precision results*

	Precision	**Recall**	**F-score**
Course 1	72%	77%	74%
Course 2	71%	79%	75%
Course 3	72%	76%	74%
Course 4	70%	87%	78%
Course 5	72%	83%	77%
Course 6	72%	82%	77%
Average	**72%**	**81%**	**76%**

We compare the results of experimental models OFMM and the results of other systems have been published including: systems of Opinion Observer Bing Liu and Minqing Hu [2], the system of Kanayama and Nasukawa [5] and the system of Opine Popescu and et al [3].

Recall of OFMM system achieved quite good (81%). However, the precision is low (72%) when compared with systems rest. Cause of low precision is in the process of mining the infrequent features, the algorithm has added some features to the target feature set incorrectly and the results as the size of this set increases. When using F-score, system OFMM results of 76%, ranked second among the system was announced.

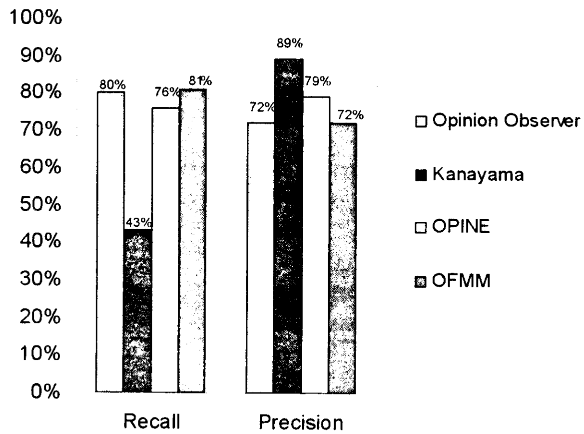

Figure 6. *Results of OFMM and published systems*

5. CONCLUSION

In this research, we based on some results of the published research projects to propose an model (OFMM) for exploiting course features in an e-learning system. This paper has obtained the satisfactory results, especially the recall reached 81%, promises ability to be applied in practice.

However, when increasing the size of the result feature set by mining the infrequent features, the algorithms extract both incorrect features. This affects the accuracy of the algorithm, making the accuracy decreased (only 72%).

In our future work, we will improve techniques of mining infrequent features by adding additional processing steps to remove incorrect features to increase the precision for the results.

REFERENCES

1. Bing Liu & Minquing Hu, "Mining Opionion Features in Customer Reviews", *American Association for Artificial Intelligence*, 2004.
2. B. Liu, M. Hu, and J. Cheng, "Opinion observer: Analyzing and comparing opinions on the web," *in Proceedings of WWW*, 2005.

3. A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP'05)*, 2005.

4. X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining", *in Proceedings of the Conference on Web Search and Web Data Mining (WSDM)*, 2008.

5. H. Kanayama and T. Nasukawa, "Fully Automatic Lexicon Expansion for Domain-Oriented Sentiment Analysis", *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*, 2006.

6. B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis", *Foundations and Trends in Information Retrieval,* 2008.

7. R. Agrawal, R. Srikant, "Fast Algorithm for Mining Association Rules", VLDB'94, 1994.

8. P. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," Proceedings of the Association for Computational Linguistics (ACL), pp. 417–424, 2002.

9. Pang, B., Lee, L. and Vaithyanathan, S. "Thumbs up? Sentiment classification using machine learning techniques", In proceedings of EMNLP'2002.

10. Zhuang, L., Jing, F. and Zhu, X.Y. Movie Review Mining and Summarization. In proceedings of CIKM'06, pp.43-50.

11. Ohno, H., Kusumura, Y., Hijikata, Y., Nishida, S. "Social summarization of text feedback for online auctions and interactive presentation of the summary", In proceedings of IUI'2006, pp.242-249.

12. Rilooff, E. and Wiebe, J. "Learning extraction patterns for subjective expressions", In the proceedings of HLT-EMNLP'2003, pp.25-32.

13. Dave, K., Lawrence, S., and Pennock, D. "Mining thePeanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews", WWW'03, 2003.

14. Dan Song, Hongfei Lin, and Zhihao Yang "Opinion Mining in e-Learning System", International Conference on Network and Parallel Computing, 2007.

15. Zhai, Y., and Liu, B. "Web data extraction based on partial tree alignment", WWW'05, 2005.

16. Wilson, T., Wiebe, J., and Hwa, R. "Just how mad are you? Finding strong and weak opinion clauses", AAAI'04, 2004.