Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis With Co-Occurrence Data

Kim Schouten, Onne van der Weijde, Flavius Frasincar, and Rommert Dekker

***Abstract*—Using online consumer reviews as electronic word of mouth to assist purchase-decision making has become increasingly popular. The Web provides an extensive source of consumer reviews, but one can hardly read all reviews to obtain a fair evaluation of a product or service. A text processing framework that can summarize reviews, would therefore be desirable. A subtask to be performed by such a framework would be to find the general aspect categories addressed in review sentences, for which this paper presents two methods. In contrast to most existing approaches, the first method presented is an unsupervised method that applies association rule mining on co-occurrence frequency data obtained from a corpus to find these aspect categories. While not on par with state-of-the-art supervised methods, the proposed unsupervised method performs better than several simple baselines, a similar but supervised method, and a supervised baseline, with an F_1-score of 67%. The second method is a supervised variant that outperforms existing methods with an F_1-score of 84%.**

***Index Terms*—Aspect category detection, consumer reviews, co-occurrence data, sentiment analysis, spreading activation.**

I. Introduction

WORD of mouth (WoM) has always been influential on consumer decision-making. Family and friend are usually asked for advice and recommendations before any important purchase-decisions are made. These recommendations can both have short as well as long term influence on consumer decision-making [1].

With the Web, WoM has greatly expanded. Anyone who wishes to share their experiences, can now do so electronically. Social media, like Twitter and Facebook allow for easy ways to exchange statements about products, services, and brands. The term for this expanded form of WoM is electronic WoM (EWoM).

Over the last few years, EWoM has become increasingly popular [2]. One of the most important forms of EWoM communication are product and service reviews [3] posted on the Web by consumers. Retail companies such as Amazon and Bol have numerous reviews of the products they sell, which provide a wealth of information, and sites like Yelp offer detailed consumer reviews of local restaurants, hotels, and other businesses. Research has shown these reviews are considered more valuable for consumers than market-generated information and editorial recommendations [4]–[6], and are increasingly used in purchase decision-making [7].

The information that can be obtained from product and service reviews is not only beneficial to consumers, but also to companies. Knowing what has been posted on the Web can help companies improve their products or services [8].

However, to effectively handle the large amount of information available in these reviews, a framework for the automated summarization of reviews is desirable [9]. An important task for such a framework would be to recognize the topics (i.e., characteristics of the product or service) people write about. These topics can be fine-grained, in the case of aspect-level sentiment analysis, or more generic in the case of aspect categories. For example, in the following sentence, taken from a restaurant review set [10], the fine-grained aspects are "fish," "rice," and "seaweed" whereas the aspect category is "food."

> "My goodness, everything from the fish to the rice to the seaweed was absolutely amazing."

As one can see, aspect categories are usually implied, that is, the names of the categories are not explicitly mentioned in the sentence. The same holds for fine-grained aspects: while most of them are referred to explicitly in a sentence, some are only implied by a sentence. For example, in the sentence below, the implied fine-grained aspect is "staff," whereas the implied aspect category is "service."

> "They did not listen properly and served me the wrong dish!"

When the aspect categories are known beforehand, and enough training data is available, a supervised machine learning approach to aspect category detection is feasible, yielding a high performance [11]. Many approaches to find aspect categories are supervised [11]–[14]. However, sometimes the flexibility inherent to an unsupervised method is desirable.

The task addressed in this paper stems from a subtask of the SemEval-2014 Challenge [10], which purpose is to identify aspect categories discussed in sentences, given a

Manuscript received October 12, 2016; revised February 20, 2017; accepted March 21, 2017. This work was supported by the Dutch National Program COMMIT. This paper was recommended by Associate Editor S. Ventura. *(Corresponding author: Flavius Frasincar.)*

K. Schouten, F. Frasincar, and R. Dekker are with the Econometric Institute, Erasmus University Rotterdam, 3062 PA Rotterdam, The Netherlands (e-mail: schouten@ese.eur.nl; frasincar@ese.eur.nl; rdekker@ese.eur.nl).

O. van der Weijde is with OneUp.Company, 1066 EP Amsterdam, The Netherlands (e-mail: onnevanderweijde@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2688801

set of aspect categories. The sentences come from customer reviews and should be classified into one or more aspect categories based on its overall meaning. For example, given the set of aspect categories (*food*, *service*, *price*, *ambience*, and *anecdotes/miscellaneous*), two annotated sentences are as follows.

"The food was great." → (*food*)

"It is very overpriced and not very tasty." → (*price, food*)

As shown in the above examples, aspect categories do not necessarily occur as explicit terms in sentences. While in the first sentence food, is mentioned explicitly, in the second sentence it is done implicitly. In our experiments all sentences are assumed to have at least one aspect category present. Because it may not always be clear which category applies to a sentence, due to incomplete domain coverage of the categories and the wide variation of aspects a reviewer can use, a "default" category is used. An example of a sentence where a default category is used, is presented below. Here, the second part of the sentence ("but everything else ... is the pits.") is too general to classify it as one of the other categories (i.e., food, service, price, and ambience).

"The food is outstanding, but everything else about this restaurant is the pits." → (*food*, *anecdotes/miscellaneous*)

In this paper, both an unsupervised and a supervised method are proposed that are able to find aspect categories based on co-occurrence frequencies. The unsupervised method uses spreading activation on a graph built from word co-occurrence frequencies in order to detect aspect categories. In addition, no assumption has to be made that the implicit aspects are always referred to explicitly, like it is done in [15]. The proposed unsupervised method uses more than just the literal category label by creating a set of explicit lexical representations for each category. The only required information is the set of aspect categories that is used in the data set. The supervised method on the other hand uses the co-occurrences between words, as well as grammatical relation triples, and the annotated aspect categories to calculate conditional probabilities from which detection rules are mined.

This paper is structured as follows. First, in Section II, an overview of the related work that inspired this paper is presented, then Section III gives the details of the proposed unsupervised method, while Section IV discusses the details of the supervised method. Section V contains the evaluation of both methods, comparing them to several baselines and to two state-of-the-art methods. Last, in Section VI, the conclusions are drawn and some pointers for future work are given.

II. Related Work

Since most aspect categories are left implicit in text,[1] methods for detecting implicit fine-grained aspects might be used for aspect categories as well. As such, some works on implicit aspect detection that inspired this paper are discussed below. For a comprehensive survey on detecting both explicit and implicit aspects, and their associated sentiment, we refer the reader to [16].

An early work on implicit aspect detection is [17]. The authors propose to use semantic association analysis based on point-wise mutual information (PMI) to differentiate implicit aspects from single notional words. Unfortunately, there were no quantitative experimental results reported in their work, but intuitively the use of statistical semantic association analysis should allow for certain opinion words such as "large," to estimate the associated aspect ("size").

In [18], an approach is suggested that simultaneously and iteratively clusters product aspects and opinion words. Aspects/opinion words with high similarity are clustered together, and aspects/opinion words from different clusters are dissimilar. The similarity between two aspects/opinion words is measured by fusing both homogeneous similarity between the aspects/opinion words (content information), calculated by traditional approach, and similarity by their respective heterogeneous relationships they have with the opinion words/aspects (link information). Based on the product aspect categories and opinion word groups, a sentiment association set between the two groups is then constructed by identifying the strongest n sentiment links. This approach, however, only considered adjectives as opinion words which are not able to cover every opinion, yet the approach was capable of finding hidden links between product aspects and adjectives. Unfortunately, there were no quantitative experimental results reported, specifically for implicit aspect identification.

A two-phase co-occurrence association rule mining approach to identify implicit aspects is proposed by Hai *et al.* [15]. In the first phase of rule generation, association rules are mined of the from [*opinion word* → *explicit aspect*], from a co-occurrence matrix. Each entry in the co-occurrence matrix represents the frequency degree of a certain opinion-word co-occurring with a certain explicitly mentioned aspect. In the second phase, the rule consequents (i.e., the explicit aspects) are clustered to generate more robust rules for each opinion word. Implicit aspects can then be found by identifying the best cluster for a given sentiment word with no explicit aspect, and assigning the most representative word of that cluster as the implicitly mentioned aspect. This method is reported to yield an F_1-score of 74% on a Chinese mobile phone review data set. However, this frequency-based method requires a very good coverage of opinion words with explicit aspects. It assumes that explicit feature annotations are given and that an implicit feature has to relate to an explicit feature. In this paper, we do not use these assumptions, providing for more generality of the proposed solution.

