Implicit Feature Detection for Sentiment Analysis in Consumer Reviews

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 increasing popularity of aspect-level sentiment analysis, where sentiment is attributed to the actual aspects, or features, on which it is uttered, much attention is given to the problem of detecting these features. While most aspects appear as literal words, some are instead implied by the choice of words. With research in aspect detection advancing, we shift our focus to the less researched group of implicit features. By leveraging the co-occurrence between a set of known implicit features and notional words, we are able to predict the implicit feature based on the choice of words in a sentence. Using two different types of consumer reviews (product reviews and restaurant reviews), an F_1-measure of 38% and 68% is obtained on these data sets, respectively.

1 Introduction

Every day a vast amount of consumer reviews are written on the Web, where customers express their opinions about a product or service [4]. Not only do they describe their general sentiment or attitude towards the product or service, oftentimes specific aspects or features of that product or service are discussed in great detail [9]. This leaves researchers and companies alike with a valuable source of information about consumer sentiment.

Aggregated aspect-level sentiment analysis is a valuable source of information when a company is introducing a new product and it want to create a hype [3]. Carefully managing sentiment of potential customers is paramount to succeeding in creating buzz for the new product. For products or services that already exists, this detailed and often honest information that can be extracted from customer reviews is useful to improve the service or product. On some forums or similar websites, companies are actively involved in responding to negative reviews or complaints made by customers.

For customers, on the other hand, reviews are a primary source of information about a product they wish to buy. In fact, product reviews have been shown to influence potential customers more than official information on the vendor's website [1]. Investigating this relation between company information and consumer generated information can beneficial to improve sales [2]. One can even state that opinions on the Web have become a resource for companies, not unlike word-of-mouth [6].

However, in order to achieve the fine grained information that is needed for such analyses, the various aspects, or features, of a product or service must be recognized in the text first. Examples of such features include 'price', 'service', parts of a product like 'battery', or different meals and ingredients for restaurants. In most cases, these features are literally mentioned in the text:

> "All the appetizers and salads were fabulous, the steak was mouth watering and the pasta was delicious!!!"

In this sentence, taken from the data set of restaurant reviews [5], 'appetizers', 'salads', 'steak', and 'pasta' are all features on which sentiment is expressed. However, this is not always the case, as demonstrated in the example below, which is taken from the product review data set [8]:

> "I like my phones to be small so I can fit it in my pockets."

Evidently, the feature referred to here is the size of the product, even though the word 'size' is never mentioned. However, words like 'small' and 'fit' give away that the feature implied here is the product's size. Unfortunately, detecting the implicit features is not always this straightforward.

> "I love the fact I can carry it in my shirt or pants pocket and forget about it."

The above example is an actual sentence in the product review data set, and according to the available annotations, the implicit feature in this case is also its size, however, it is easy to see that weight would also have been a good candidate. In fact, with only this sentence, one will not be able to distinguish between one or the other. This makes detecting implicit features a complex endeavor.

2 Related Work

One of the first works to focus on implicit feature detection is [11], where implicit features are found using semantic association analysis based on Point-wise Mutual Information. Unfortunately, no quantitative results were reported, so it is hard to assess the strength of this method.

A method based on co-occurrence Association Rule Mining is proposed in [7]. First a set of opinion words and a set of features is created from the sentences in the data set that have an annotated explicit feature. Next, a co-occurrence matrix is constructed representing the co-occurrence frequency between each opinion word and each feature. Then, association rule mining is performed for each opinion word, mapping the opinion word as the antecedent to potential features as consequents. These rule consequents are clustered in order to yield more robust rules. When an opinion word is encountered that has no associated explicit feature, the list of rules is checked, firing the one with the majority feature cluster. The word that is representative of that cluster is then assigned to the opinion word as its implicit feature. This approach is evaluated on a custom data set of Chinese mobile phone reviews, yielding an F$_1$-measure of 0.74.

Instead of association rule mining, [12] uses the idea of double propagation [10] to iteratively construct a matrix M that represent which opinion words are linked to which features. Then, a matrix C is constructed that holds the co-occurrence frequencies between each explicit feature and each notional word in the text corpus. To determine what feature is implied by a given sentence, a shortlist of possible features is compiled by retrieving all features from M that are sufficiently linked to the set of opinion words in the given sentence. For the features in this list, referred to as F_c, a score $T(f_i)$ is computed based on the co-occurrence frequency between $f_i \in F_c$ and the set of notional words in the sentence, W_-, with the minus denoting the fact that notional words which are also present in F_c as features are removed in W_-.

$$T(f_i) = \frac{1}{v} \sum_{j=1}^{v} \frac{P(f_i|w_j)}{v}, \tag{1}$$

where v is the number of notional words, f_i is the ith feature for which the $T(f_i)$ score is computed, and w_j is the jth notional word in W_-.

