COMMIT-P1WP3: A Co-occurrence Based Approach to Aspect-Level Sentiment Analysis

Kim Schouten[1,2] **Flavius Frasincar**[1] **Franciska de Jong**[2]
schouten@ese.eur.nl frasincar@ese.eur.nl fdejong@ese.eur.nl
[1]Econometric Institute, Erasmus University Rotterdam, The Netherlands
[2]Erasmus Studio, Erasmus University Rotterdam, The Netherlands

Abstract

In this paper, the crucial ingredients for our submission to SemEval-2014 Task 4 "Aspect Level Sentiment Analysis" are discussed. We present a simple aspect detection algorithm, a co-occurrence based method for category detection and a dictionary based sentiment classification algorithm. The dictionary for the latter is based on co-occurrences as well. The failure analysis and related work section focus mainly on the category detection method as it is most distinctive for our work.

1 Introduction

In recent years, sentiment analysis has taken flight and is now actively used, on the Web and beyond (Liu, 2012). To provide users of sentiment tools with more detailed and useful information, a number of innovations have been introduced, and among others a switch from document-level sentiment analysis towards fine-grained, aspect-level sentiment analysis can be seen (Feldman, 2013). In line with the many challenges associated with this, SemEval-2014 Task 4 "Aspect Level Sentiment Analysis" (Pontiki et al., 2014) is split into four sub-tasks: Aspect Detection, Aspect Sentiment Classification, Category Detection, and Category Sentiment Classification.

The main focus of this paper is on the category detection algorithm we developed, but a method for aspect detection and a sentiment classification algorithm (both for aspects and categories) are also included. The aspect detection algorithm will be presented first, followed by the category detection algorithm and the sentiment classification method. Next, the benchmark results for all algorithms are presented, plus a discussion and failure analysis of the category detection method. Finally, conclusions are drawn and some suggestions for future work are presented.

2 Related Work

Because the focus of this paper lies on the category detection method, only for that method a short snippet of related work is given. That algorithm, being an adapted version of Schouten and Frasincar (2014), is inspired by the work of Zhang and Zhu (2013) and Hai et al (2011). In these works, a co-occurrence matrix is created between words in the sentence and aspects in order to find implicit aspects (i.e., aspects that are not literally mentioned, as opposed to the explicit aspects used in this task).

While implicit aspects are similar to aspect categories to some extent, these methods do not work when a fixed, limited set of possible aspect categories is used that is, most importantly, not a subset of the set of aspects. The above methods could never, for instance, identify the 'anecdotes/miscellaneous' category, as this word never appears as an aspect in the data set. This is the main reason why we have chosen to count the co-occurrences between words and the annotated aspect categories.

3 Aspect Detection Method

In the work reported here, the aspect detection method plays the role of a baseline method rather than a full-fledged algorithm. In its most basic form, it annotates all noun phrases as aspects. However, by using the training set to count how often each word appears within an aspect, a simple probability can be computed representing the chance that this word is an aspect word or not. This probability is used to filter the set of noun phrases, such that only noun phrases remain that

have at least one word for which the aspect probability ≥ 0.05 and for those noun phrases, all leading words in the noun phrase with a probability below 0.05 are removed. This will remove words like determiners from the initial noun phrase, as those are not included in the aspect term. Because this filtering is strict, the result is a typical high precision, low recall algorithm for aspect detection.

4 Category Detection Method

To find the aspect categories, the co-occurrence based algorithm from Schouten and Frasincar (2014) is used and improved upon. The central construct in this algorithm is a co-occurrence matrix that captures the frequency of the co-occurrences between words (i.e., the lemmas of the words) in the sentence and the annotated aspect category. This gives a mapping from words to aspect categories. When processing an unlabelled sentence, a score is computed for each aspect category as shown in Eq. 1.

$$score_{a_i} = \frac{1}{v} \sum_{j=1}^{v} \frac{c_{i,j}}{o_j}, \tag{1}$$

where v is the number of words in the sentence, a_i is the ith aspect category in the list of possible aspect categories for which the $score$ is computed, j represents the jth word in the sentence, $c_{i,j}$ is the co-occurrence frequency of aspect category i and lemma j in the data set, and o_j is the frequency of lemma j in the data set.

Whereas in Schouten and Frasincar (2014), the highest scoring category was chosen on the condition that its score exceeded a threshold, our method is now able to choose more than one aspect category per sentence. This is done by training a separate threshold for each of the five aspect categories using all training data. When the score for some aspect category is higher than its associated threshold (i.e., $score_{a_i} > threshold_{a_i}$), the sentence is annotated as having that aspect category.