In [19], a semi-unsupervised method is proposed that can simultaneously extract both sentiment words and product/service aspects from review sentences. The method first extracts appraisal expression patterns (AEPs), which are representations of how people express opinions regarding products or services. The set of AEPs is obtained by selecting frequently occurring shortest dependency paths between two words in a dependency graph. Next the authors propose an AEP latent Dirichlet allocation model for mining the aspect and sentiment

[1] In the restaurant data set [10] that is used for evaluation, around 77% of the aspect categories was not literally mentioned in sentences.

words. The model does, however, assume that all words in a sentence are drawn from one topic. This method is reported to yield at best an F_1-score of 78% on a restaurant review data set.

Association rule mining is also employed in [20], where first the candidate aspect indicators are extracted based on word segmentation, part-of-speech (POS) tagging, and aspect clustering. After that, the co-occurrence degree between these candidate aspect indicators and aspect words are calculated, using five collocation extraction algorithms. These five algorithms use frequency, PMI, frequency*PMI, t test, and χ^2 test, respectively, out of which frequency*PMI is the most promising. Rules are then mined of the form [*aspect indicator* $\rightarrow$ *aspect word*], and only the best rules from the five different rule sets are chosen as the basic rules. The basic set of rules is then extended by mining additional rules from the lower co-occurrence aspect indicators and nonindicator words. The authors propose three methods for doing so: 1) adding dependency rules; 2) adding substring rules; and 3) adding constrained topic model rules. This method is reported to yield at best an F_1-score of 76% on a Chinese mobile phone review data set.

Association rule mining is also the main technique in [21]. Unlike [15] and [20], no annotated explicit aspects are required, instead the double propagation algorithm from [22] is employed to identify the explicit aspects. An advantage of this double propagation method is that it links explicit aspects to opinion words. This is used later, to restrict the set of possible implicit aspects in a sentence to just those that are linked to the opinion words present in that sentence. The notional words are then used to further investigate which of these aspect is most likely the implicit aspect mentioned in this sentence. Their method yielded an F_1-score of 80% on a Chinese mobile phone review data set, and apart from a small seed set of opinion words, it operates completely unsupervised.

The SemEval-2014 competition has given rise to a number of proposed methods to find aspect categories. The first to mention, because of its similarity with the currently proposed approaches, in that it is also co-occurrence-based, is [23], where co-occurrence frequencies are recorded between annotated aspect categories and notional words. This enables the direct association of words with categories. However, this does come at the cost of making the method supervised. Furthermore, its reported performance is one of the lowest in the SemEval-2014 rankings.

A high performing supervised method for category detection is presented in [12]. The authors use a set of binary maximum entropy classifier with bag-of-words and TF-IDF features for each aspect category. With a reported F_1-score of 81% this method was one of the best submitted constrained methods (i.e., no additional training resources were used apart from the official training data).

Another high performing supervised aspect category detection is proposed in [11]. Instead of a MaxEnt classifier, five binary (one-versus-all) SVMs are employed, one for each aspect category. The SVMs use various types of n-grams (e.g., stemmed, character, etc.) and information from a word clustering and a lexicon, both learned from YELP data. The lexicon directly associates aspects with categories. Sentences with no assigned category went through the post-processing step, where the sentence was labeled with the category with maximum posterior probability. The lexicon learned from YELP data significantly improved the F_1-score, which was reported to be 88.6% and ranked first among 21 submissions in SemEval-2014 workshop. However, for fair comparison, the score obtained without using the lexical resources derived from the YELP data, which is an F_1-score of 82.2%, is reported in the evaluation, as our proposed supervised method to which it is compared also does not use external knowledge.

As far as it goes for unsupervised approaches in the SemEval-2014 competition, the one presented in [24] performs best, which reported an F_1-score of 60.0%. Garcia-Pablos *et al.* [24] proposed a basic approach that first detects aspects (another subtask of the SemEval competition), which would then be compared with the category words using the similarity measure described by Wu and Palmer [25]. The category with the highest similarity measure is then selected, if it surpasses a manually set threshold.

III. Unsupervised Method

The proposed unsupervised method (called the spreading activation method) uses co-occurrence association rule mining in a similar way as [15], by learning relevant rules between notional words, defined as the words in the sentence after removing stop words and low frequency words, and the considered categories. This enables the algorithm to imply a category based on the words in a sentence. To avoid having to use the ground truth annotations for this and to keep this method unsupervised, we introduce for each category a set of seed words, consisting of words or terms that describe that category. These words or terms are found by taking the lexicalization of the category, and its synonyms from a semantic lexicon like WordNet. For example, the *ambience* category has the seed set {*ambience, ambiance, atmosphere*}.

With the seed words known, the general idea of implicit aspect detection can be exploited to detect categories as well. The idea is to mine association rules of the form [*notional word* $\rightarrow$ *category*] from a co-occurrence matrix. Each entry in this co-occurrence matrix represents the frequency degree of two notional words co-occurring in the same sentence. Stop words, like *the* and *and*, as well as less frequent words are omitted because they add little value for determining the categories in review sentences.

The reason why we choose to mine for rules similar to that of [15]'s, and do not consider all notional words in the sentence at once to determine the implied categories, like [21], is based on the hypothesis that categories are better captured by single words. If we have for example categories like *food* and *service* all it takes to categorize sentences is to find single words like *chicken*, *staff*, or *helpful*.

Association rules are mined when a strong relation between a notional word and one of the aspect categories exists, with the strength of the relation being modeled using the co-occurrence frequency between category and notional word.

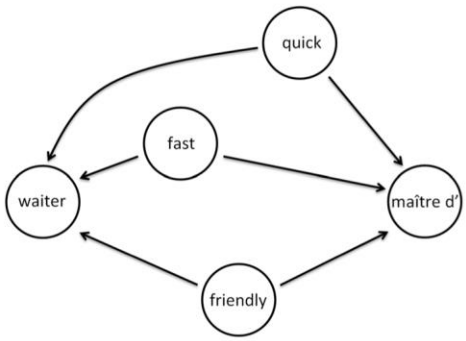


Fig. 1. Example of an indirect relation: "*waiter*" and "*maître d*" are indirectly related by having the same set of directly related notional words.

We distinguish between two different relation types: 1) *direct* and 2) *indirect* relations. A direct relation between two words A and B is modeled as the positive conditional probability $P(B|A)$ that word B is present in a sentence given the fact that word A is present. An indirect relation between two words A and B exists when both A and B have a direct relation with a third word C. This indicates that A and B could be substitutes for each other, even though their semantics might not be the same. Without checking for indirect relations, substitutes are usually not found since they do not co-occur often together. A visual example of an indirect relation can be found in Fig. 1.

To exploit the direct, as well as the indirect relation information between notional words and seed words, the spreading activation algorithm [26] is utilized, which is a method to search for associative networks. Spreading activation has been successfully applied in various fields, e.g., [27] and [28]. For that, a network data structure is needed, consisting of vertices connected by links, as depicted in Fig. 1. The vertices are labeled and the links may receive direction and/or weights to model the relations between vertices. The search process of finding an associative network is initiated by giving each vertex an activation value. These initial values determine the area of the search as the activation values are iteratively spread out to other, linked, vertices.

In our case we want to use spreading activation to find, for each category, a network of words associated with the category's set of seed words. To do this, a network data structure is created, having vertices for all notional words and edges to model the direct relations between these words. In the network data structure all notional words receive an initial activation value of zero except for the category's seed words, which receive positive activation values. In the first iterative step of the spreading activation algorithm, these positive activation values are spread out to other words directly related to the seed words, based on the strength of the direct relation. In this way, words that have strong direct relations with the seed words receive high association values. The following iterative steps will be looking for words with high association values that are then activated and will spread out their activation value to other words directly related to them. In this way, notional words that are indirectly related to one of the seed words are also identified. The end result will be a network of notional words, each with their own activation value, the higher the activation value, the more related the notional word will be to the category.

