Let's Tango – Finding the Right Couple for Feature-Opinion Association in Sentiment Analysis

Kam Tong Chan and Irwin King

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{ktchan,king}@cse.cuhk.edu.hk

Abstract. One approach in opinion mining is to perform sentiment classification at the sentence level. User's view on a discovered product feature is predicted by the opinion words, e.g. adjectives, appeared in the same sentence. A number of previous works has been proposed and these approaches typically treat the feature and word relations identically. Blindly using sentiments of all opinion words to perform classification would lead to false results. In this paper, we investigate the relationship between features and opinion words using the corpus-based approach. We proposed a Feature-Opinion Association (FOA) algorithm to match these two in sentences to improve sentiment analysis results. We construct a feature-based sentiment lexicon using the proposed algorithm in the sentiment identification process. Extensive experiments based on a commercial product review site show that our method is quite effective in obtaining a more accurate result.

Key words: Opinion mining, sentiment analysis

1 Introduction

One task in opinion mining is to perform feature-based summarization (FBS), which identifies target features mentioned in the reviews and determine their sentiment. Early approaches in FBS use a predefined sentiment lexicon to check against the extracted terms in reviews [2]. The problem of this approach is that sentiments of opinion words are context sensitive. Improvements have been made by generating lexicons with are domain-specific [3] or feature-specific [1].

An accurate and robust Feature-Opinion Association (FOA) method is crucial for both lexicon generation and sentiment classification. That is because more than one feature and opinion word may be mentioned in a sentence. It is not necessary that all opinion words appeared in the sentence are used to describe every feature. An accurate FOA algorithm tells us which feature an opinion word is describing and thus the correct *sense* of that word can be used.

In this paper, we propose to use a function to compute the relevance score between features and opinion words. We then propose an FOA algorithm to match features and opinion words by maximizing the sum of the relevance scores of sentences. The algorithm can be used to generate a sentiment lexicon and be used in sentiment analysis. Experiment results show that our method is useful in improving the sentiment classification accuracy.

2 Proposed Approach

In this section, we give definitions to the Feature-Opinion Association problem and present our proposed solution to the problem.

2.1 Feature-Opinion Association

We define *Feature* and *Opinion words* as follows:

Feature can be a component of the product, e.g. Flash, Lenses, or it can be an attribute of the product, e.g. Weight, Size. Products in the same category share a similar set of common features.

Opinion words in general can be anything that is used to describe a feature. However, due to the difficulties in natural language understanding, it is not easy to perform sentiment analysis on all types of opinion. As with most of the existing works, we limit our scope to handle opinion words that are in the from of adjectives and adverbs.

Here, we define the Feature-Opinion Association Problem (FOA) as follows:

Definition 1. *Given a sentence that contains a non-empty set of features $F = \{f_0, f_1, ..., f_n\}$ and opinion words $W = \{w_0, w_1, ..., w_m\}$, find out a matching M that matches product features with their related opinion words so that the following function is maximized:*

$$foa(F, M) = (\sum_{i=0}^{n} \sum_{j=0}^{|M(fi)|} rel(f_i, M(f_i)(j)). \tag{1}$$

In Eq. (1), $M(f)$ is the set of opinion words matched to feature f and $M(f)(i)$ is the i-th opinion word in the set. The function $rel(f, w)$ returns the relevance score of opinion word w to feature f. The key to this association problem is to define a good $rel(f, w)$ function.

2.2 *Rel* Function

Here is a list of possible rel functions for the FOA problem:

- **Nearest Opinion Word (DIST)**
  The simplest solution to the FOA problem is to associate opinion words that are nearest to features. In this case, $rel(f, w)$ is defined as the inverse of the distance between the opinion word w and feature f.