Hence, the $T(f_i)$ score can be thought of as the average conditional probability of a feature being implied, given the set of notional words in the sentence. The feature with the highest score is selected as the single implicit feature for that sentence. The method is evaluated on a Chinese corpus of mobile phone reviews and clothes reviews, with an F$_1$-measure of 0.80 and 0.79, respectively. A strong point of this method is that it is completely unsupervised, requiring no annotated training data. Since it relies on co-occurrence frequencies, the training data needs to be sufficiently large to have reliable frequency numbers.

The method from [12] thus uses explicit features to find the implicit ones. This has several drawbacks, the first of which is that this assumes that the set of unique explicit features in the text corpus is identical to, or at least contains, the set of unique implicit features in the text corpus. Because T scores are only computed for explicit features, only those can be selected as the implicit feature for a sentence. However, certain aspects like weight, size, and price are usually referred to implicitly, depriving the algorithm of the chance to properly include these terms in the co-occurrence matrix. For example, it is much more likely to encounter the first sentence than the second:

> "This phone is too heavy and too expensive."
> "The weight of this phone is too high, its price as well."

Hence, the assumption is made that implicit features also occur as explicit features in the same data set. Furthermore, it links explicit features to a context of notional words, thereby assuming that when the same feature is implied, it is implied by the same context of notional words that is also present when the feature is explicitly mentioned.

An observation that seems to contradict these assumption is the notion that implicit features can often be characterized as coarse-grained features, not unlike categories, whereas explicit features are usually detailed, fine-grained aspects of the product or service. So, while more general, coarse-grained features can be implied by a sentence, it is much harder to do the same for very detailed features. This makes sense, since implications are based on a mapping, that is shared among readers, between the used words and the implied concepts. Such mappings are only useful when frequently used, which cannot be the case for very detailed features. The first example given, shown again below, demonstrates this fact. Four very specific meals are mentioned, which are explicit features of this sentence, but it is hard to imagine these four things being implied in some sentence. It is however easy to see that the coarser feature 'food' is implied in this sentence.

> "All the appetizers and salads were fabulous, the steak was mouth watering and the pasta was delicious!!!"

3 Method

This research addresses the issues raised in the previous section, by revising and extending the work in [12]. We have chosen [12] due to its good reported performance and low complexity. This section will discuss both the proposed major revision to [12], as well as one minor addition.

3.1 Training using Implicit Feature Annotations

To deal with the two violated assumptions mentioned in Sect. 2, a small but significant change is made in the construction of the co-occurrence matrix between F_c and W_-. Instead of using explicit features as F_c, the annotated implicit features are used as F_c. This results in direct co-occurrence data between notional words and the implicit features that are to be determined. This change renders the two violated assumptions irrelevant, but introduces a dependence on annotated data. Thus, it requires the splitting of data into a training set, which is used to count the co-occurrences between implicit features and notional words, and a test set, which is used to test the predictions of the algorithm.

3.2 Part-of-Speech Filter

In the basic version of this algorithm, all words are considered as notional words. However, stopwords, like determiners, prepositions, numbers, etc., are often removed before text analysis to improve the results. In a similar fashion, all words that are not nouns, verbs, adjectives, or adverbs are not taken into account when creating the co-occurrence matrix C. Since it is unknown which of these four word groups will contribute most to finding implicit features, all combinations should be tested.

4 Data Analysis

In this section we will give an insight into the data sets that are used to train and evaluate the proposed methods. The primary data set is the set of product reviews from [8]. In this data set, both explicit and implicit features are annotated. The secondary data set is a set of restaurant reviews from [5]. Here, explicit aspects are annotated, along with aspect categories. The latter consists of five coarse-grained features (e.g., 'food', 'service', 'ambience', 'price', and 'anecdotes/miscellaneous'). Because these categories are not literally mentioned in the sentence, but implied by the choice of words, they function as implicit features as well. The fact that there are only five implicit features to choose from obviously makes it much easier for the algorithm to pick the right one, and in that sense the results on both data sets are not directly comparable. Nevertheless, it is of interest to see how well the algorithm and the proposed revisions perform on different data.