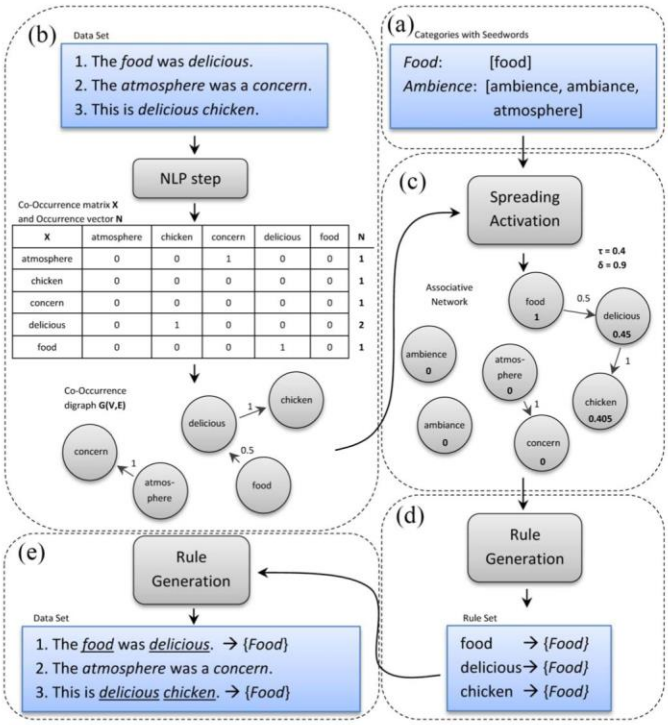


Fig. 2. Example flowchart of the unsupervised method. [the steps are: (a) Identify category seed word sets. (b) Determine co-occurrence digraph. (c) Apply spreading activation. (d) Mine association rules. (e) Assign aspect categories.].

The data network structure used for the spreading activation algorithm will consist of vertices that represent the notional words, and links between two vertices representing a strictly positive co-occurrence frequency. Each link represents the direct relation between two notional words and receives weight equal to the conditional probability that word A co-occurs with word B, given that B appears in a sentence. This also means that the links receive direction as the conditional probability is not symmetric, making the data network structure a co-occurrence digraph.

Once each category has its own associative network, rules can be mined of the form [*notional word* $\rightarrow$ *category*] from vertices in these networks, based on the activation value of the vertex. Since the same word can be present in multiple associative networks, one word might trigger multiple aspect categories. Based on the words in the sentence, a set of rules is triggered and their associated aspect categories are assigned to the sentence. Fig. 2 illustrates how the unsupervised method works on a simple example corpus, with a decay factor of 0.9 and firing threshold of 0.4. The example shows how an associative network for the category food is found and rules are extracted.

A. Algorithm

The method can best be described according to the following steps.

1) Identify Category Seed Word Sets S_c: First, we identify for each of the given categories $c \in C$ a set of seed words S_c

containing the category word and any synonyms of that word. This first step is represented by step (a) in Fig. 2.

2) Determine Co-Occurrence Digraph G(V, E): Next, as a natural language preprocessing step, both training and test data are run through the lemmatizer of the Stanford CoreNLP [29]. We keep track of all lemmas in the text corpus and count their occurrence frequencies. Stop words and lemmas that have an occurrence frequency lower than a small degree α are discarded, while the rest of the lemmas and corresponding frequencies are stored in the occurrence vector N. The parameter α is used to filter out low occurring lemmas. Each lemma in N is now considered to be a notional word. A co-occurrence matrix X is then constructed where each entry represents how often notional word from N_i appeared before N_j the same sentence.

From X and N the co-occurrence digraph $G(V, E)$ is constructed with nodes V and edges E. Each notional word $i \in N$ receives its own node $i \in V$. A directed edge $(i, j) \in E$ between nodes i and j exists if and only if the co-occurrence frequency $X_{i,j}$ is strictly positive. The weight of each edge $(i, j) \in E$ is denoted by $W_{i,j}$ and represents the conditional probability that notional word i co-occurs with notional word j in a sentence after it, given that j is present in that sentence. This formula is shown as follows:

$$W_{i,j} = \frac{X_{i,j}}{N_j} \tag{1}$$

where $X_{i,j}$ is the co-occurrence frequency of words i and j (word i after word j) and N_j is the frequency of word j. Step (b) in Fig. 2 illustrates this step.

3) Apply Spreading Activation: Once the co-occurrence digraph $G(V, E)$ is obtained, we apply for each category $c \in C$ the spreading activation algorithm to obtain for each vertex $i \in V$ an activation value $A_{c,i}$. Each activation value has a range of [0, 1], and the closer it is to 1 the stronger the notional word is associated with the considered category.

The process of obtaining these activation values for category $c \in C$ is initiated by giving all vertices $i \in V$ an activation value $A_{c,i}$. Vertices that are labeled as one of the category's seed words $s \in S_c$ receive the maximum activation value of 1, while the rest of the vertices receive the minimum activation value of 0.

After this initialization step, the iterative process of spreading the activation values starts. The actual spreading of activation values is done by "firing" or "activating" vertices. A vertex that is fired, spreads its activation value to all vertices directly linked to the fired vertex. The activation value added to the linked words depends on the activation value of the fired vertex and the weight of the link between the fired vertex and the vertex receiving the added activation value. The formula for the new activation value for one of the vertices j linked to the fired vertex i is shown as follows:

$$A_{c,j} = \min\{A_{c,j} + A_{c,i} \cdot W_{i,j} \cdot \delta, 1\}. \tag{2}$$

The parameter δ in (2) models the decay of the activation value as it travels further through the network, ranging from 0 to 1. The closer this decay factor gets to 0 the more the firing activation value will have decayed (i.e., it will be closer to 0).

Algorithm 1: Spreading Activation Algorithm

```
input : category c
input : vertices V
input : seed vertices S_c
input : weight matrix W
input : decay factor δ
input : firing threshold τ_c
output: activation values A_{c,i} for category c
foreach s ∈ S_c do
    A_{c,s} ← 1
end
foreach i ∈ V \ S_c do
    A_{c,i} ← 0
end
F ← S_c
M ← S_c
while M ≠ ∅ do
    foreach i ∈ M do
        foreach j ∈ V do
            A_{c,j} ← min{A_{c,j} + A_{c,i} · W_{i,j} · δ, 1}
        end
    end
    M ← ∅
    foreach i ∈ V \ F do
        if A_{c,i} > τ_c then
            add i to F
            add i to M
        end
    end
end
```

Furthermore, any activation value $A_{c,j}$ can have a maximum value 1. Firing vertices is only allowed if its activation value reaches a certain firing threshold τ_c, depending on the category $c \in C$. Once a vertex has been fired it may not fire again. The sets M and F keep track of which vertex may be fired and which vertex has already been fired, respectively.

A single step in the iterative process of spreading the activation values starts by searching for vertices $i \notin F$ with activation value $A_{c,i}$ greater than firing threshold τ_c. These vertices are temporarily stored in M. Then for vertex $i \in M$ we look for vertex j linked to this vertex with edge $(i, j) \in E$, and modify its activation value $A_{c,j}$ according to (2). This is done for each vertex $j \in V$ linked to vertex i with edge $(i, j) \in E$, after which vertex i is removed from M and stored in F, the same process is then executed for the remaining vertices $i \in M$. This concludes one iterative step, that is repeated until no more vertices $i \notin F$ with activation value $A_{c,i}$ greater than firing threshold τ_c exists. The pseudocode for the spreading activation algorithm can be found in Algorithm 1, and an illustration of this complete step can be found in step (c) of Fig. 2.

