

Ontology-Driven Sentiment Analysis of Product and Service Aspects

Kim Schouten, Flavius Frasincar

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

Abstract. With so much opinionated, but unstructured, data available on the Web, sentiment analysis has become popular with both companies and researchers. Aspect-based sentiment analysis goes one step further by relating the expressed sentiment in a text to the topic, or aspect, the sentiment is expressed on. This enables a detailed analysis of the sentiment expressed in, for example, reviews of products or services. In this paper we propose a knowledge-driven approach to aspect sentiment analysis that complements traditional machine learning methods. By utilizing common domain knowledge, as encoded in an ontology, we improve the sentiment analysis of a given aspect. The domain knowledge is used to determine which words are expressing sentiment on the given aspect as well as to disambiguate sentiment carrying words or phrases. The proposed method has a highly competitive performance of over 80% accuracy on both SemEval-2015 and SemEval-2016 data, significantly outperforming the considered baselines.

1 Introduction

With so much opinionated, but unstructured, data available on the Web, sentiment analysis has become popular with both companies and researchers. Its goal is to extract the sentiment of content creators, such as the writers of consumer reviews, and to aggregate this information into easy to digest overviews, infographics, or dashboards. Depending on the specific scenario, sentiment can be modeled as a set of emotions, or, more commonly, as a point on a polarity scale ranging from positive to negative. Polarity can be binary with just the positive and negative value, or it can be modeled as a 5-star score, or even as a real number within a given interval (e.g., between -1.0 and 1.0).

Since reviews often go into detail about certain characteristics of the entity under review, it is useful to go one step further and perform aspect-based sentiment analysis. Here, instead of computing a sentiment score for the whole review, or even per sentence, the goal is to locate the different characteristics, or aspects, the reviewer writes about, and then compute a sentiment score for each of the mentioned aspects. This yields more in-depth results, as people are often not positive (or negative) about every aspect of the product or service they bought.

In general, aspect-based sentiment analysis methods can be classified as knowledge-based or as machine learning based [13]. This is of course not a perfect classification, as machine learning methods often incorporate information from dictionaries, such as sentiment lexicons. Nevertheless, it is a useful distinction as machine learning methods require a sufficient amount of training data to perform well, while knowledge-based methods do not. In the pursuit of high performance, machine learning classifiers have become very popular at the expense of knowledge-based systems. In this paper we hypothesize that both have their use and that the two methods are in fact complementary. Using both statistical learning and rules with a knowledge repository is thus hypothesized to work best. To that end, we have designed an ontology in the restaurant domain with rules to decide what sentiment to assign in which situation, as well as a bag-of-words model, with additional features, such as a sentiment value of the sentence, based on a Support Vector Machine classifier. With the focus being solely on the sentiment analysis of aspects, the aspect detection phase is not considered in this paper, and hence, the aspect annotations in the data are used as a starting point.

In the next section, some related work is discussed, followed by the problem definition and overview of the used data sets in Sect. 3. In Sect. 4, the employed domain ontology is explained, as well as the rules to predict a sentiment value for a given aspect. It also contains a short overview of the used bag-of-words model. The base models and the hybrid combinations are evaluated in Sect. 5, and in Sect. 6, conclusions are given and directions for future research are provided. The developed ontology and source code can be found at <https://github.com/KSchouten/Heracles>.

2 Related Work

A short overview of the field of affective computing, which encompasses sentiment analysis, is presented in [3]. The author argues that hybrid methods, combining the intuitive nature and explanatory power of knowledge-driven approaches and the high performance of statistical methods, are the most promising way to improve the effectiveness of affective algorithms. This forms the research hypothesis of this work as well, as we combine both approaches in a way that is similar to [1]. In that work, statistical methods are combined with a set of linguistic patterns based on SenticNet [2]. Each sentence is processed in order to find the concepts expressed in it. The discovered concepts are linked to the SenticNet knowledge repository, which enables the inference of the sentiment value associated to the sentence. If there are no concepts expressed in this sentence or if the found concepts are not in the knowledge base, then a deep learning, bag-of-words method is employed to determine the sentiment for that sentence. Note that this is a sentence-level approach and not an aspect-based approach, like we consider here. Our work has a similar setup in that it first tries to use the knowledge-driven approach to make a prediction, using the statistical method as a backup when the knowledge-base is insufficient.

A multi-domain approach to sentence-level sentiment analysis is presented in [4]. While sentiment is assigned to sentences instead of aspects, the sentences can come from different domains, so the proposed method needs to disambiguate sentiment words based on the domain the sentence is from. This is similar to our approach where sentiment words are disambiguated based on the aspect they are about. Differently from [4], our ontology does not feature a strict separation of semantic information and sentiment information. Furthermore, [4] uses fuzzy membership functions to describe the relations between concepts, sentiment, and domains, and while this gives more modeling flexibility, it makes it harder to reason over the knowledge graph, which is one of the things we want to explore in this work. Other work that uses fuzzy ontologies includes [8], where an ontology is used to aid in aspect-based sentiment analysis. However, the used ontology is automatically generated and only captures a concept taxonomy, missing out on the more advanced options such as using axioms for context-dependent sentiment words.