4.1 Product Reviews

The product review data contains customer reviews from amazon.com for five different products: Apex AD2600 Progressive-scan DVD player, Canon G3, Creative Labs Nomad Jukebox Zen Xtra 40GB, Nikon Coolpix 4300, and Nokia 6610. In this data set, a feature is only annotated as such if an opinion is specified about this feature. Below, a short example is given, where in the first sentence the feature 'software' is only mentioned. In the second sentence an opinion ('great') is given about the feature ('software'). In the product review data set the feature 'software' would only have been annotated by the manual annotators as a feature in the second sentence, because only in that sentence an opinion is given about the feature.

"Installed *software*."
"The *software* is *great*."

The occurrences of implicit features per sentence are shown in Table 1. Sentences without explicit features contain no more than 2 implicit features, and small fraction of those sentences (0.3%) contain 2 implicit features. In Section 3, the assumption that only the best candidate feature per sentence should be chosen (and thus multiple implicit features per sentence are not allowed) will therefore only be restrictive for a small fraction of sentences. Consequently, it is safe to only estimate just 1 implicit feature at most, which discards the issue of determining exactly how many implicit features the sentence will contain.

Another assumption is that only an implicit feature is estimated in a sentence where there is no explicit feature already present. This assumption is also made by [12]. For every implicit feature in the data set, it is checked whether that sentence also contains an explicit feature. Table 1 shows the number of occurrences of implicit features with and without explicit features. For all products, about 84% of the implicit features appear in a sentence without an explicit feature. Thus the assumption removes only 16% of the sentences with implicit features.

Table 1: General statistics of the product review data.

nr. of implicit features	none	1	2	3 or more
all sentences	3797	140	8	0
sentences without explicit features	2726	119	5	0

4.2 Restaurant Reviews

The restaurant review set is quite different in terms of statistics compared to the product review set, as shown in Table 2. As mentioned before, there are only five different implicit features, and each sentence has at least one annotated implicit feature. This effectively results in a much larger data set: where the product review set has only 148 sentences with one or more implicit features, the restaurant review set has over 3000 sentences with at least one implicit feature. However, another assumption made in the original method was that an implicit feature is only determined for sentences having no explicit feature. While this might work for the product review set, it would result in throwing away about 75% of the data set. Furthermore, it was assumed that sentences will have at most one implicit feature. Again, while this assumption works for the product review set, it would result in throwing away roughly 19% of the restaurant data. For this reason, we will evaluate the algorithm and its variants both with and and without enforcing these assumptions on the data set.

Table 2: General statistics of the restaurant review data.

nr. of implicit features	none	1	2	3	4	5 or more
all sentences	0	2468	488	82	6	0
sentences without explicit features	0	989	32	0	0	0

4.3 Data Quality

The implicit features in the data sets are determined manually, making it prone to human errors. Even for human readers, it is not always obvious to what feature a reviewer is referring! The sentence below shows an example of such an error.

> “I didn’t think I would find this quality and ease of use for under $ 1500 - I’m thrilled with my purchase!”

In this sentence, ‘camera’ is indicated as implicit feature, because the sentence is very general about the product. On the other hand, ‘quality’ and ‘use’ are explicitly referred to in the sentence, and ‘price’ seems a better choice for the implicit feature for this sentence, because the sentence indicates that this customer believes that the price is low for this product. Choosing both ‘price’ and ‘camera’ as implicit features for this sentence seems too much, because they are both pointing to the same opinion.

In addition, sometimes a feature is indicated as implicit, even though it is explicitly named in the sentence. In the example below, ‘performance’ is indicated as an implicit feature, but since the feature literally appears in the sentence, it is actually an explicit feature.

> “I bought it for my trip to Buenos Aires, and also used it at the Iguazu Falls, and could not have asked for more perfect *performance*!”

5 Results Analysis

All evaluations are performed using 10-fold cross-validation, with all sentences without an implicit feature being removed from the test set. The evaluation metric is the F_1-measure, although precision and recall are also given.