4) Mine Association Rules: Once spreading activation is applied to all categories $c \in C$, matrix $A_{c,i}$ is obtained, containing, for each notional word $i \in N$, activation values for each category $c \in C$. From these associations values, rules are mined, based on the magnitude of these values. Vertices that

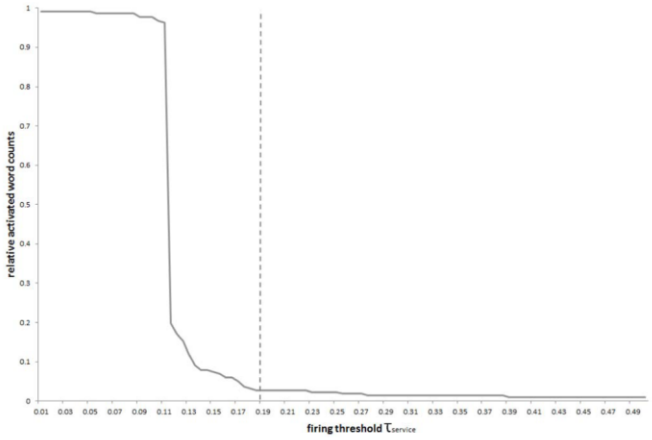


Fig. 3. Graph displaying the relative activated word counts for different values of firing threshold τ_{service} together with the threshold chosen by the heuristic.

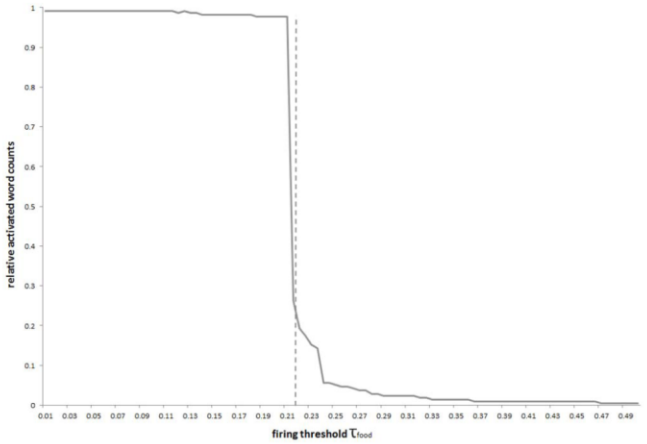


Fig. 4. Graph displaying the relative activated word counts for different values of firing threshold τ_{food} together with the threshold chosen by the heuristic.

have fired are seen as part of the associative network and from each vertex in that network, a rule is mined. Any vertex whose activation value $A_{c,i}$ is higher than parameter τ_c produces a rule [*notional word* $i \rightarrow$ *category* c] that is stored in rule set R. All notional words are allowed to imply multiple categories except for seed words, which can only imply the category they belong to. This step is depicted as step (d) of Fig. 2.

5) Assign Aspect Categories: In the last step we predict categories for each unprocessed sentence, using the rule set R obtained from the previous step. For each unprocessed sentence we use lemmatization, and look if any word matches a rule, after which that rule is applied. Since multiple rules can be fired, it is possible to predict multiple aspect categories per sentence. This last step corresponds to step (e) in Fig. 2.

B. Parameter Setting

Three parameters, α, δ, and τ_c need to be set manually. For α, the minimal occurrence threshold, a value of $0.005 \times$ number of sentences in the data set is used. In this way, low-frequency words are excluded from the co-occurrence matrix. The decay factor δ is set at 0.9 to increase the number of indicators (recall).

The τ_c parameter is set differently for each category c. With parameters α and δ fixed, the algorithm is run for each category using a range of values for τ_c. For each τ_c, the method constructs an association network, counting the number of notional words in it. The decision for the best value for τ_c can be made based on a plot of the activated word count relative to the total number of words in the network. The plots for categories service and food (see Section V for a description of the used data set) are shown in Figs. 3 and 4, respectively.

Fig. 3 shows that having high value for τ_c results in only seed words indicating the presence of a category (i.e., these are the explicitly mentioned categories). This is shown by the long flat tail to the right. On the other hand, having $\tau_c = 0$ results in all words being indicators, producing much noise. To find the optimal, or at least a good, value for τ_c, we use the breakpoint heuristic, where we find the breakpoint in the graph for relative word count, having the flat part of the graph to the right and the sloped part of the graph on the left. This is shown as the dashed vertical line. For most categories this results in a near-optimal choice for τ_c.

One exception is the *food* category, as shown in Fig. 4. Here, we choose to have more words as indicators, because *food* is by far the largest of the aspect categories we aim to detect. Hence, it is reasonable to have a larger associative network, with more words pointing to the *food* category. Given the fact that many different words, such as all kinds of meals and ingredients point to food, it is rather intuitive to have a bigger associate network for this category. Hence, when dealing with a dominant category like *food*, the τ_c should be lower than the one given by the heuristic, for example by setting it similar to Fig. 4.

C. Limitations

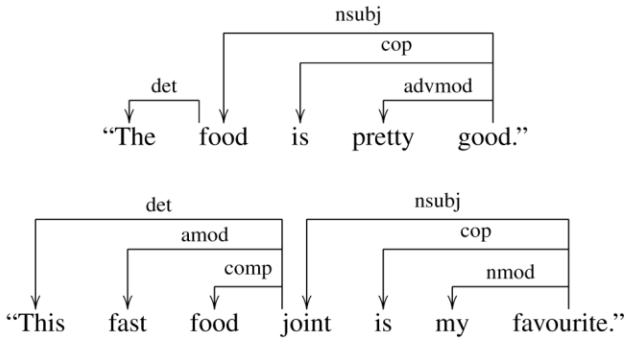
A practical limitation of this unsupervised method is that it requires tuning for multiple parameters. Although one can implement a training regime to learn these parameters, this would render the method supervised, removing one of its key advantages. Another shortcoming, albeit a minor one, is the requirement of determining a seed set up front for each aspect category one wants to find. Using the lexical representation of the category complemented by some synonyms is an easy way of retrieving a suitable seed set words, but abstract or vague categories like "anecdotes/miscellaneous" cannot be dealt with effectively in this way.

IV. Supervised Method

Similar to the first method, the supervised method (called the probabilistic activation method) employs co-occurrence association rule mining to detect categories. We borrow the idea from [23] to count co-occurrence frequencies between lemmas and the annotated categories of a sentence. However, low frequency words are not taken into account in order to prevent overfitting. This is achieved using a parameter α_L, similar to the unsupervised method. Furthermore, stop words are also removed.

In addition to counting the co-occurrences of lemmas and aspect categories, the co-occurrences between grammatical dependencies and aspect categories are also counted. Similar to lemmas, low frequency dependencies are not taken into account to prevent overfitting, using the parameter α_D. Dependencies, describing the grammatical relations between words in a sentence, are more specific than lemmas, as each dependency has three components: 1) governor word; 2) dependent word; and 3) relation type. The added information provided by dependencies, may provide more accurate predictions, when it comes to category detection. Knowing whether a lemma is used in a subject relation or as a modifier can make the difference between predicting and not predicting a category.

To illustrate the value of dependencies, a small example is provided using the following two sentences.



Assuming that the category food exists, and that its category word is a good indicator word for this category, most of the time, the word food will actually indicate the category food, as in the first sentence. However, there are also sentences where the word food does not indicate the category food, as shown in the second sentence. By using the word food as indicator for the category food, both sentences will be annotated with the category food, but by looking at dependencies this does not have to be the case. In the first sentence food is used in relation to "good" as nominal subject, while in the second sentence food is used to modify "joint." From these dependency relations we might learn that only when the word food is used as a nominal subject, it implies the category food.

The fact that dependencies are more specific than lemmas also has a disadvantage. With dependencies being triples, and hence more diverse than lemmas alone, they tend to have a much lower frequency count than single lemmas. This means that many dependencies would not occur frequently enough to be considered, since low frequency dependencies are omitted to mitigate overfitting. To cope with this problem, two variants of each dependency are added: the first is the pair of governor word and dependency type, and the second is the pair of depending word and dependency type. These pairs convey less information than the complete triples, but are still informative compared to having just lemmas. Since the frequency of these pairs is generally higher than that of the triples, more pairs are expected to pass the frequency filter. Hence, we extract, for each dependency, the following three forms: 1) {dependency relation, governor, dependent} (D_1); 2) {dependency relation, dependent} (D_2); and 3) {dependency relation, governor} (D_3).

