Finding Implicit Features in Consumer Reviews for Sentiment Analysis

Kim Schouten and Flavius Frasincar

Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, the Netherlands
{schouten, frasincar}@ese.eur.nl

Abstract. With the explosion of e-commerce shopping, customer reviews on the Web have become essential in the decision making process for consumers. Much of the research in this field focuses on explicit feature extraction and sentiment extraction. However, implicit feature extraction is a relatively new research field. Whereas previous works focused on finding the correct implicit feature in a sentence, given the fact that one is known to be present, this research aims at finding the right implicit feature without this pre-knowledge. Potential implicit features are assigned a score based on their co-occurrence frequencies with the words of a sentence, with the highest-scoring one being assigned to that sentence. To distinguish between sentences that have an implicit feature and the ones that do not, a threshold parameter is introduced, filtering out potential features whose score is too low. Using restaurant reviews and product reviews, the threshold-based approach improves the F_1-measure by 3.6 and 8.7 percentage points, respectively.

1 Introduction

With the explosion of online shopping at e-commerce companies like Amazon (US), Bol (NL), Alibaba (CN), etc., the use of consumer product reviews has become instrumental in the decision making process of consumers. In fact, potential consumers trust reviews from other consumers more than information on the vendor's website [1]. As a result, the number of reviews for a single product can be quite high, especially for a popular product. When a consumer is interested in the overall sentiment of a product, (s)he must first read through many of the reviews to come to a conclusion. Since reading through these reviews is a tedious process, this may hinder decision making. Therefore an efficient way of displaying the overall sentiment of a product based on costumer reviews is desirable.

Much of the current research in the analysis of product reviews is concerned with classifying the overall sentiment for a certain product. To better describe the overall sentiment of a product, it is useful to look at the sentiment per product aspect, from now on referred to as a feature. Sentiment classification per feature can be difficult as a customer review does not have a standard structure and may include spelling errors and synonyms for product features. Although a consumer might explicitly mention a feature for a product, many of the important features are mentioned implicitly as well. For example:

"The battery of this phone is quite good."
"The phone lasts all day."

In the first sentence, the battery is explicitly mentioned and the second one refers to the battery lasting all day. Notice that while in the second sentence the battery is not explicitly mentioned, we can infer that the comment is about the battery. This inference is based on the other words in the sentence that direct the reader towards the actual feature being described. This mapping from words in the sentence to the implied feature must be shared between writer and reader of a text in order for the reader to understand what the writer meant to imply. Because of this, it is usually a small group of well-known, coarse-grained features that is used implicitly. Examples include generic features like price, size, weight, etc., or very important product-specific features like the already mentioned battery, sound quality, ease of use, etc. Since it is this class of features that is often implied, it is important to include them in any sentiment analysis application, as they represent key features for consumers.

This research presents a method to both determine whether an implicit feature is present in a sentence, and if so, which one it is. After describing some of the related work that inspired this research, the method will be presented. Then, the two data sets that are used in the experiments are discussed, followed by the evaluation of the proposed method. This will lead to the conclusions and suggestions for future work in the last section.

2 Related work

While many methods have been proposed to find features for the task of aspect-level sentiment analysis, most of them focus on explicit features only. This is logical, given that the vast majority of the features in consumer reviews is mentioned explicitly. However, as discussed in the previous section, it is often the important features that are mentioned implicitly. Alas, only few works focus on this task. One of the first to address the problem of detecting implicit features is [8]. An interesting solution is presented in the form of semantic association analysis based on Pointwise Mutual Information. However, since no quantitative results are given, it is impossible to know how well this method performs.

In [5], a method based on co-occurrence Association Rule Mining is proposed. It is making use of the co-occurrence counts between opinion words and explicit features. The latter can be extracted from labeled data, or can be provided by an existing method that finds explicit features. Association rule mining is used to create a mapping from the opinion words to possible features. The opinion word then functions as the antecedent and the feature as the consequent in the rules that are found. When an opinion word is encountered without a linked feature, the list of rules is checked to see which feature is most likely implied by that opinion word. On a custom set of Chinese mobile phone reviews, this method is reported to yield an F_1-measure of 74%.