The first step is to evaluate the decision to train the algorithm by counting co-occurrences with annotated implicit features instead of using explicit features, which requires no training. As mentioned in the previous section, the original paper enforces strict assumptions on the test data: no explicit feature and exactly one implicit feature should be present in the sentence. Since these assumptions can have a large impact on the quantity of the data and thus on the outcome of the evaluation, the first test is conducted for four levels of assumptions. The first is the strict level, where both assumptions are enforced on the test set (`strict`), resulting in all sentences that do not comply with the assumption being removed from the test set. The second (allow sentence with explicit feature(s) to remain in the test set: `allow_explicit`) and third (allow sentences with more than one implicit feature to remain in the test set: `allow_multi`) level each remove one of the assumptions, whereas the fourth allows both (`allow_both`). The results using the product review data set are shown in Table 3, whereas the performance on the restaurant review data set is reported in Table 4.

Evidently, training on implicit feature annotations, using those to construct the co-occurrence matrix is very helpful compared to using the explicit features. Interestingly, the two data sets feature many differences with respect to the relative performance of the various parameter settings. First, the assumption levels seem to have a different effect, depending on the data set and the used algorithm. Second, the Part-of-Speech filter, while definitely having impact on the revised method, doesn’t have any notable effect on the original method in most cases.

The assumptions, as mentioned in the original research of [12], are indeed useful for the original method. As can be seen in Table 3, the original method, when used with the product reviews data set, performs best under `strict` assumptions. For the revised method however, this is the exact opposite. Here, the `strict` level actually performs the worst, and `allow_both`, which is least

Table 3: Evaluation results on the product review data set.

	assumption level	strict		allow_explicit		allow_multi		allow_both	
PoS filter	method	original	revised	original	revised	original	revised	original	revised
only_NN	precision:	0.17	0.19	0.15	0.18	0.18	0.20	0.15	0.20
	recall:	0.17	0.19	0.15	0.18	0.17	0.19	0.14	0.19