All the dependencies relations from the Stanford parser [29] are used to build up the dependency forms, except for the determinant relation. For the previous first sentence, this would mean the following dependency sets: [{advmod, good, pretty}, {cop, good, is}, {nsubj, good, food}] (D_1), [{advmod, pretty}, {cop, is}, {nsubj, food}] (D_2), and [{advmod, good}, {cop, good}, {nsubj, good}] (D_3).

The co-occurrence frequencies provide the information needed to find good indicators (i.e., words or dependencies) for the categories. To determine the strength of an indicator, the conditional probability $P(B|A)$ is computed from the co-occurrence frequency, where category B is implied when lemma or dependency form A is found in a sentence. These conditional probabilities are easily computed by dividing the co-occurrence frequency of (B, A) by the occurrence frequency of A. The higher this probability, the more likely it is that A implies B. If this value exceeds a trained threshold, the lemma or dependency form indicates the presence of the corresponding category.

This threshold that the conditional probability has to pass is different for each category. It also depends on whether a dependency form or lemma is involved, since dependency forms generally have a lower frequency, requiring a lower threshold to be effective. Hence, given that there are three dependency forms and one lemma form, four thresholds need to be trained for each category in the training data. To find theses thresholds a simple linear search is performed, picking the best performing (i.e., on the training data) value from a range of values for each different threshold.

Once the conditional probabilities are computed and the thresholds are known, unseen sentences from the test set are processed. For each unseen sentence we check whether any of the lemmas or dependency forms in that sentence have a conditional probability greater than its corresponding threshold, in which case the corresponding category is assigned to that sentence. Fig. 5 illustrates how the supervised method works on a very simple test and training set.

A. Algorithm

The method can best be described according to the following steps.

1) Determine Lemmas/Dependencies: As a natural language preprocessing step, both training and test data are run through the POS tagger, lemmatizer, and dependency parser [30] of the Stanford CoreNLP [29]. This results in all sentences having a set of lemmas, denoted by s_L, and three dependency form sets, denoted by s_{D_1}, s_{D_2}, and s_{D_3}, respectively. The training set provides the annotated categories of each sentence s, which is denoted by s_C.

2) Determine Weight Matrix W: Next all unique categories are identified, storing them in category set C. Additionally, the occurrence frequencies of all lemmas and dependency forms are stored in vector Y, while the co-occurrence frequencies of all dependency form/lemma-category combinations, are counted and stored in matrix X, respectively. These three

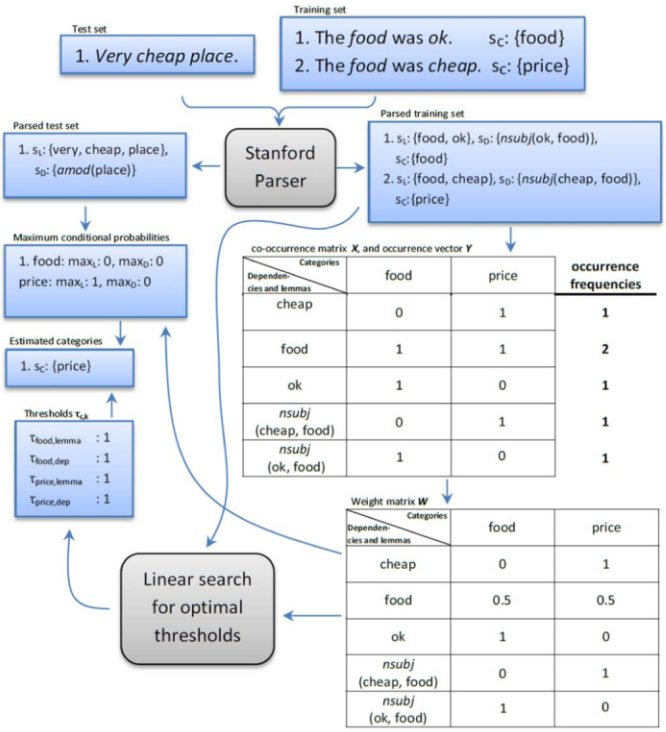


Fig. 5. Example flowchart of the supervised method.

steps of gathering statistical information on the data are all performed on the training data alone.

After the occurrence vector Y and co-occurrence matrix X are obtained, we calculate for each co-occurrence entry $X_{c,j}$, with occurrence frequency Y_j greater than θ, its associated conditional probability $P(c|j)$, and store it in weight matrix W. The threshold θ prevents low occurring lemmas and dependency forms from becoming indicators. This way we aim to mitigate possible overfitting. The value of θ is, based on intuition, set to 4 for these experiments, however, this could be part of the training regime as well. The formula for calculating these conditional probabilities is shown in (3). The pseudo-code for identifying the category set C, counting the occurrence and co-occurrence frequencies, and computing the weight matrix W, is shown in Algorithm 2

$$W_{c,j} = \frac{X_{c,j}}{Y_j}. \tag{3}$$

3) Find Optimal Thresholds $\tau_{c,k}$: Next we execute a linear search for optimal thresholds $\tau_{c,k}$, $c \in C$, $k \in \{L, D_1, D_2, D_2\}$ on the training set. For each category $c \in C$ we optimize the four thresholds $\tau_{c,L}$, τ_{c,D_1}, τ_{c,D_2}, and τ_{c,D_3}. Because the selection of one threshold influences the selection of the other three thresholds, all thresholds are optimized together.

The linear search uses (4) to find the maximum conditional probability $\max_{c,k}$. If the maximum conditional probability $\max_{c,k}$ is higher than the corresponding threshold $\tau_{c,k}$, we predict category c.

The training set is then evaluated for a range of values of thresholds $\tau_{c,k}$, and the thresholds which provided the highest evaluation metric are selected as thresholds for the test set. In our experiments we used as evaluation metric the F_1-score

Algorithm 2: Identify Category Set C and Compute Weight Matrix W

```
input : training set
input : occurrence threshold θ
output: category set C, Weight matrix W
C, X, Y ← ∅
foreach sentence s ∈ Training set do
    // s_k are the lemmas/dependecies of s
    foreach s_k ∈ {s_L, s_D1, s_D2, s_D3} do
        foreach dependency forms/lemmas j ∈ s_k do
            // count dependency form/lemma
               occurrence j in Y
            if j ∉ Y then
                add j to Y
            end
            Y_j ← Y_j + 1
            // s_C are the categories of s
            foreach category c ∈ s_C do
                // Add unique categories in
                   category set C
                if c ∉ C then
                    add c to C
                end
                // count co-occurrence (c, j)
                   in X
                if (c, j) ∉ X then
                    add (c, j) to X
                end
                X_{c,j} ← X_{c,j} + 1
            end
        end
    end
end
// Compute conditional probabilities
foreach (c, j) ∈ X do
    if Y_j > θ then
        W_{c,j} ← X_{c,j}/Y_j
    end
end
```

and as range [0.5, 1) with a step of 0.01 for thresholds $\tau_{c,k}$

$$\max_{c,k} = \max_{j \in s_k} W_{c,j}. \tag{4}$$

4) Estimate Categories: The final step is to predict the aspect categories for each unseen sentence $s \in$ test set. From all lemmas and dependency forms s_L, s_{D_1}, s_{D_2}, and s_{D_3} in sentence s we find the maximum conditional probability $P(c|j)$, as described in (4), for each category $c \in C$. Then, if any of these maximum conditional probabilities surpasses their threshold $\tau_{c,k}$, category c is assigned as an aspect category for sentence s. The pseudo-code for this step is shown in Algorithm 3.