In [16], a method is presented that predicts the sentiment value for sentiment-bearing words based on the context they are in. For this task, a Bayesian model is created that uses the words surrounding a sentiment-bearing word, including the words that denote the aspect, to predict the actual sentiment value of the word given the context. Similar to our approach, it uses a two-stage setup, where a backup method is used when the first method cannot make a decision. In this case, if the Bayesian model cannot make a decision about the sentiment value of the word, the previous opinion in the text is checked and if there is a conjunction between the two (i.e., no contrasting or negation words), it will assign the same sentiment value to the current word.

The methods presented in this work improve on our previous approach for ontology-enhanced sentiment analysis, presented in [14], in two major ways. First, the ontology is designed more effectively, being able to support both aspect detection and sentiment analysis better, although this work only focuses on sentiment analysis. This is achieved by clearly distinguishing between three types of sentiment words: generic sentiment words that always have the same sentiment value regardless of the context, aspect-specific sentiment words that infer the presence of a single aspect and are only applicable to that aspect (e.g., “rude” for the service aspect), and context-dependent sentiment words that are applicable to more than one aspect, but not necessarily all of them, and that may have different sentiment values for different aspects (e.g., “small” being generally negative for portions, but usually positive for price). Our previous work, while designating the generic sentiment words as such, does not distinguish between the second and third type of sentiment words, which leads to mistakes.

Last, our previous approach utilized the ontology-derived information in the form of additional input features for the Support Vector Machine (SVM) model, while in the current work we use a two-stage approach. In the primary stage, the ontology is used to find and infer sentiment for the current aspect, and if successful, that becomes the prediction of the method. Only when the ontology either finds both positive and negative signals, or none at all, we employ an SVM

model to predict the sentiment. This secondary, or backup, model is a slightly improved bag-of-words model that does not use ontology features. For improved comparison, the performance comparison in Sect. 5 includes an SVM model with additional ontology features, similar to [14].

3 Specification of Data and Tasks

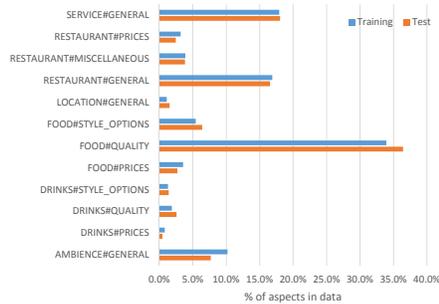
For this research, the widely used set of restaurant reviews from SemEval-2015 Task 12 [12] and SemEval-2016 Task 5 Subtask 1 [11] is employed. The SemEval-2016 data contains the SemEval-2015 data and consists of a training set of 350 reviews with in total 2506 sentiment-labeled aspects and a test set of 90 reviews with in total 859 sentiment-labeled aspects. Given that the SemEval-2015 data is a subset of SemEval-2016, it has similar properties, which are therefore not discussed separately.

An excerpt of the raw data is given in Fig. 1. The provided annotations already split the dataset into reviews and sentences, and each sentence can be labeled with zero or more opinions, which is an aspect together with the expressed sentiment related to that aspect.

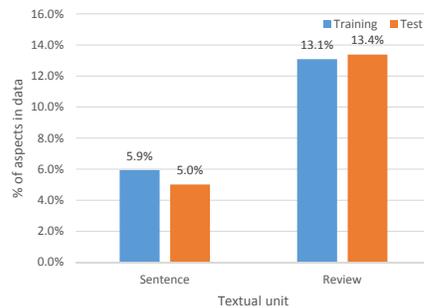
```
<sentence id="1032695:1">
<text>Everything is always cooked to perfection, the
    service is excellent, the decor cool and understated.</
text>
<Opinions>
<Opinion target="NULL" category="FOOD#QUALITY" polarity="
    positive" from="0" to="0"/>
<Opinion target="service" category="SERVICE#GENERAL"
    polarity="positive" from="47" to="54"/>
<Opinion target="decor" category="AMBIENCE#GENERAL"
    polarity="positive" from="73" to="78"/>
</Opinions>
</sentence>
```

Fig. 1: A snippet from the used dataset showing an annotated sentence from a restaurant review.