	F_1:	0.17	0.19	0.15	0.18	0.17	0.20	0.14	0.19
only_VB	precision:	0.18	0.24	0.15	0.23	0.18	0.26	0.15	0.26
	recall:	0.18	0.24	0.15	0.23	0.17	0.25	0.14	0.24
	F_1:	0.18	0.24	0.15	0.23	0.17	0.25	0.14	0.25
only_JJ	precision:	0.18	0.21	0.15	0.24	0.18	0.24	0.15	0.28
	recall:	0.18	0.21	0.15	0.24	0.17	0.23	0.14	0.26
	F_1:	0.18	0.21	0.15	0.24	0.17	0.24	0.14	0.27
only_RB	precision:	0.18	0.13	0.15	0.12	0.18	0.15	0.15	0.14
	recall:	0.18	0.13	0.15	0.12	0.17	0.14	0.14	0.13
	F_1:	0.18	0.13	0.15	0.12	0.17	0.14	0.14	0.13
only_NN_VB	precision:	0.17	0.26	0.15	0.26	0.18	0.27	0.15	0.28
	recall:	0.17	0.26	0.15	0.26	0.17	0.26	0.14	0.27
	F_1:	0.17	0.26	0.15	0.26	0.17	0.27	0.14	0.28
only_NN_JJ	precision:	0.18	0.29	0.15	0.31	0.18	0.31	0.15	0.34
	recall:	0.18	0.29	0.15	0.31	0.17	0.30	0.14	0.33
	F_1:	0.18	0.29	0.15	0.31	0.17	0.31	0.14	0.34
only_NN_RB	precision:	0.18	0.20	0.15	0.19	0.18	0.23	0.15	0.22
	recall:	0.18	0.20	0.15	0.19	0.17	0.22	0.14	0.21
	F_1:	0.18	0.20	0.15	0.19	0.17	0.22	0.14	0.22
only_VB_JJ	precision:	0.18	0.29	0.15	0.32	0.18	0.32	0.15	0.36
	recall:	0.18	0.29	0.15	0.32	0.17	0.31	0.14	0.34
	F_1:	0.18	0.29	0.15	0.32	0.17	0.32	0.14	0.35
only_VB_RB	precision:	0.18	0.23	0.15	0.23	0.18	0.25	0.15	0.26
	recall:	0.18	0.23	0.15	0.23	0.17	0.24	0.14	0.24
	F_1:	0.18	0.23	0.15	0.23	0.17	0.25	0.14	0.25
only_JJ_RB	precision:	0.18	0.23	0.15	0.25	0.18	0.26	0.15	0.28
	recall:	0.18	0.23	0.15	0.25	0.17	0.25	0.14	0.27
	F_1:	0.18	0.23	0.15	0.25	0.17	0.25	0.14	0.28
only_NN_VB_JJ	precision:	0.18	**0.33**	0.15	**0.35**	0.18	**0.35**	0.15	**0.39**
	recall:	0.18	**0.33**	0.15	**0.35**	0.17	**0.34**	0.14	**0.37**
	F_1:	0.18	**0.33**	0.15	**0.35**	0.17	**0.35**	0.14	**0.38**
only_NN_VB_RB	precision:	0.17	0.24	0.15	0.24	0.18	0.26	0.15	0.27
	recall:	0.17	0.24	0.15	0.24	0.17	0.25	0.14	0.26
	F_1:	0.17	0.24	0.15	0.24	0.17	0.25	0.14	0.26
only_NN_JJ_RB	precision:	0.18	0.29	0.15	0.31	0.18	0.32	0.15	0.34
	recall:	0.18	0.29	0.15	0.31	0.17	0.31	0.14	0.33
	F_1:	0.18	0.29	0.15	0.31	0.17	0.32	0.14	0.34
only_VB_JJ_RB	precision:	0.18	0.27	0.15	0.29	0.18	0.30	0.15	0.32
	recall:	0.18	0.27	0.15	0.29	0.17	0.29	0.14	0.31
	F_1:	0.18	0.27	0.15	0.29	0.17	0.29	0.14	0.32
only_NN_VB_JJ_RB	precision:	0.18	0.31	0.15	0.33	0.18	0.34	0.15	0.36
	recall:	0.18	0.31	0.15	0.33	0.17	0.33	0.14	0.34
	F_1:	0.18	0.31	0.15	0.33	0.17	0.33	0.14	0.35
all	precision:	**0.20**	0.28	0.15	0.31	0.18	0.31	0.15	0.33
	recall:	**0.20**	0.28	0.15	0.31	0.17	0.30	0.14	0.31
	F_1:	**0.20**	0.28	0.15	0.31	0.17	0.31	0.14	0.32