Similar to [5], the same idea of association rule mining is used in [9]. With association rule mining being used to find a set of basic rules, three possible ways of extending the set of rules are investigated: adding substring rules, adding dependency rules, and adding constrained topic model rules. Especially the latter turned out to be a successful way of improving the results. By constraining the topic model (e.g., Latent Dirichlet Allocation [2] in this case), to include one of the feature words and build the topic around that word, meaningful clusters are generated. Thus, a different way of finding co-occurrences between features and other words in the text is used, and it is reported that this complements the association rule mining method. The best reported result is an F_1-measure of 75.51% on a Chinese data set of mobile phone reviews.

Instead of using annotated explicit features, [10] uses the idea of double propagation [7] to find a set of explicit words and a set of opinion words. An advantage is that the found explicit features are already linked to appropriate opinion words. Then a co-occurrence matrix is created, not between only opinion words and explicit features, but between the words in the sentences and the found explicit features. In this way, the right implicit feature is chosen, not based on just the opinion words in the sentence, but based on all words in the sentence. The opinion words in the sentence are used to constrain the number of possible features from which the right one must be chosen: only features that have co-occurred with the encountered opinion word before, are eligible to be chosen.

In the previously introduced method, for each eligible explicit feature, a score is computed that represents the average conditional probability of a feature being implied, given the set of words in the sentence. The feature with the highest score is chosen as the implicit feature for this sentence. This method is reported to yield an F_1-measure of 0.80 and 0.79 on a Chinese corpus of mobile phone reviews, and a Chinese collection of clothes reviews, respectively. Like [9], it uses all words to find implicit features instead of only opinion words as in [5], and, apart from a small seed set of opinion words, it operates completely unsupervised.

However, there are several drawbacks that are apparent, both in [5], [9], and in [10]. The first problem is that only features that have been found as explicit features somewhere in the corpus can be chosen as implicit features. This assumes that the same features are present in reviews, both explicitly and implicitly. However, as we have discussed before, well-known or important features are implied more often than features that are less important or less described. Furthermore, by counting the co-occurrence frequencies between a feature that is mentioned explicitly and the words in the sentence, it is assumed that when the feature is used implicitly, the same sentential context is present. We argue, however, that this is not necessarily the case. For example, when saying that 'this phone is too expensive', the word 'expensive' prevents the word 'price' from being used. Either one uses the word 'expensive', or one uses the word 'price'. Because of that, there is no real co-occurrence between 'expensive' and 'price', even though the first definitely points to the latter as its implicit feature.

3 Method

In this section the issues discussed in the previous section are addressed and an algorithm is presented that improves upon previous work in the given, more realistic, scenario. This scenario entails the following:

- Sentences can have both explicit and implicit features;
- Sentences can have zero or more implicit features;
- Implicit features do not have to appear explicitly as well;
- The sentential context of explicit features does not have to be the same as the sentential context for implicit features.

The algorithm first scans the training data and constructs a list F of all unique implicit features, a list O of all unique lemmas (i.e., the syntactic root form of a word) and their frequencies, and a matrix C to store all co-occurrences between annotated implicit features and the words in a sentence. Hence, matrix C has dimensions $|F|$ x $|O|$.

When F, O, and C have been constructed, processing the test data goes as follows. For each potential implicit feature f_i, a score is computed that is the sum of the co-occurrence of each word in the sentence divided by the frequency of that word:

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

where v is the number of words, f_i is the ith feature in F for which the *score* is computed, j represents the jth word in the sentence, $c_{i,j}$ is the co-occurrence frequency of feature i and lemma j in C, and o_j is the frequency of lemma o in O. Subsequently, for each sentence the highest scoring feature is chosen.

However, since there are many sentences without any implicit feature, a threshold is added, such that the highest scoring feature must exceed the threshold in order to be chosen. If the computed score does not exceed the threshold, the considered implicit feature is not assigned to that sentence. The pseudocode for the whole process is shown in Alg. 1, where the training process is shown (i.e., constructing co-occurrence matrix C and lists O and F), and in Alg. 2, where the processing of new sentences using the trained algorithm is shown.