Since we assume the five threshold values to be independent of each other, a simple linear search is performed separately for all five of them to find the optimal threshold value by optimizing F_1 (cf. Sec. 6). As a default option, the fifth category ('anecdotes/miscellaneous') is associated to any sentence for which none of the five categories exceeded their threshold. The trained threshold values for the five categories are:

ambience	price	food	service	misc
0.042	0.024	0.211	0.071	0.143

The pseudocode for the creation of the co-occurrence matrix can be found in Algorithm 1, and Algorithm 2 describes the process of annotating a sentence with aspect categories.

Algorithm 1 Aspect category detection training.

```
Initialize set of word lemmas with frequencies O
Initialize set of aspect categories A
Initialize co-occurrence matrix C
for sentence s ∈ training data do
    for word w ∈ s do
        O(w) = O(w) + 1
    end for
    for aspect category a ∈ s do
        add a to A
        for word w ∈ s do
            C(w, a) = C(w, a) + 1
        end for
    end for
end for
for aspect category a in A do
    threshold_a=0
    bestF_1 = 0
    for t = 0 to 1 step 0.001 do
        Execute Algorithm 2 on training data
        Compute F_1
        if F_1 > bestF_1 then threshold_a = t
        end if
    end for
end for
```

5 Sentiment Classification Method

For sentiment classification, a method is developed that first creates a sentiment lexicon based on the aspect sentiment annotation. That lexicon is then consequently used to determine the sentiment of both aspects and categories that have no sentiment annotation. The intuition behind this method is that a lexicon should cover domain-specific words and expressions in order to be effective. To avoid creating such a sentiment lexicon manually, the aspect sentiment annotations are leveraged to create one automatically. The idea is that words that often appear close to positive or

Algorithm 2 Aspect category detection execution.

```
for sentence s ∈ test data do
    for aspect category a ∈ A do
        score = 0
        for word w ∈ s do
            if O(w) > 0 then
                score = score + C(w, a)/O(w)
            end if
        end for
        score = score/ length(s)
        if score > threshold_a then
            Assign aspect category a to s
        end if
    end for
    if s has no assigned aspect categories then
        Assign 'anecdotes/miscellaneous' to s
    end if
end for
```

negative aspects are likely to have the same polarity. Since sentiment is also carried by expressions, rather than single words only, the constructed sentiment lexicon has entries for encountered unigrams, bigrams, and trigrams. In each sentence, the distance between each n-gram and each aspect is computed and the sentiment of the aspect, discounted by the computed distance, is added to the sentiment value of the n-gram, as shown in Eq. 2.

$$sentiment_g = \frac{1}{freq_g} \cdot \sum_{s \in S_g} p \cdot t_{order(g)} \cdot \sum_{a \in A_s} \frac{polarity_a}{(distance_{g,a})^m}, \quad (2)$$

where g is the n-gram (i.e., word unigram, bigram, or trigram), $freq_g$ is the frequency of n-gram g in the data set, s is a sentence in S_g, which is the set of sentences that contain n-gram g, p is a parameter to correct for the overall positivity of the data set, t is a parameter that corrects for the relative influence of the type of n-gram (i.e., different values are used for t_1, t_2, and t_3), a is an aspect in A_s, which is the set of aspects in sentence s, $polarity_a$ is 1 when aspect a is positive and -1 when a is negative, and m is a parameter that determines how strong the discounting by the distance should be. The distance computed is the shortest word distance between the aspect and the n-gram (i.e., both an n-gram and an aspect can consist of multiple words, in which case the closest two are used to compute the distance). Note that essentially, dictionary creation is based on how often an n-gram co-occurs with positive or negative aspects. In our submitted run on the restaurant data, we set $t_{order(g)}$ to 1, 5, and 4 for unigrams, bigrams, and trigrams, respectively, and $p = 2$ and for the laptop data we set $t_{order(g)}$ to 1, 0, and 3 for the n-grams and $p = 1$. In both cases, m was kept at 1. These values were determined by manual experimentation.