Table 4: Evaluation results on the restaurant review data set.

	assumption level	strict		allow_explicit		allow_multi		allow_both	
PoS filter	method	original	revised	original	revised	original	revised	original	revised
only_NN	precision:	0.04	0.55	0.15	0.66	0.05	0.55	0.23	0.70
	recall:	0.04	0.55	0.15	0.66	0.05	0.54	0.19	0.57
	F_1:	0.04	0.55	0.15	0.66	0.05	0.54	0.21	0.63
only_VB	precision:	0.04	0.44	0.15	0.48	0.05	0.44	0.23	0.51
	recall:	0.04	0.44	0.15	0.48	0.05	0.43	0.19	0.42
	F_1:	0.04	0.44	0.15	0.48	0.05	0.44	0.21	0.46
only_JJ	precision:	0.04	0.23	0.15	0.40	0.05	0.25	0.23	0.47
	recall:	0.04	0.23	0.15	0.40	0.05	0.24	0.19	0.38
	F_1:	0.04	0.23	0.15	0.40	0.05	0.24	0.21	0.42
only_RB	precision:	0.04	0.32	0.15	0.30	0.05	0.32	0.23	0.34
	recall:	0.04	0.32	0.15	0.30	0.05	0.31	0.19	0.28
	F_1:	0.04	0.32	0.15	0.30	0.05	0.32	0.21	0.31
only_NN_VB	precision:	0.04	0.63	0.15	**0.68**	0.05	0.63	0.23	**0.71**
	recall:	0.04	0.63	0.15	**0.68**	0.05	0.62	0.19	**0.58**
	F_1:	0.04	0.63	0.15	**0.68**	0.05	0.63	0.21	**0.64**
only_NN_JJ	precision:	0.04	0.54	0.15	0.66	0.05	0.55	0.23	0.70
	recall:	0.04	0.54	0.15	0.66	0.05	0.53	0.19	0.58
	F_1:	0.04	0.54	0.15	0.66	0.05	0.54	0.21	0.63
only_NN_RB	precision:	0.04	0.61	0.15	0.67	0.05	0.62	0.23	0.70
	recall:	0.04	0.61	0.15	0.67	0.05	0.60	0.19	0.58
	F_1:	0.04	0.61	0.15	0.67	0.05	0.61	0.21	0.64
only_VB_JJ	precision:	0.04	0.47	0.15	0.55	0.05	0.48	0.23	0.59
	recall:	0.04	0.47	0.15	0.55	0.05	0.46	0.19	0.49
	F_1:	0.04	0.47	0.15	0.55	0.05	0.47	0.21	0.54
only_VB_RB	precision:	0.04	0.49	0.15	0.50	0.05	0.49	0.23	0.53
	recall:	0.04	0.49	0.15	0.50	0.05	0.48	0.19	0.44
	F_1:	0.04	0.49	0.15	0.50	0.05	0.49	0.21	0.48
only_JJ_RB	precision:	0.04	0.38	0.15	0.47	0.05	0.39	0.23	0.53
	recall:	0.04	0.38	0.15	0.47	0.05	0.38	0.19	0.43
	F_1:	0.04	0.38	0.15	0.47	0.05	0.38	0.21	0.48
only_NN_VB_JJ	precision:	0.04	0.62	0.15	**0.68**	0.05	0.62	0.23	**0.71**
	recall:	0.04	0.62	0.15	**0.68**	0.05	0.60	0.19	**0.58**
	F_1:	0.04	0.62	0.15	**0.68**	0.05	0.61	0.21	**0.64**
only_NN_VB_RB	precision:	0.04	**0.64**	0.15	0.66	0.05	**0.64**	0.23	0.69
	recall:	0.04	**0.64**	0.15	0.66	0.05	**0.62**	0.19	0.57
	F_1:	0.04	**0.64**	0.15	0.66	0.05	**0.63**	0.21	0.62
only_NN_JJ_RB	precision:	0.04	0.59	0.15	0.67	0.05	0.60	0.23	0.70
	recall:	0.04	0.59	0.15	0.67	0.05	0.58	0.19	0.58
	F_1:	0.04	0.59	0.15	0.67	0.05	0.59	0.21	0.63
only_VB_JJ_RB	precision:	0.04	0.51	0.15	0.56	0.05	0.51	0.23	0.60
	recall:	0.04	0.51	0.15	0.56	0.05	0.49	0.19	0.49
	F_1:	0.04	0.51	0.15	0.56	0.05	0.50	0.21	0.54
only_NN_VB_JJ_RB	precision:	0.04	0.62	0.15	0.66	0.05	0.62	0.23	0.70
	recall:	0.04	0.62	0.15	0.66	0.05	0.60	0.19	0.57
	F_1:	0.04	0.62	0.15	0.66	0.05	0.61	0.21	0.63
all	precision:	0.04	0.57	0.15	0.59	0.05	0.58	0.23	0.63
	recall:	0.04	0.57	0.15	0.59	0.05	0.56	0.19	0.52
	F_1:	0.04	0.57	0.15	0.59	0.05	0.57	0.21	0.57

constraining, performs best. A possible reason for this might be that, since the revised method directly uses the data to find implicit features, more available data will yield a better algorithm and thus better performance. The original method, however, uses the found explicit features to find the implicit one. Since this requires two steps, the effect of having more data is diminished. For the restaurant reviews data set, the situation is again different, with the original method performing best under `allow_both`, while the revised method works best with `allow_explicit`. Given the different statistics of both data sets (cf. Tables 1 and 2), the four assumption levels have a much stronger impact on the restaurant data than on the product data.

Keeping sentences with more than one implicit feature in the test set (e.g., `allow_multi` and `allow_both`) will generally result in a slightly higher precision, as the chance will be higher that the one determined feature is either one of the golden features, but the recall will be slightly lower, as only one feature will be estimated and all others will result in false negatives. While on the much smaller product data set, the influence of having more sentences seems to dwarf this effect, it is clearly visible in the `allow_multi` results for the restaurant data. Here, the F_1-measure is actually a bit lower compared with `allow_explicit`.

By selecting which kind of words will be taken into account when counting the co-occurrence frequencies, the information value of the various types of words is made visible. The four groups that are distinguished are nouns (`NN`), verbs (`VB`), adjectives (`JJ`), and adverbs (`RB`). For both data sets, all combinations of these four groups are evaluated.