The optimal threshold is computed based on the training data only, and consists of a simple linear search. A range of values is manually defined, all of them which are then tested consequently. The values ranged from 0 to 1, with a step size of 0.001. The best performing threshold is then used when evaluating on the test data. Since there is only one parameter to train and the range of possible values is rather limited, more advanced machine learning techniques were not deemed necessary to arrive at a good threshold value.

A limitation of this method is the fact that it will choose at most one implicit feature for each sentence. Both of our data sets, as can be seen in the next section, contain sentences that have more than one implicit feature. In these cases, chances are higher that the chosen implicit feature is in the golden standard, but

Algorithm 1 Training the algorithm with annotated data.

Initialize list of unique word lemmas with frequencies O
Initialize list of unique implicit features F
Initialize co-occurrence matrix C
for sentence $s \in$ training data **do**
    for word $w \in s$ **do**
        if $\neg(w \in O)$ **then**
            add w to O
        end if
        $O(w) = O(w) + 1$
    end for
    for implicit feature $f \in s$ **do**
        if $\neg(f \in F)$ **then**
            add f to F
        end if
        for word $w \in s$ **do**
            if $\neg((w, f) \in C)$ **then**
                add (w, f) to C
            end if
            $C(w, f) = C(w, f) + 1$
        end for
    end for
    Determine optimal threshold.
end for

Algorithm 2 Executing the algorithm to process new sentences.

for sentence $s \in$ test data **do**
    $currentBestFeature = empty$
    $scoreOfCurrentBestFeature = 0$
    for feature $f \in F$ **do**
        $score = 0$
        for word $w \in s$ **do**
            $score = score + C(w, f)/O(w)$
        end for
        if $score > scoreOfCurrentBestFeature$ **then**
            $currentBestFeature = f$
            $scoreOfCurrentBestFeature = score$
        end if
    end for
    if $scoreOfCurrentBestFeature > threshold$ **then**
        Assign $currentBestFeature$ to s as its implicit feature
    end if
end for

all features beyond the first will be missed by the algorithm. Another limitation is the obvious need for labeled data. Since this method is trained, not on explicit features, which can be determined by some other method, but on annotated implicit features, a sufficient amount of annotated data is required for our method to work properly.

4 Data Analysis

This section presents an overview of the two data sets that are used to train and evaluate the proposed method and its variants. The first data set is a collection of product reviews [6], where both explicit and implicit features are labeled. The second data set consists of restaurant reviews [4], where explicit aspects are labeled, as well as implicit aspect categories. Each sentence can have zero or more of these coarse-grained aspect categories. The restaurant set features five different aspect categories: 'food', 'service', 'ambience', 'price', and 'anecdotes/miscellaneous'. Since these aspects are implied by the sentence instead of being referred to explicitly, they function as implicit features as well. However, since there are only five options to choose from, it is much easier to obtain good performance on the restaurant set compared to the product set, where there are many different implicit features. Because of this, results for both data sets are not directly comparable. Even so, it is interesting to see how the proposed method performs on different data.

4.1 Product Reviews

The collection of product reviews are extracted from amazon.com, covering five different products: Apex AD2600 Progressive-scan DVD player, Canon G3, Creative Labs Nomad Jukebox Zen Xtra 40GB, Nikon Coolpix 4300, and Nokia 6610. Because the primary purpose of this data set is to perform aspect-level sentiment analysis, it is the case that features are only labeled as a feature when an opinion is expressed about that feature in the same sentence. In the example below, both sentences have a feature 'camera', but only in the second sentence is 'camera' labeled as a feature since only in the second sentence it is associated with a sentiment word.

> "I took a picture with my phone's *camera*."
> "The *camera* on this phone takes *great* pictures."

Because the product data set contains a lot of different, but sometimes similar, features, a manual clustering step has been performed. This makes the set of features more uniform and reduces unnecessary differences between similar features. It also removes some misspellings that were present in the data set. In total, the number of unique implicit features is reduced from 47 to 25.

As can be seen in Fig. 1, there are not many sentences with an implicit feature. This only stresses the need for a good selection criterion to distinguish the ones with an implicit feature from the ones that do not have one. There is also a small number of sentences (0.2%) that have two implicit features. Since the algorithm will only choose zero or one implicit feature for each sentence, this can potentially impact performance in a negative way. The second implicit feature will always be missed, leading to a lower recall. This is however slightly mitigated by the fact that it is easier to pick a correct feature, as it is checked against both annotated features in the sentence.