B. Limitations

The main disadvantage of this method is that, contrary to unsupervised methods, this method requires a sufficient

Algorithm 3: Estimating Categories for the Test Set

input : training set
input : test set
input : occurrence threshold θ
output: Estimated categories for each sentence in the test set

$W, C \leftarrow$ `Algorithm 2(Training set,` θ`)`
$\tau_{c,L}, \tau_{c,D_1}, \tau_{c,D_2}, \tau_{c,D_3} \leftarrow$ `LinearSearch (Training set,` W, C`)`
`// Processing of review sentences`
foreach *sentence* $s \in$ *test set* **do**
  foreach *category* $c \in C$ **do**
    `// Obtain maximum conditional probabilities` $P(c|j) = W_{c,j}$ `per type, for sentence` s
    $\max_{c,L} \leftarrow \max_{l \in s_L} W_{c,l}$
    $\max_{c,D_1} \leftarrow \max_{d_1 \in s_{D_1}} W_{c,d_1}$
    $\max_{c,D_2} \leftarrow \max_{d_2 \in s_{D_2}} W_{c,d_2}$
    $\max_{c,D_3} \leftarrow \max_{d_3 \in s_{D_3}} W_{c,d_3}$
    if $max_{c,L} > \tau_{c,L}$ **or** $max_{c,D_1} > \tau_{c,D_1}$ **or** $max_{c,D_2} > \tau_{c,D_2}$ **or** $max_{c,D_3} > \tau_{c,D_3}$ **then**
      estimate category c for sentence s
    end
  end
end

amount of annotated data in order to work properly. For a small annotated data set this method will be inaccurate. Especially the dependency indicators require enough training data in order to be effectively used to predict categories.

Another limitation stems from the use of dependency relations. These are found by using a syntactical parser, which relies on the grammatical correctness of the sentence. However, the grammar used in review sentences can be quite disappointing. If sentences have weird grammatical structures, the parser will not be able to extract relevant dependency relations from these sentences, and may even misrepresent certain dependencies.

Furthermore, because dependencies are triplets, and many different dependency relations exist, the number of different dependency triplets is huge, which makes it harder to find rules that generalize well to unseen data. While a sufficiently large training set will negate this issue, this might unfortunately not always be available.

V. Evaluation

For the evaluation of the proposed methods, the training and test data from SemEval-2014 [10] are used. It contains 3000 training sentences and 800 test sentences taken from restaurant reviews. Each sentence has one or more annotated aspect categories. Fig. 6 shows that each sentence has at least one category and that approximately 20% of the sentences have multiple categories. With 20% of the sentences having multiple categories, a method would benefit from being able to predict multiple categories. This is one of the reasons why association rule mining is useful in this scenario as multiple rules can apply to a single sentence.

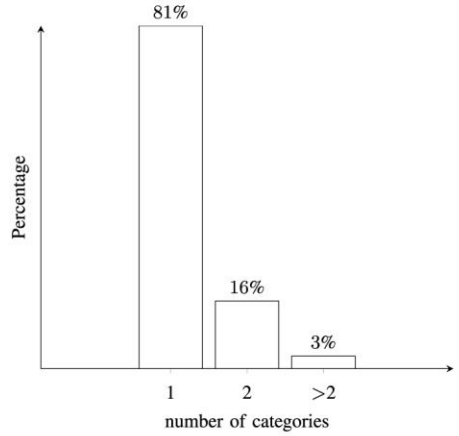


Fig. 6. Distribution of number of aspect categories per sentence.

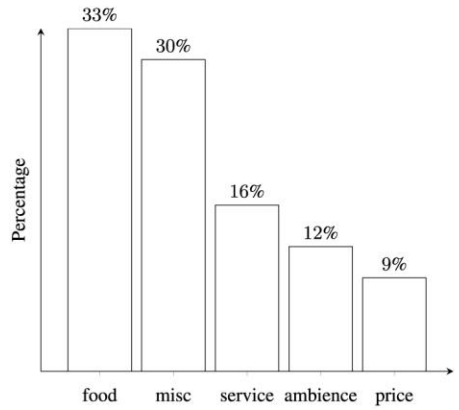


Fig. 7. Relative frequency of the aspect categories.

Fig. 7 presents the relative frequency of each aspect category, showing that the two largest categories, food and anecdotes/miscellaneous, are found in more than 60% of the sentences. This should make these categories easier to predict than the other categories, not only because of the increased chance these categories appear, but also because there is more information about them.

Last, in Fig. 8, the proportion of implicit and explicit aspect categories is shown. It is clear that using techniques related to implicit aspect detection is appropriate here, given that more than three quarters of the aspect categories is not literally mentioned in the text.

Because both unsupervised and supervised method work best for well-defined aspect categories, the last category in this data set, anecdotes/miscellaneous poses a challenge. It is unclear what exactly belongs in this category, and its concept is rather abstract. For that reason, we have chosen not to assign this category using any of the actual algorithms, but instead, this category is assigned when no other category is assigned by the algorithm. The characteristics in Fig. 6 also show that the use of anecdotes/miscellaneous as a "fallback" is justified given its large size and the fact that every sentence has at least one category.

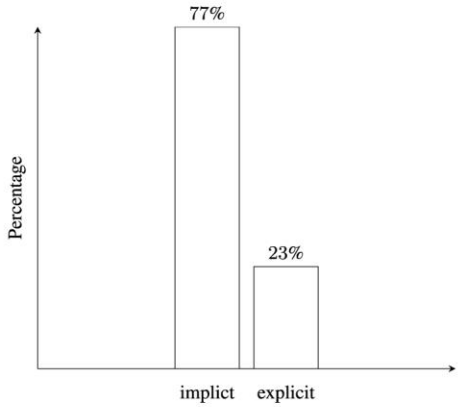


Fig. 8. Ratio between implicit aspect categories and explicitly mentioned ones.

TABLE I
CHOSEN FIRING THRESHOLDS AND THEIR EVALUATION SCORES ON THE TEST SET

Category	TP's	FP's	FN's	τ_c	precision	recall	F_1
food	313	103	105	0.22	75.1%	74.4%	74.8%
service	100	4	72	0.19	96.2%	58.1%	72.5%
ambience	41	10	77	0.09	80.4%	34.8%	48.5%
price	52	16	31	0.09	79.0%	54.2%	64.3%
misc.	163	159	71	-	50.6%	70.9%	59.1%
all	852	157	173	-	70.0%	64.7%	67.0%

As is done at the SemEval-2014 [10] competition, the methods are evaluated based on the micro averaged F_1-score defined as follows:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \tag{5}$$

where precision (P) and recall (R) are defined as follows:

$$P = \frac{\text{TP}}{\text{FP} + \text{TP}}, R = \frac{\text{TP}}{\text{FN} + \text{TP}} \tag{6}$$

where TP, FP, and FN represent the true positives, false positives, and false negatives, respectively, of the estimated aspect categories with respect to the (gold) aspect category annotations.

A. Unsupervised Method

Table I displays, for each aspect category, the chosen firing threshold together with the resulting precision, recall, and F_1-score on the test set. The category *anecdotes/miscellaneous* is estimated when none of the other four categories are chosen in the sentence.

With an overall F_1-score of 67.0% on the test set, the method seems to perform well, but the performance strongly depends on the choice of the parameters. If we would for example, had chosen to treat the category service as a dominant category, like we did with the category food, and had lowered the firing threshold, then the precision of this category would have dropped significantly, while the recall would only increase slightly. Likewise, if we would not have treated the category food as a dominant category, its recall would have severely dropped, while the precision would have increased. So certain domain knowledge about the data set is required when choosing parameter values. Table II shows this sensitivity of the firing thresholds, where the relative change in terms of F_1-score is given when deviating from the chosen thresholds. As can be seen the proposed method is sensitive to threshold variations.