For the original method on the product review data set, the choice of word groups has no visible effect on performance for three out of the four assumption levels. Only in the case of using `strict`, it is better to simply use all words instead of a filter. For the revised method, however, the Part-of-Speech filter identifies that, for the product data, the combination of nouns, verbs, and adjectives is the most informative when trying to find implicit features, regardless of which assumption level is used. For the restaurant data, all combinations with at least nouns and verbs yield good scores, with the exact one performing best being dependent on the assumption level.

In general, one can conclude that directly creating the co-occurrence matrix with implicit features instead of indirectly with explicit features is a good strategy. The performance gain is significant, which will offset the disadvantage of needing labeled data. In terms of overall performance, the revised algorithm works best with a Part-of-Speech filter that only allows nouns, verbs, and adjectives, with an assumption level of `allow_explicit` for the restaurant data and `allow_both` for the product data. Concerning data sets, the revised algorithm works best with the restaurant data, which is relatively large and has only five different implicit features to choose from. Using the product data results in the worst performance, due to its limited size and increased difficulty: it has more different implicit features than the restaurant data and less instances per unique implicit feature. This makes it hard to properly train the algorithm. For the original method, there is no real difference (e.g., an F_1 of 0.23 on the restaurant data with allow_both and 0.20 on the product data with strict). Here the two-step methodology may also have a dampening influence.

Because of differences in implementation and the use of a different data set, the results of this research are not directly comparable with the results from the two previous studies. One the one hand, the implementation of the original method from [12] is slightly different. Although this might account for a certain decrease in performance compared to the F_1 scores reported in [12], it is still surprising to get such a big difference in performance: our implementation scores an F_1 of 0.20, whereas [12] reports an F_1 of 0.80.

On the other hand, and this might also influence the great difference in F_1 scores, the two previous methods evaluate on a set of Chinese mobile phone reviews and a set of clothes review, while this research uses electronic product reviews and restaurant reviews in English. While the electronic product review set should be comparable to the mobile phone review sets, its size is very small. We hypothesize that, together with only having to choose from among five possible implicit features, the size of the restaurant set is a major factor in explaining the performance difference between the two data sets.

6 Conclusion

The detection of features from reviews is important when measuring consumer sentiment on a fine-grained level. Adding the detection of implicit features, while a difficult task because the features themselves do not appear in the sentence, can increase the overall coverage of an aspect-level sentiment analysis tool. Besides a base method [12], several revisions and extensions were discussed and evaluated on two data sets [5,8].

The main conclusion, based on the performed evaluation, is that it is much better to count the co-occurrence frequency between annotated implicit feature and notional words than to count the co-occurrence frequency between explicit features and notional words. Since the number of implicit features is usually much smaller than the number of explicit features, this will greatly reduce the size of the co-occurrence matrix as well, yielding better performance in terms of system load and processing time. The only drawback would be that this method is more domain dependent, as annotations of implicit features are required to train the system (i.e., do the counting).

Possible directions for future work might include an extension to deal with more than one implicit feature in a sentence. While this is arguably not useful for the product review data, roughly one sixth of the restaurant review sentences has more than one implicit feature, rendering this a good way of reducing the number of false negatives. Another option might be to introduce a weighting scheme for the co-occurrences where the co-occurrence with different words can be weighted differently, based on for example additional domain or world knowledge. This could, for example, be taken from structured data like ontologies.

Acknowledgments

The authors are partially supported by the Dutch national program COMMIT. We also would like to thank Sven van den Berg, Marnix Moerland, Marijn Waltman, and Onne van der Weijde, 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. Y. Chen and J. Xie. Online Consumer Review: Word-of-Mouth as a New Element of Marketing Communication Mix. *Management Science*, 54(3):477–491, 2008.
3. Ellen van Kleef and Hans C.M. van Trijp and Pieternel Luning. Consumer Research in the Early Stages of New Product Development: a Critical Review of Methods and Techniques. *Food Quality and Preference*, 16(3):181–201, 2005.
4. R. Feldman. Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89, 2013.
5. G. Ganu, N. Elhadad, and A. Marian. Beyond the Stars: Improving Ratig Predictions using Review Content. In *Proceedings of the 12th International Workshop on the Web and Databases (WebDB 2009)*, 2009.
6. R. E. Goldsmith and D. Horowitz. Measuring Motivations for Online Opinion Seeking. *Journal of Interactive Advertising*, 6(2):3–14, 2006.
7. 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.
8. 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.
9. B. Liu. *Sentiment Analysis and Opinion Mining*, volume 16 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool, 2012.
10. 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.
11. 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.
12. 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.