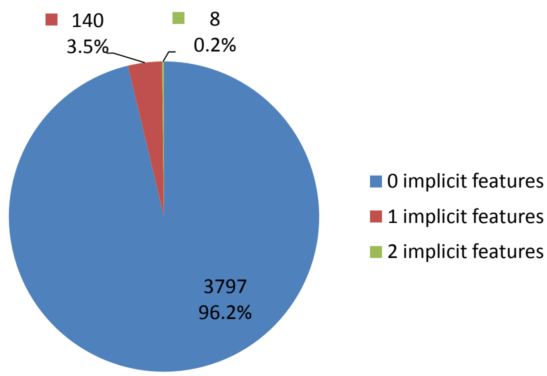

Fig. 1. Distribution of sentences in the product review data set, according to the number of implicit features they contain.

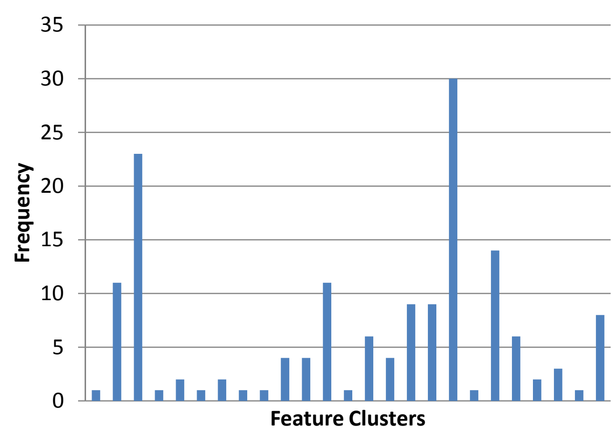

Fig. 2. Frequencies for all 25 unique feature clusters in the product review data set.

In Fig. 2, the frequency distribution of the set of implicit features is given. Frequency is measured as the number of sentences a certain implicit feature appears in. As can be seen, there are quite a few implicit features which appear in only a couple of sentences. Fifteen out of the 25 feature appear in less than 5 sentences, with eight features occurring in only one sentence. This makes it extremely difficult to learn a classifier that is able to find these features. In case of the features that appear only once, it is completely impossible to devise a classifier, since they cannot both appear in the test and in the training set.

4.2 Restaurant Reviews

Compared to the product reviews, the restaurant review data set has clearly different statistical characteristics, as shown in Fig. 3. Where the product review set has only a few sentences that contain an implicit feature, in the restaurant set, all of them have an aspect category, which we will regard as an implicit feature in this research. The much bigger size, together with the already mentioned fact that there are only five different implicit features in this data set, makes for a much easier task. To measure the influence of the threshold parameter, the fifth category of 'anecdotes/miscellaneous' is removed from the data set. Since this category does not really describe a concrete implicit feature, removing it leaves us with sentences that do not have any implicit feature, allowing the performance of the threshold to be assessed on this data as well.

Compared to the product reviews data set, the frequency distribution of the implicit features in the restaurant reviews set, shown in Fig. 4 is more balanced. Every features has at least a couple of hundred sentences in which it is appearing. The one outlier is the 'food' category, which appears twice as much as the second largest feature which is 'service'. Still, the difference between the feature that appears the most ('food') and the one that appears the least ('price') is only a factor of three, whereas for the product features, this would be much higher (i.e., around 30).