- **Co-Occurrence Frequency (COF)**
  Another approach to the FOA problem is to define $rel(f, w)$ as the Co-Occurrence Frequency (COF) between feature f and opinion word w. The intuition is that the higher the COF score, the more likely that the opinion word is related to the feature.
- **Co-Occurrence Ratio (COR)**
  COF has the problem that common terms (i.e., "good", "great", "excellent") dominate the top of the rank. To get a more feature-specific ranking, we should also take into accounts the corpus frequency (CF) (i.e. the number of times a term appears in the corpus) of opinion words. In this case, the rel function becomes:

$$ref(f, w) = \begin{cases} \frac{COF(f,w)}{CF(w)} & \text{if COF(f,w)} > \text{threshold} \\ 0 & \text{otherwise} \end{cases}$$

- **Likelihood-Ratio Test (LHR)**
  Using the method proposed by Yi [4], assuming the association between feature and opinion word is a Bernoulli event and it follows a binomial distribution, we can compute the likelihood ratio as follows:

$$-2 \log \lambda = \begin{cases} -2 * lr & \text{if } r_2 < r_1 \\ 0 & \text{if } r_1 < r_2 \end{cases}$$

  where

$$\begin{aligned} lr =& (C_{11} + C_{21}) \log(r) + (C_{12} + C_{22}) \log(1 - lr) \\ & - C_{11} \log(r_1) - C_{12} \log(1 - r_1) - C_{21} \log(r_2) \\ & - C_{22} \log(1 - r_2) \end{aligned}$$

$$\begin{aligned} r_1 =& \frac{C_{11}}{C_{11} + C_{12}} \\ r_2 =& \frac{C_{21}}{C_{21} + C_{22}} \\ r =& \frac{C_{11} + C_{21}}{C_{11} + C_{12} + C_{21} + C_{22}} \end{aligned}$$

$$\begin{aligned} C_{11} =& COF(f, w) \\ C_{12} =& CF(w) - COF(f, w) \\ C_{21} =& CF(f) - COF(f, w) \\ C_{22} =& SentenceCount - C_{11} - C_{12} - C_{21} \end{aligned}$$

  The computed log likelihood ratio $-2 \log \lambda$ is used as rel score in this case. A larger value represents a higher chance that opinion word w should be associated with feature f.

- **Combined Method**
  In here, we combine DIST, a distance based measurement, with statistical measurements COR and LHR. This is to combine the best of both types of measurements. They are defined as follows:
  COR+DIST (C+D):

$$rel(f, w) = \frac{COF(f, w)}{CF(w) * dist(f, w)}. \tag{2}$$

  LHR+DIST (L+D):

$$rel(f, w) = \frac{-2 \log \lambda}{dist(f, w)}. \tag{3}$$

2.3 Feature-Opinion Association Algorithm

We define the Feature-Opinion Association Algorithm (FOAA) to associate opinion words to features subject to the *rel* function in Algorithm 1:

Algorithm 1 Feature-Opinion Association Algorithm

```
$F$ ← features in $S$
$W$ ← opinion words in $S$
for each $w$ in opinion word list $W$ do
    $score$ ← highest $rel(f, w)$ for all $f \in F$
    if $score \geq threshold$ then
        if the same word is already assign to $f$ then
            Try another $f$ with the next highest $foa$ score
        else
            associate $w$ to $f$
        end if
    end if
end for
```

In general, each opinion word will be associated to the feature with the highest *rel* score except in sentences like: *"Good lenses, good pictures!"*. Depending on the *rel* function used, it is possible that the scores for the same pair of feature and opinion word are the same for different appearances of the same opinion word. The algorithm will check if the same opinion word is already associated to a feature, and if so, it tries the next feature until no more features exist. A threshold value is used to prune opinion words that have low *rel* scores to all features appeared in the sentence.

2.4 Sentiment Lexicon Expansion

With the Feature-Opinion Association algorithm, steps for sentiment lexicon expansion become straightforward. Two sets of opinion words (positive and negative) are defined initially as seeds. FOA is applied on each sentence appeared in the training corpus. Same opinion words associated to different feature are treated as different words. Using the linguistic rules proposed by Ding [1], we can compute the orientation score for each word as follows:

$$orientation(w) = \frac{c_{+ve} - c_{-ve}}{CF(w)}, \tag{4}$$

where c_{+ve} and c_{-ve} is the number of times word w is in conjunction with the known positive and negative set respectively. The higher the score, the more likely that the word should be in the positive set and vice vera. Words with absolute score smaller than a threshold T should not be treated as either polarity. In each iteration, Words with the highest and lowest score are added to the two sets respectively.