To compute the sentiment of an aspect, the sentiment value of each n-gram is divided by the distance between that n-gram and the aspect, computed in a similar fashion as in the above formula) and summed up, as shown in Eq. 3.

$$sentiment_{a,s_a} = \sum_{g \in s_a} \frac{sentiment_g}{(\min distance_{g,a})^m}, \quad (3)$$

where, in addition to the definitions in the previous equation, g is an n-gram in s_a, which is the sentence in which aspect a occurs. Note that for each occurrence of a term, its sentiment value is added to the total score. If the result is above zero, the class will be 'positive', and if the result is below zero, the class will be 'negative'. In the rare event of the sentiment score being exactly zero, the 'neutral' class is assigned.

For category sentiment classification, the formula of Eq. 3 remains the same, except that the distance term $\min distance^m_{g,a}$ is set to 1, since aspect categories pertain to the whole sentence instead of having specific offsets.

6 Evaluation

All three algorithms presented above were evaluated through a submission in the SemEval-2014 Task 4 "Aspect Level Sentiment Analysis". Two data sets have been used, one consisting of sentences from restaurant reviews, the other consisting of sentences from laptop reviews. Both sets have been annotated with aspects and aspect sentiment, but only the restaurant set is also annotated with aspect categories and their associated sentiment class. Both data sets are split into a training set of roughly 3000 sentences and a test set of 800 sentences.

All sentences in the data set have been preprocessed by a tokenizer, a Part-of-Speech tagger, and a lemmatizer. These tasks were performed by

Table 1: Official results for both algorithms.

aspect detection (subtask 1)			
	precision	recall	F_1
laptop	0.836	0.148	0.252
restaurant	0.909	0.388	0.544
category detection (subtask 3)			
	precision	recall	F_1
restaurant	0.633	0.558	0.593
aspect sentiment classification (subtask 2)			
laptop	accuracy	0.570	
restaurant	accuracy	0.660	
category sentiment classification (subtask 4)			
restaurant	accuracy	0.677	

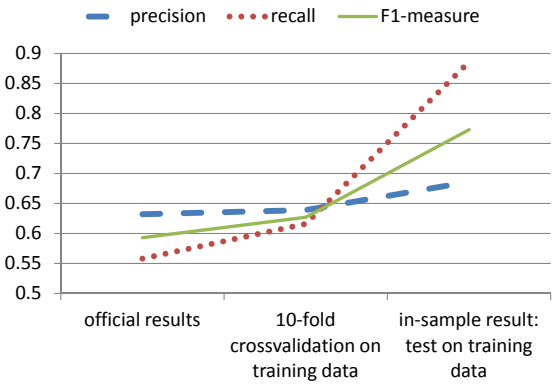

Figure 1: Performance measure of category detection on different parts of data.

the Stanford CoreNLP framework[1]. Furthermore, the OpenNLP[2] chunker was used to provide basic phrase chunking in order to retrieve noun phrases for instance.

The official scores, as computed by the task organizers are shown in Table 1. Note that the sentiment classification algorithm is used for subtasks 2 and 4, so two scores are reported, and that subtasks 3 and 4 can only be performed with the restaurant data set.

As the performance of the category detection method was lower than anticipated, a failure analysis has been performed. This led to the observation that overfitting is one of major factors in explaining the lower performance . This is shown in Figure 1, in which one can easily notice the great difference in in-sample performance, and the performance on unseen data. Notice that by using 10-fold cross-validation, better results are achieved than on the official test set. This indicates that there are factors other than overfitting that influence the performance.

Interestingly, especially recall is influenced by the overfitting problem: precision is almost the same for the 10-fold cross-validation and even with the in-sample performance it increases only a little bit. To gain more insight into the difference in recall, a graph showing the relative contribution to false negatives of the five categories is shown in Figure 2. For completeness, the same graph but for false positives is also shown, together with the frequency distribution of the categories in both training and test set.

Immediately visible is the effect of defaulting to the 'anecdotes/miscellaneous' when no category is assigned to that sentence: many false positives are generated by this rule, but there are almost no false negatives for this category. Note that without this default, F_1-measure would drop by roughly 3 percentage points.

Also notable is the difference between the in-sample bar and the official results bar: two categories, namely 'anecdotes/miscellaneous' and 'food' show large differences in contribution to false positives and false negatives. The algorithm finds fewer 'food' categories in the test set, than in the training set, while for 'anecdotes/miscellaneous', the reverse is the case. This can at least be partly explained by the change in data statistics: in the training set, 33% of the annotated categories are 'food' and 30% are 'anecdotes/miscellaneous', whereas in the test set, these numbers are 40% and 22%, respectively. With much more sentences having the 'food' category, false positives will be lower but false negatives will be higher. For 'anecdotes/miscellaneous', the reverse is true: with less sentences in the test set having this category, false positives will by higher, but false negatives will be lower, a change reinforced by 'anecdotes/miscellaneous' being the default.