5 Evaluation

All evaluations are performed using 10-fold cross-evaluation. Each tenth of the data set is used to evaluate an instance of the algorithm that is trained on the other 90% of the data. Both the co-occurrence frequencies and the threshold parameter are determined based on the training data only. When evaluating the algorithm's output, the following definitions are used:

- $truePositives$ are the features that have been correctly identified by the algorithm;
- $falsePositives$ are those features that have been annotated by the algorithm, that are not present in the golden standard;
- $falseNegatives$ are those features that are present in the golden standard, but that have not been annotated by the algorithm;

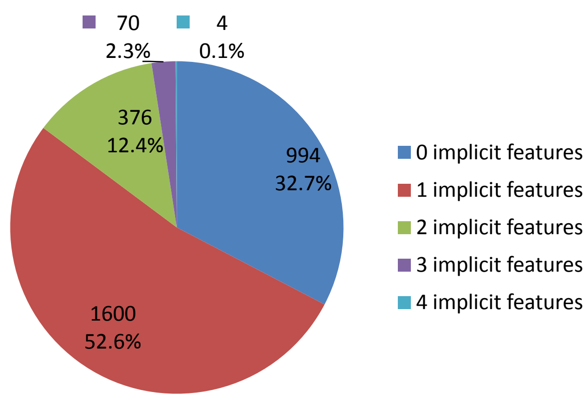

Fig. 3. Distribution of sentences in the restaurant review data set, according to the number of implicit features they contain.

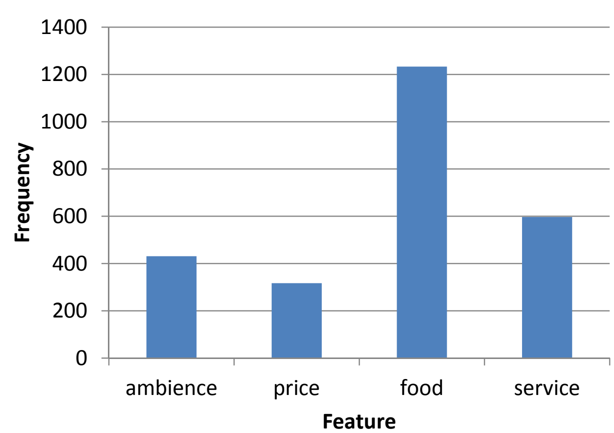

Fig. 4. Frequencies for all 4 unique features in the restaurant review data set.

- $trueNegatives$ are features that are not present in the golden standard, and are correctly not annotated by the algorithm.

When evaluating, a feature always has to be the same one as in the golden feature to count as a true positive. Simply stating that there is some implicit feature in a sentence, which might be true, is not enough. In order to count as a true positive, it has to be the right implicit feature. From this follows that, given a sentence with only one annotated implicit feature and one golden implicit feature, when the algorithm correctly identifies that a sentence contains an implicit feature, but it chooses the wrong one, the wrongly assigned feature will count as a false positive and the annotated one will count as a false negative. As such, both precision and recall will be lower. In general the algorithm can make three kinds of mistakes:

- State that a sentence contains an implicit feature, while actually it does not: precision will be lower;
- State that a sentence does not contain an implicit feature, while actually it does: recall will be lower;
- Correctly stating that a sentence contains an implicit feature, but picking the wrong one: both precision and recall will be lower.

Because of the ten-fold cross-validation, the reported scores are computed on the sum of the ten confusion matrices (i.e., derived from the ten folds). For example, precision would be computed as:

$$precision = \frac{\sum_{fold=1}^{10} truePositives_{fold}}{\sum_{fold=1}^{10} truePositives_{fold} + falsePositives_{fold}}. \quad (2)$$

Recall is computed in a similar way, leaving the F_1-measure, being the harmonic mean of precision and recall, to be computed as usual. In the end, each sentence will be processed exactly once, but will be used nine times as training instance.

The proposed algorithm is tested both with and without the proposed threshold, to assess the benefit of training such a threshold. Furthermore, both versions are evaluated using a Part-of-Speech filter. The latter is used to filter out words in the co-occurrence matrix that may not be useful to find implicit features. Besides evaluating using all words (i.e., including stopwords), both algorithms are evaluated using an exhaustive combination of four word groups, namely nouns, verbs, adjectives, and adverbs.

Since the algorithm without a threshold will generally choose some implicit feature for every sentence, any trained threshold is expected to surpass that score. To provide more insight in this problem, a maximum score is also provided. This maximum score is computed by filtering out all sentences without any implicit feature and then letting the algorithm simply pick the most appropriate feature. This situation reflects a perfect threshold that is always able to make the distinction between the presence or absence of an implicit feature. Obviously, in reality, the trained threshold does not come close to this ideal performance, but including this ideal line allows the separation of errors due to threshold problems from errors due to not picking the right feature. The latter is an intrinsic problem of the algorithm, not of the threshold. With this in mind, one can see that the gap between the ideal line and the bars represents errors that can be attributed to the threshold, while the gap between 100% performance and the ideal line represents errors that can be attributed to the method of using co-occurrence frequencies to find the right feature.