3 Evaluation

In order to verify our ideas, we collected two data sets from the internet. These two data sets are used to conduct our experiments. NLTK[1] is used to perform natural language processing tasks such as sentence splitting and part-of-speech (POS) tagging. Feature list is constructed based on the glossary page of the Digital Photography Review[2]. Synonyms are grouped together and treated as the same feature.

There are two data sets in our experiment:

Corpus Data Set: User reviews of all cameras of popular brands are crawled from Digital Photography Review. This data set contains 400+ different camera models, 17,000+ user reviews and 250,000+ sentences. This data set is used as a statistical database for computing the relevance scores discussed in Section 2.

Test Data Set: We used another data set [2, 1] for testing. The reviews of 4 different camera models are used. The data set contains over 100 reviews and 1500 sentences. The reviews are re-tagged so that each sentence is attached with the mentioned features and their associated opinion words.

3.1 Feature-Opinion Association Accuracy

For each tagged sentence in the test data set, we use the FOA algorithm to match appeared features and opinion words. Therefore, for each feature, there will be a list of opinion words tagged by human and the FOA algorithm respectively. We have defined the following values to count the number of opinion-feature pair that are tagged by either Human or the FOA algorithm:

- C_{AA}: Both Human and FOA tagged
- C_{AN}: Only Human tagged
- C_{NA}: Only FOA tagged

[1] http://nltk.org/

[2] http://www.dpreview.com/

Table 1. FOA Accuracies

	DIST	**COF**	**COR**	**LHR**	**C+D**	**L+D**
th	0.20	1.00	0.01	0.00	0.02	9.00
p	0.58	0.45	0.46	0.49	0.59	0.66
r	0.64	0.75	0.71	**0.79**	0.65	0.64
f	0.61	0.57	0.56	0.60	**0.62**	**0.65**

- C_{NN}: Neither Human nor FOA tagged

Then, the association accuracy can be computed by the traditional precision (p), recall (r) and F-score (f):

$$p = \frac{C_{AA}}{C_{AA} + C_{NA}}, r = \frac{C_{AA}}{C_{AA} + C_{AN}}, f = \frac{2 * p}{p + r}.$$

We evaluate the accuracy of FOA using each *rel* function with a range of possible threshold *th*. For COR, DIST and COR+DIST, the range is from 0 to 1 with step size 0.01. Range for other functions is from 0 - 100 with step size 1. The results of each data set with the best F-Score is presented in Table 1. We only report the average results of the 4 data set due to space limitations.

From the results we can see that non-combined *rel* functions are capable of achieving good recalls (around $65-79\%$). However, their precisions are generally quite low. Among all the non-combined *rel* functions, LHR and DIST performed the best, reaching an average F-Score of 60%. DIST achieved the highest average precision, but it is mainly due to the outliner camera 1 (not shown here). LHR has the highest average recall (nearly 80%).

The combined *rel* functions, in general, perform better in terms of the F-Score measurements with LHR+DIST slightly outperforming COR+DIST. The results here indicate the importance of using both types of measurement in the feature-opinion association process. Missing either information will lead to worse association results.

3.2 Sentiment Classification Accuracy

We used LHR and LHR+DIST to generate two sentiment lexicons by the algorithm discussed in section 2.4. The reason for choosing them is that they performed the best in terms of average Precision, Recall and F-Score in the FOA process. The FOA thresholds (th) 0, and 9 are selected for LHR and LHR+DIST respectively. They are thresholds that each methods achieve the best average F-Score in the FOA process. Opinion words {"excellent", "good"} and {"poor", "bad"} are used as the initial seed words for the positive and negative sentiment respectively. In our experiment, the orientation threshold T for lexicon generation is set to 0.2.

We conducted two set of experiments with identical settings except that one includes FOA while the other does not. The sentiment classification process is