TABLE II
RELATIVE CHANGE IN F_1, WHEN VARYING FIRING THRESHOLDS

Category	-0.05	-0.02	-0.01	0	0.01	0.02	0.05
food	-8	-7.9	-7.9	0	-1.9	-6.7	-25
service	-8.6	-3.3	-4.6	0	0	0	0
ambience	-47	3.1	8.9	0	0	0	-5.6
price	-72.1	-18.7	-11	0	0	0.1	1.6

TABLE III
EVALUATION SCORES OF THE SUPERVISED METHOD WITH BOTH DEPENDENCY AND LEMMA INDICATORS ON THE TEST SET

Category	TP's	FP's	FN's	precision	recall	F_1
food	371	51	47	87.9%	88.8%	88.3%
service	159	32	13	83.2%	92.4%	87.6%
ambience	83	28	35	73.8%	70.3%	72.5%
price	74	8	9	90.2%	89.2%	89.7%
anecdotes/misc.	165	38	69	81.3%	70.5%	75.5%
all	852	157	173	84.4%	83.1%	83.8%

TABLE IV
EVALUATION SCORES OF THE SUPERVISED METHOD WITH ONLY LEMMA INDICATORS ON THE TEST SET

Category	TP's	FP's	FN's	precision	recall	F_1
food	348	35	70	90.9%	83.3%	86.9%
service	153	13	19	92.2%	89.0%	90.5%
ambience	78	28	40	73.6%	66.1%	69.6%
price	79	9	4	89.9%	95.2%	92.4%
anecdotes/misc.	165	38	69	81.3%	70.5%	75.5%
all	823	123	202	87.5%	80.3%	83.5%

From Table I, one can conclude that this approach has difficulty predicting the category *ambience*. This might be due to the nature of that particular category, as it is often not specified in a sentence by just one word, but is usually derived from a sentence by looking at the sentence as a whole. This can be illustrated with the following example.

> "Secondly, on this night the place was overwhelmed by upper east side ladies perfume"

where there is no particular word that strongly suggests that "ambience" is the right category for this sentence.

B. Supervised Method

For the supervised method we use the training set to learn the parameters and co-occurrence frequencies, after which we evaluate the method on the test set. To see the impact the dependency indicators have, this method is executed separately for the dependency indicators, lemma indicators and a combined version where both lemma and dependency indicators are used, and evaluated on the test set. Tables III–V show the results.

Comparing the results from Tables III–V shows that using both dependency and lemma indicators provides best results. However, these results are only slightly better than when only lemma indicators are used, which means that we cannot claim that dependency indicators are beneficial when predicting categories in terms of F_1 score. Table V does show that by

TABLE V
EVALUATION SCORES OF THE SUPERVISED METHOD WITH ONLY DEPENDENCY INDICATORS ON THE TEST SET

Category	TP's	FP's	FN's	precision	recall	F_1
food	343	45	75	88.4%	82.1%	85.1%
service	152	27	20	84.9%	88.4%	86.6%
ambience	62	34	56	64.6%	52.5%	57.9%
price	61	5	22	92.4%	73.5%	81.9%
anecdotes/misc.	165	38	69	81.3%	70.5%	75.5%
all	783	149	242	84.0%	76.4%	80.0%

themselves, dependency indicators do have predicting power, albeit less than lemma indicators. This was as expected, since dependency indicators consists of more than one component, which makes it harder to find rules that generalize well to unseen data, and, in addition, they also rely on the grammatical correctness of the sentence.

Using dependency indicators, in addition to lemma indicators, does seem to result into finding more categories, even though it is less precise in doing so. This is especially the case for the category food. The main reason for this is that food is by far the largest category, resulting in more available training data for this category, which makes it easier to find rules.

C. Comparison

To evaluate the quantitative performance of the proposed method, it is compared against several baseline methods and three successful methods from the SemEval-2014 competition. The four baseline methods are as follows.

1) *Seed Word Baseline:* This baseline estimates a category if one of its seed words is present in the sentence.
2) *Majority Baseline:* This baseline predicts for every sentence the two most common categories (i.e., *food* and *anecdotes/miscellaneous*) present in the data.
3) *Random Baseline:* For this baseline we randomly select categories for each sentence. The chance of selecting a certain category depends on the appearance in the training set, just as the number of selected categories depends on the distribution of the number of categories per sentence in the training set.
4) *SemEval Baseline:* The final baseline comes from [10], and it is a simple supervised method. For every test sentence s, the k most similar to s training sentences are retrieved. Here the similarity between two sentences is measured by calculating the Dice coefficient of the sets of distinct words of two sentences. Then, s is assigned the m most frequent aspect categories of the k retrieved sentences. This baseline is clearly a supervised method and thus requires a training set.

The four methods from the literature are V3 [24], an unsupervised semantic similarity algorithm, Schouten *et al.* [23], a supervised co-occurrence-based algorithm, Brychcin *et al.* [12], the best performing submitted constrained (i.e., no external training data is used) method, and a constrained version (see end of Section II) of Kiritchenko *et al.* [11], the best constrained supervised machine learning approach at this particular task at SemEval-2014. The resulting overall precision, recall, and F_1-score are displayed in Table VI.

TABLE VI
F_1-SCORES OF DIFFERENT (CONSTRAINED) METHODS

Method	precision	recall	F_1
Random baseline	30.8%	30.5%	30.6%
Majority baseline	38.8%	63.7%	48.2%
Seed word baseline	57.2%	46.4%	51.2%
Schouten et al. [23]	63.3%	55.8%	59.3%
V3 [24]	63.3%	56.9%	60.2%
SemEval-2014 baseline [10]	-	-	63.9%
Proposed Unsupervised Method	69.5%	64.7%	67.0%
UWB [12]	85.1%	77.4%	81.0%
constrained Kiritchenko et al. [11]	86.5%	78.3%	82.2%
Proposed Supervised Method	84.4%	83.1%	83.8%

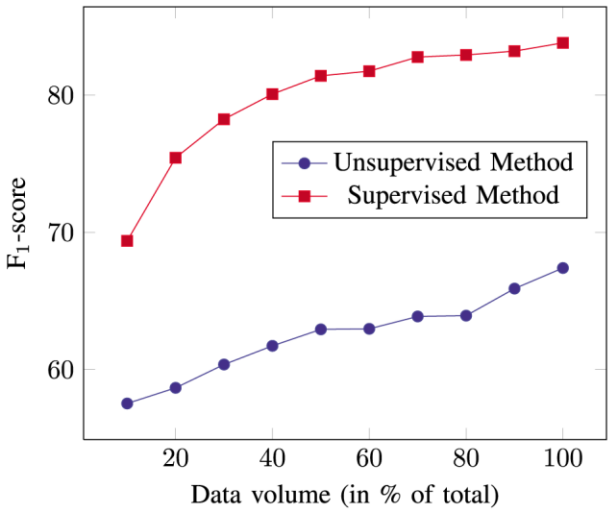


Fig. 9. F_1-scores for different sizes of the training set (% of 3000 sentence).

Clearly, the supervised method, as well as many other (supervised) methods presented at SemEval-2014 perform better than the proposed unsupervised method. However, this is to be expected for an unsupervised method. Interestingly, it is able to outperform the basic bag-of-words supervised approach of the SemEval-2014 baseline, as well as the supervised co-occurrence-based method from [23]. On the other hand, the second proposed method beats all constrained methods from the SemEval-2014 competition. Note however, that the full (unconstrained) method from Kiritchenko *et al.* [11] outperforms our proposed method by a few percent, which is due to the fact that it enjoys an unconstrained training regime.

In Fig. 9, F_1-scores are shown for different sizes of the training set, using a stratified sampling technique where the distribution of the categories remains similar to the original data set. Each data point in the figure represents an incremental increase of 10% (300 sentences) in labeled data, for the supervised method, and unlabeled data for the unsupervised method. The supervised method always seems to outperform the unsupervised method, although larger training sizes for the unsupervised method seem to perform on par with the supervised method for which very small amounts of labeled data are available (F_1-score around 70%).

VI. CONCLUSION

In this paper we have presented two methods for detecting aspect categories, that is useful for online review

summarization. The first, unsupervised, method, uses spreading activation over a graph built from word co-occurrence data, enabling the use of both direct and indirect relations between words. This results in every word having an activation value for each category that represents how likely it is to imply that category. While other approaches need labeled training data to operate, this method works unsupervised. The major drawback of this method is that a few parameters need to be set beforehand, and especially the category firing thresholds (i.e., τ_c) need to be carefully set to gain a good performance. We have given heuristics on how these parameters can be set.