The results on the product review data set are presented in Fig. 5, whereas the results on the restaurant review data set are presented in Fig. 6. In each graph there are two grouped bars for each Part-of-Speech filter, where the first bar shows the performance without a threshold and the second bar the performance with the trained threshold. The line above the bars represents the ideal, or maximum possible, performance with respect to the threshold, as discussed above. There are 16 different Part-of-Speech filters shown in both graphs. The first `all`, simply means that all words, including stopwords, are used in the co-occurrence matrix. The other fifteen filters only allow words of the types that are mentioned, where NN stands for nouns, VB stands for verbs, JJ stands for adjectives, and RB stands for adverbs.

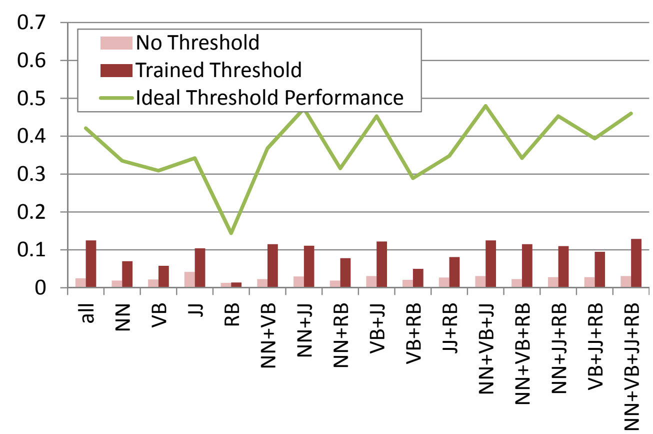

Fig. 5. The performance on the product review data set in F_1-measure for the various PoS-filters.

For the product set, it is beneficial to keep as many words as possible, something that is probably caused by the small size of the data set. However, removing stopwords results in a slightly higher performance: the `NN+VB+JJ+RB` filter scores highest. Looking at the four individual categories, it is clear that adjectives are most important to find implicit features. For the restaurant set, the situation is a bit different. Here, nouns are the most important word group, followed by adjectives. Because of its larger size, it is possible to remove verbs and adverbs

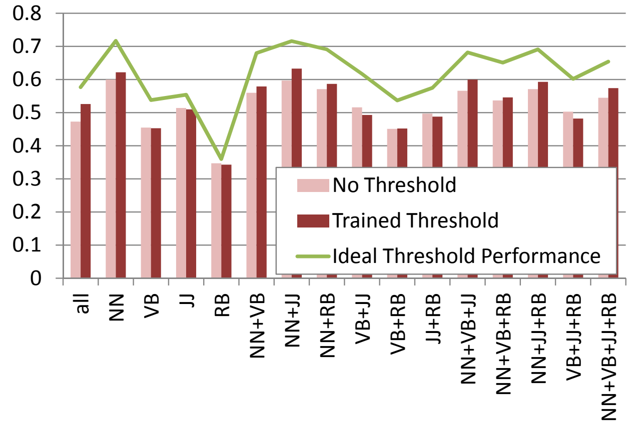

Fig. 6. The performance on the restaurant review data set in F_1-measure for the various PoS-filters.

without any detrimental effects. Hence, the NN+JJ filter yields the best performance.

Another observation we can draw from comparing Fig. 5 and Fig. 6 is that the restaurant set is in general much easier for the algorithm to process. Not only are the ideal performances higher on the restaurant set, also the gap between the ideal and the realized performance is smaller. The most likely reason for this difference is the fact that in the restaurant set there are roughly 2000 sentences that contain at least one of the four possible implicit features, whereas in the product set, there are 140 sentences that contain at least one of 25 possible implicit features. Not only does this render the task of picking the right feature more difficult, it also increases the complexity of judging whether a sentence contains one of these features.