Table 2. Sentiment Classification Accuracy

		LHR	**L+D**		**LHR**	**L+D**		**LHR**	**L+D**		**LHR**	**L+D**		**LHR**	**L+D**
All	**Camera 1**	0.745	0.668	**Camera 2**	0.703	0.695	**Camera 3**	0.799	0.672	**Camera 4**	0.663	0.626	**Average**	0.727	0.665
Human		0.761	0.719		0.740	0.740		0.797	0.709		0.656	0.660		0.739	0.707
Human*		0.772	0.710		0.749	0.748		0.845	0.721		0.653	0.667		0.755	0.711
FOA		0.692	0.662		0.685	0.647		0.785	0.656		0.629	0.585		0.698	0.638
FOA*		**0.755**	0.695		0.729	**0.736**		**0.828**	0.716		**0.694**	0.636		**0.752**	0.696

as follows: For each tagged sentences, all the appeared features and opinion words are extracted. The algorithm computes the sentiment score for each feature mentioned in the sentence solely based on the associated opinion words. Features that are not associated to any opinion words will have their sentiments inferred using two different methods. The first method (*Human* and *FOA*) uses the majority sentiment of other features that appeared at the same sentence. The second method (*Human** and *FOA**) falls back to use all opinion words of the same sentence to infer sentiments of these features. This is the same as the case where FOA is not used. Under both cases, we use the opinion aggregation function [1] for sentiment scoring:

$$score(f) = \sum_{w_i : w_i \in M(f)} \frac{w_i.SO}{dist(w_i, f)}. \qquad (5)$$

$M(f)$ is the set of opinion words associated to feature f. For the case where FOA is not used (*All*), $M(f)$ will be replaced by the set of all opinion words within the same sentence as feature f. (*All*) represents the opinion aggregation method proposed in [1]. This give us a direct comparison of FOA versus the state of the art method in sentiment analysis. $dist(w_i, f)$ is the distance between opinion word w_i and feature f. $w_i.SO$ is the sentiment score of opinion word w_i. Positive words have a score of +1 and negative words have a score of -1. Sentiments of opinion words are retrieved from sentiment lexicons generated in the above steps. The predicted sentiment orientations are compared against human tags to calculate accuracies. Table 2 summarizes the results of our sentiment classification experiments.

Classification results using human tags We first compare the sentiment classification accuracies of all opinion words (*All*) and human association (*Human* and *Human**). We can see that the overall accuracies increase if we limit the sentiment classifier to use only opinion words that are tagged by human. These agree with our intuitions that blindly using all opinion words actually produces false results.

An interesting observation is that *Human** performs better than *Human*. The reason is that a sentence usually mentions only 1 or 2 features. When there are no associated opinion words, we either cannot find another feature, or the remaining features are not enough to help inferring its sentiment correctly. Falling back to use all opinion words actually helps in this case.

Classification results using FOA algorithm Solely using opinion words that are associated by the FOA algorithm for sentiment analysis (*FOA* in Table 6) actually produces poorer results. This is reasonable, given that the accuracies of using human FOA alone (*Human*) are just slightly better than the case where all opinion words are used (*All*). However, if we use the second method to deal with the case where no opinion words are associated to a feature (*FOA**), the overall accuracy improves and it consistently outperform *All* in our experimental data sets. In fact, we can interpret FOA* as a pruning heuristic that tries to tighten the set of opinion words used for sentiment classification whenever possible. The results show that this heuristic is effective in improving the sentiment classification accuracies.

Finally, we observe that using LHR to perform FOA actually performed better. A possible reason is that LHR is good at achieving high recalls. It extracts most of the feature-related opinion words in the lexicon building process. Although its precision is not high, the incorrectly associated opinion words are unlikely to be inferred to carry a sentiment because of the threshold limitations.

4 Conclusion

We have presented a Feature-Opinion Association (FOA) algorithm to improve the sentiment analysis results. The algorithm maximizes the sum of the relevance scores between features and opinion words. We proposed 6 relevance measures that make use of the structural information of sentences as well as the statistical information collected from a commercial review web site. The proposed FOA algorithm can be used in both the lexicon generation and sentiment classification process. The evaluation results show that it is effective in improving the sentiment analysis accuracy over the traditional methods where classification is done using all opinion words.

5 Acknowledgements

This work is fully supported by two grants (Project No. CUHK 2050379 and Project No. CUHK 6902498) and is affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

References

1. Ding, X., Liu, B.: A holistic lexicon-based approach to opinion mining. WSDM '08, 231–240 (2008)
2. Hu, M., Liu, B.: Mining and summarizing customer reviews. KDD '04, 168–177 (2004)
3. Kanayama, H., Nasukawa, T.: Fully Automatic Lexicon Expansion for Domain-oriented Sentiment Analysis. EMNLP '06, 355–363 (2006)
4. Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. ICDM '03, 427 (2003)