Two factors remain that might have negatively impacted the performance of the algorithm. The first is that in the restaurant set, many words appear only once (e.g., dishes, ingredients), and when words do not appear in the training set, no co-occurrence with any category can be recorded. This primarily affects recall. The second is that the category thresholds, while working well on the training set, do not seem to generalize well to the

[1] http://nlp.stanford.edu/software/corenlp.shtml
[2] https://opennlp.apache.org/

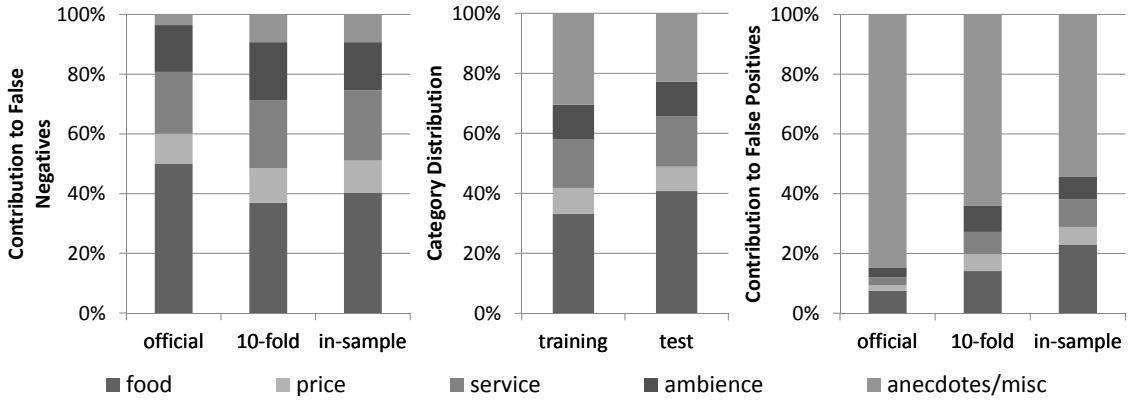

Figure 2: The frequency distribution of each category and its relative contribution to the total number of false negatives (left graph) and false positives (right graph). The middle graph shows the change in the distribution of categories in the training and test set.

test set. Testing the algorithm with one threshold for all five categories, while showing a sharply decreased in-sample performance, yields an out-of-sample F_1-measure that is only slightly lower than F_1-measure with different thresholds.

7 Conclusion

In this paper the main ingredients for our submission to SemEval-2014 Task 4 "Aspect Level Sentiment Analysis" are discussed: a simple aspect detection method, a co-occurrence based method for category detection, and a dictionary based sentiment classification algorithm. Since the category detection algorithm did not perform as expected, a failure analysis has been performed, while for the others this was less necessary as they performed roughly as expected.

The failure analysis provides several starting points for future research. First, it would be interesting to determine the exact nature of the dependency between category performance and category frequency, as discussed above, and to remove this dependency, since it is not guaranteed in real-life scenarios that the frequency distribution of the training set is the same as the set of instances an algorithm will encounter when in use. Furthermore, training five separate category threshold, while good for performance in general, also aggravates the problem of overfitting. Hence, improving the generalization of the algorithm, and the thresholds in particular, is important. Last, a method to deal with very low frequency words could prove useful as well.

Acknowledgment

The authors are partially supported by the Dutch national program COMMIT.

References

Ronen Feldman. 2013. Techniques and Applications for Sentiment Analysis. *Communications of the ACM*, 56(4):82–89.

Zhen Hai, Kuiyu Chang, and J. Kim. 2011. Implicit Feature Identification via Co-occurrence Association Rule Mining. In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text processing (CICLing 2011)*, volume 6608, pages 393–404. Springer.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*, volume 16 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval 2014)*.

Kim Schouten and Flavius Frasincar. 2014. Finding Implicit Features in Consumer Reviews for Sentiment Analysis. In *Proceedings of the 14th International Conference on Web Engineering (ICWE 2014)*, pages 130–144. Springer.

Yu Zhang and Weixiang Zhu. 2013. Extracting Implicit Features in Online Customer Reviews for Opinion Mining. In *Proceedings of the 22nd International Conference on World Wide Web Companion (WWW 2013 Companion)*, pages 103–104. International World Wide Web Conferences Steering Committee.