The fact that the vast majority of the product set has no implicit feature at all makes the utilization of a threshold all the more important. This is in contrast to the restaurant set, where two-thirds of the sentences have an implicit feature. Again, this is shown clearly in Fig 5 and Fig 6: the relative improvement of the threshold is much higher for the product data than the restaurant data.

In Fig. 7 and Fig. 8, the precision-recall trade-off is shown for the best scoring Part-of-Speech filter. The restaurant set yields a well-defined curve, which is to be expected due to the large quantity of available data. Note that as in all other graphs, two tasks are being evaluated: determine whether or not there is an implicit feature in a sentence, and if so, determine which one it is. This is the reason that, even with a threshold of zero, the recall will not be 100%: while it does state that every sentence will have an implicit feature, it still has to pick the right one in order to avoid a lower recall (and precision for that matter).

A comparison with the method of Zhang & Zhu [10] is given in Table 1. To increase comparability, both methods are tested with all sixteen possible Part-

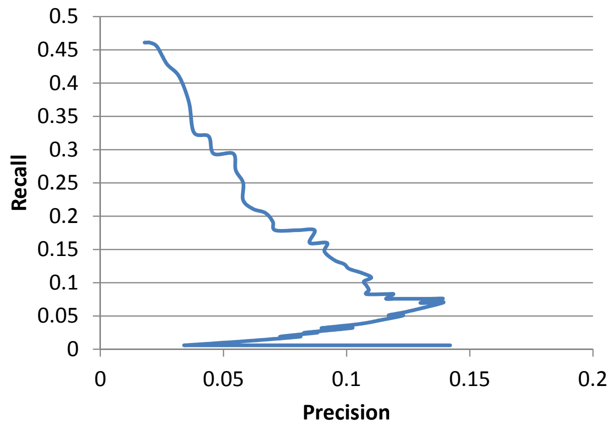

Fig. 7. The precision-recall trade-off on the product review data set, when manipulating the threshold variable (using the `NN+VB+JJ+RB` filter).

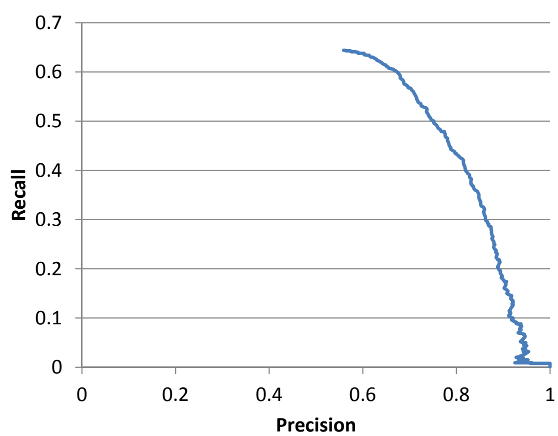

Fig. 8. The precision-recall trade-off on the restaurant review data set, when manipulating the threshold variable (using the `NN+JJ` filter).

of-Speech filters (only the best one is reported). To be fair, the original method is also tested with a threshold added, using the same procedure as for the proposed method, even though this contributes little to its performance.

Interestingly, the effect of the threshold is bigger on the product set compared to the restaurant set. This might point to the fact that training this parameter can partly mitigate the negative effect of having a small data set. Consequently, when the data set is larger, the algorithm on its own already performs quite well, leaving less room for improvement by other methods, like adding a threshold.

Table 1. Comparison of results with Zhang & Zhu [10], with and without the proposed threshold. Reported scores are F_1-measures for the best scoring Part-of-Speech filter. Differences between scores are expressed in percentage points (pp.), the arithmetic difference between two percentages.

product review data set

method	no threshold	trained threshold	difference
Zhang & Zhu	1.2% (`all`)	1.4% (`NN+VB+JJ+RB`)	+0.2 pp.
proposed method	4.2% (`JJ`)	12.9% (`NN+VB+JJ+RB`)	+8.7 pp.
difference	+3 pp.	+11.5 pp.	

restaurant review data set

method	no threshold	trained threshold	difference
Zhang & Zhu	31.5% (`all`)	32.4% (`all`)	+0.9 pp.
proposed method	59.7% (`NN+JJ`)	63.3% (`NN+JJ`)	+3.6 pp.
difference	+28.2 pp.	31.1 pp.	