The second, supervised, method uses a rather straightforward co-occurrence method where the co-occurrence frequency between annotated aspect categories and both lemmas and dependencies is used to calculate conditional probabilities. If the maximum conditional probability is higher than the associated, trained, threshold, the category is assigned to that sentence. Evaluating this approach on the official SemEval-2014 test set [10], shows a high F_1-score of 83%.

In terms of future work, we would like to investigate how injecting external knowledge would improve the results. While lexicons are a good way of doing that, as shown by Kiritchenko *et al.* [11], we are especially interested in exploiting more semantic alternatives, like ontologies or other semantic networks. Also, as we are dealing with unbalanced data, we plan to explore machine learning techniques that address this problem [31].

REFERENCES

[1] P. F. Bone, "Word-of-mouth effects on short-term and long-term product judgments," *J. Bus. Res.*, vol. 32, no. 3, pp. 213–223, 1995.

[2] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[3] S. Sen and D. Lerman, "Why are you telling me this? An examination into negative consumer reviews on the Web," *J. Interact. Marketing*, vol. 21, no. 4, pp. 76–94, 2007.

[4] B. Bickart and R. M. Shindler, "Internet forums as influential sources of consumer information," *J. Consum. Res.*, vol. 15, no. 3, pp. 31–40, 2001.

[5] D. Smith, S. Menon, and K. Sivakumar, "Online peer and editorial recommendations, trust, and choice in virtual markets," *J. Interact. Marketing*, vol. 19, no. 3, pp. 15–37, 2005.

[6] M. Trusov, R. E. Bucklin, and K. Pauwels, "Effects of word-of-mouth versus traditional marketing: Findings from an Internet social networking site," *J. Marketing*, vol. 73, no. 5, pp. 90–102, 2009.

[7] M. T. Adjei, S. M. Noble, and C. H. Noble, "The influence of C2C communications in online brand communities on customer purchase behavior," *J. Acad. Marketing Sci.*, vol. 38, no. 5, pp. 634–653, 2010.

[8] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retrieval*, vol. 2, nos. 1–2, pp. 1–135, 2008.

[9] C.-L. Liu, W.-H. Hsaio, C.-H. Lee, G.-C. Lu, and E. Jou, "Movie rating and review summarization in mobile environment," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 3, pp. 397–407, May 2012.

[10] M. Pontiki *et al.*, "SemEval-2014 Task 4: Aspect based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, Dublin, Ireland, 2014, pp. 27–35.

[11] S. Kiritchenko, X. Zhu, C. Cherry, and S. M. Mohammad, "NRC-Cananda-2014: Detecting aspects and sentiment in customer reviews," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, Dublin, Ireland, 2014, pp. 437–442.

[12] T. Brychcin, M. Konkol, and J. Steinberger, "UWB: Machine learning approach to aspect-based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, Dublin, Ireland, 2014, pp. 817–822.

[13] C. R. C. Brun, D. N. Popa, and C. Roux, "XRCE: Hybrid classification for aspect-based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, Dublin, Ireland, 2014, pp. 838–842.

[14] G. Castellucci, S. Filice, D. Croce, and R. Basili, "UNITOR: Aspect based sentiment analysis with structured learning," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, Dublin, Ireland, 2014, pp. 761–767.

[15] Z. Hai, K. Chang, and J.-J. Kim, "Implicit feature identification via co-occurrence association rule mining," in *Proc. 12th Int. Conf. Comput. Linguist. Intell. Text Process. (CICLing)*, Tokyo, Japan, 2011, pp. 393–404.

[16] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 813–830, Mar. 2016.

[17] Q. Su, K. Xiang, H. Wang, B. Sun, and S. Yu, "Using pointwise mutual information to identify implicit features in customer reviews," in *Computer Processing of Oriental Languages. Beyond the Orient: The Research Challenges Ahead* (LNCS 4285), Y. Matsumoto, R. Sproat, K.-F. Wong, and M. Zhang, Eds. Berlin, Germany: Springer, 2006, pp. 22–30.

[18] Q. Su *et al.*, "Hidden sentiment association in Chinese Web opinion mining," in *Proc. 17th Conf. World Wide Web (WWW)*, Beijing, China, 2008, pp. 959–968.

[19] X. Zheng, Z. Lin, X. Wang, K.-J. Lin, and M. Song, "Incorporating appraisal expression patterns into topic modeling for aspect and sentiment word identification," *Knowl. Based Syst.*, vol. 61, no. 1, pp. 29–47, 2014.

[20] W. Wang, H. Xu, and W. Wan, "Implicit feature identification via hybrid association rule mining," *Expert Syst. Appl. Int. J.*, vol. 40, no. 9, pp. 3518–3531, 2013.

[21] Y. Zhang and W. Zhu, "Extracting implicit features in Online customer reviews for opinion mining," in *Proc. 22nd Int. Conf. World Wide Web Companion (WWW Companion)*, 2013, pp. 103–104.

[22] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Comput. Linguist.*, vol. 37, no. 1, pp. 9–27, 2011.

[23] K. Schouten, F. Frasincar, and F. de Jong, "COMMIT-P1WP3: A co-occurrence based approach to aspect-level sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, Dublin, Ireland, 2014, pp. 203–207.

[24] A. Garcia-Pablos, M. Cuadros, S. Gaines, and G. Rigau, "V3: Unsupervised generation of domain aspect terms for aspect based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, Dublin, Ireland, 2014, pp. 833–837.

[25] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proc. 32nd Annu. Meeting Assoc. Comput. Linquistics*, Las Cruces, NM, USA, 1994, pp. 133–138.

[26] F. Crestani, "Application of spreading activation techniques in information retrieval," *Artif. Intell. Rev.*, vol. 11, no. 6, pp. 453–482, 1997.

[27] S. Bagchi, G. Biswas, and K. Kawamura, "Task planning under uncertainty using a spreading activation network," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 6, pp. 639–650, Nov. 2000.

[28] A. Katifori, C. Vassilakis, and A. Dix, "Ontologies and the brain: Using spreading activation through ontologies to support personal interaction," *Cognitive Syst. Res.*, vol. 11, no. 1, pp. 25–41, 2010.

[29] C. D. Manning *et al.*, "The Stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguist. Syst. Demonstrations*, 2014, pp. 55–60. [Online]. Available: http://www.aclweb.org/anthology/P/P14/P14-5010

[30] M.-C. de Marneffe and C. D. Manning, "Stanford typed dependencies manual," Stanford NLP Group, Stanford University, Stanford, CA, USA, Tech. Rep., Sep. 2008. [Online]. Available: https://nlp.stanford.edu/software/dependencies_manual.pdf

[31] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 1, pp. 281–288, Feb. 2009.

Kim Schouten is currently pursuing the Ph.D. degree with Erasmus University Rotterdam, The Netherlands, focusing on aspect-level sentiment analysis and its application, implicitness of aspects and sentiment, and how to move toward a more semantics-oriented form of sentiment analysis.

His current research interests include the application of language technology within an economic framework, and language in relation to artificial intelligence.

Onne van der Weijde received the M.Sc. degree in econometrics and management science from Erasmus University Rotterdam, The Netherlands.

He is currently a Data Scientist at OneUp.Company, an innovative start-up in The Netherlands, focusing on semantic analysis, machine learning, and business applications.

Flavius Frasincar received the Ph.D. degree in information systems from Eindhoven University of Technology, The Netherlands.

He is an Assistant Professor of Information Systems with Erasmus University Rotterdam, The Netherlands. He has published in numerous conferences and journals in the areas of databases, Web information systems, personalization, machine learning, and the semantic Web.

Mr. Frasincar is an Editorial Board Member of the *International Journal of Web Engineering and Technology* and *Decision Support Systems*.

Rommert Dekker received the Ph.D. degree in operations research from Leiden University, The Netherlands.

He is a Professor of Operations Research, Quantitative Logistics, and IT with Erasmus University Rotterdam, The Netherlands. He currently leads an industry-sponsored research program on service logistics. His current research interests include logistics, inventory control, maintenance optimization, transport optimization, and information technology, such as sentiment analysis, that can be applied within the previously mentioned business areas.