6 Conclusion

Based on the diagnosed shortcomings in previous work, we proposed a method that directly maps between implicit features and words in a sentence. While the method effectively becomes a supervised one, it is not flawed in its assumptions as previous work, and performance is reported to increase on the two used data sets. Furthermore, a more realistic scenario is implemented wherein the proposed method not only has to determine the right implicit feature, but also whether one is actually present or not.

The proposed algorithm shows a clear improvement with respect to an existing algorithm on the two data sets considered, as it is better in distinguishing between sentences that have an implicit feature and the ones that do not. Both for product reviews and restaurant reviews, the same general improvement is observed when implementing this threshold, even though the actual performance differs much between the two data sets.

Analysis of the performance of the algorithm in relation to the characteristics of the two data sets clearly shows that having less data, but more unique implicit features to detect severely decreases performance. While the proposed algorithm is much better in dealing with this lack of data, the results for that particular data set are still too low to be useful in practice. On the set of restaurant reviews, being of adequate size and having only four unique implicit features, the proposed algorithm yields promising results. Adding a threshold further boosts the performance by another 3 percentage points, which is highly desirable for this kind of user generated content.

A primary suggestion for future work is to learn a threshold for each individual implicit feature, instead of one general threshold that applies to all implicit features. We hypothesize that because some features are used more often in an implicit way than others, and the sentential context differs from feature to feature as well, it makes sense to learn a different threshold for each unique implicit feature.

Also interesting could be to adjust the algorithm to be able to choose more than one implicit feature. Especially on the restaurant set, where about 14% of the sentences have more than one implicit feature, performance could be improved. Possible ways of doing this include choosing all features whose score exceeds the threshold, or employ a classifier that determines how many implicit features are likely to be present. The latter could also be investigated as a possible alternative for the threshold.

Last, a move from word based methods, like this one, toward concept-based methods, as advocated in [3], would be interesting as well. For example, cases like:

"This phone doesn't fit in my pocket."

is very hard to process based on words alone. It is probably feasible to determine that the implicit feature here is 'size', if enough training data is at hand, but determining that this sentence represents a negative sentiment, since mobile phones are *supposed* to fit in ones pocket, seems extremely hard for word-based methods. While concept level methods are still in their infancy, they might be up to this challenge, since common sense knowledge, world knowledge, and domain knowledge are integrated in such an approach.

Acknowledgments

The authors are partially supported by the Dutch national program COMMIT. We also would like to thank Andrew Hagens, Gino Mangnoesing, Lotte Snoek, and Arno de Wolf for their contributions to this research.

References

1. B. Bickart and R. M. Schindler. Internet Forums as Influential Sources of Consumer Information. *Journal of Interactive Marketing*, 15(3):31–40, 2001.

2. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
3. E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New Avenues in Opinion Mining and Sentiment Analysis. *IEEE Intelligent Systems*, 28(2):15–21, 2013.
4. G. Ganu, N. Elhadad, and A. Marian. Beyond the Stars: Improving Rating Predictions using Review Content. In *Proceedings of the 12th International Workshop on the Web and Databases (WebDB 2009)*, 2009.
5. Z. Hai, K. Chang, and J. Kim. 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, 2011.
6. M. Hu and B. Liu. Mining and Summarizing Customer Reviews. In *Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004)*, pages 168–177. ACM, 2004.
7. G. Qiu, B. Liu, J. Bu, and C. Chen. Opinion Word Expansion and Target Extraction through Double Propagation. *Computational Linguistics*, 37(1):9–27, 2011.
8. Q. Su, X. Xu, H. Guo, Z. Guo, X. Wu, X. Zhang, B. Swen, and Z. Su. Hidden Sentiment Association in Chinese Web Opinion Mining. In *Proceedings of the 17th International Conference on World Wide Web (WWW 2008)*, pages 959–968. ACM, 2008.
9. W. Wang, H. Xu, and W. Wan. Implicit Feature Identification via Hybrid Association Rule Mining. *Expert Systems with Applications*, 40(9):3518–3531, 2013.
10. Y. Zhang and W. Zhu. 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, 2013.