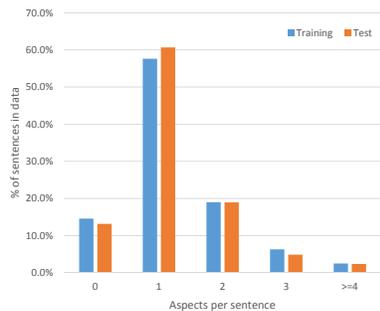
Some aspects are explicit, which means that there is a specific text segment that expresses that aspect, called the target expression, while others are implicit meaning that there is no such target expression. The target expression, if available, is part of the provided annotations. Some statistics related to aspects and sentiment can be found in Fig. 2. In Fig. 2a, the number of times each category label appears is presented and in Fig. 2b, the proportion of aspects that have a sentiment value that is different from the majority within the same textual unit is shown. This gives the minimum error rate for a sentence-level or review-level sentiment analysis system, respectively, as these systems are not able to assign different sentiment values to aspects within the same textual unit. Fig. 2c shows



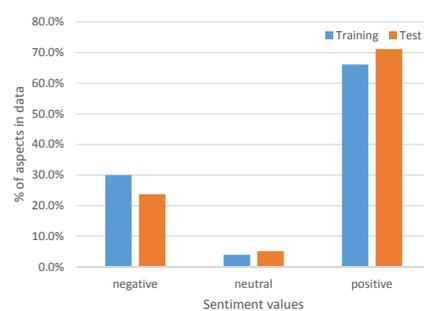
(a) Relative frequencies of each aspect category label



(b) The proportion of aspects that have a polarity that is different from the majority of the aspects within the same textual unit (sentence or review)



(c) The distribution of aspects per sentence



(d) Relative frequencies of each sentiment label

Fig. 2: Some statistics related to the used data set

that while most have just one aspect, a significant number of sentences contain more than one aspect. This complicates the sentiment analysis as it is not always clear to which aspect a certain sentiment expression pertains. Fig. 2d presents the distribution of sentiment values over aspects, showing that this data set is unbalanced with respect to sentiment.

The task of aspect sentiment classification is to give the sentiment value for each aspect, where the aspects are already provided. Thus, all annotations, like the ones given in Fig. 1, are provided, except the values of the `polarity` fields. The accuracy of the classifier is simply the number of correct classifications over the total number of aspects to be classified.

4 Method

All review sentences are preprocessed using the Stanford CoreNLP package [9], performing basic operations such as tokenization, part-of-speech tagging, lemma-

tization, syntactic analysis, as well as sentiment analysis. The latter is an already trained neural network that assigns a numeric sentiment score to each syntactic constituent in a parse tree.

For the machine learning backup method, we opted for a Support Vector Machine (SVM) with a radial basis function kernel, given that SVMs have proven to be very effective for text classification problems [10]. Since the `polarity` field can have three sentiment values, a multi-class SVM is trained that is able to classify an aspect into one of three sentiment values: positive, neutral, or negative. For this work, the Weka [5] implementation of the multiclass SVM is utilized, which internally performs 1-vs-1 pairwise classifications.

4.1 Ontology Design

For the ontology, the aim is to limit the number of asserted facts and to use the reasoner to infer the sentiment of a given expression. The ontology consists of three main classes: *AspectMention*, *SentimentMention*, and *SentimentValue*. The latter simply has *Positive* and *Negative* as its subclasses, and the setup is such that if a certain concept is positive, it is a subclass of *Positive* and if it expresses a negative sentiment, that concept is modeled as a subclass of *Negative*. The *AspectMention* class models the mentions of aspects and *SentimentMention* models the expressions of sentiment. A schematic overview of the ontology is shown in Fig. 3.

The *SentimentMentions* can be divided into three types. The first group is formed by type-1 *SentimentMentions*, which always denote a positive (negative) sentiment, regardless of which aspect they are about. In Fig. 3, these are denoted with hexagons. These subclasses of *SentimentMention* are also a subclass of the sentiment class they express. Hence, *Good* is a subclass of both *SentimentMention* and *Positive*. Type-2 *SentimentMentions* are those expressions that are exclusively used for a certain category of aspects, meaning that the presence of the aspect category can be inferred from the occurrence of the *SentimentMention*. In Fig. 3, these classes are denoted with rounded squares. For instance, *Delicious* is a subclass of *SentimentMention*, but also of both *SustenanceMention* and *Positive*, where *SustenanceMention* encompasses concepts related to food and drinks. This means that if we want to predict the sentiment value of an aspect in the service category, we will ignore the word “delicious” if it is encountered, because it cannot possibly be about the current aspect. The third type (type-3) of *SentimentMentions* contains context-dependent sentiment expressions, and this group is shown as an ellipse in Fig. 3. Here, the inferred sentiment depends on the aspect category. For instance, *Small* when combined with *Price* is a subclass of *Positive*, while when it is combined with *Portion* it is a subclass of *Negative*. Some of the words in this group are not ambiguous per se, but are simply not indicative of any particular aspect category while at the same time not being generally applicable. An example is the concept *Fresh*, which is always positive, but can only be combined with certain aspects: it matches well with subclasses of *SustenanceMention* (e.g., “fresh ingredients”) and *Ambience-*

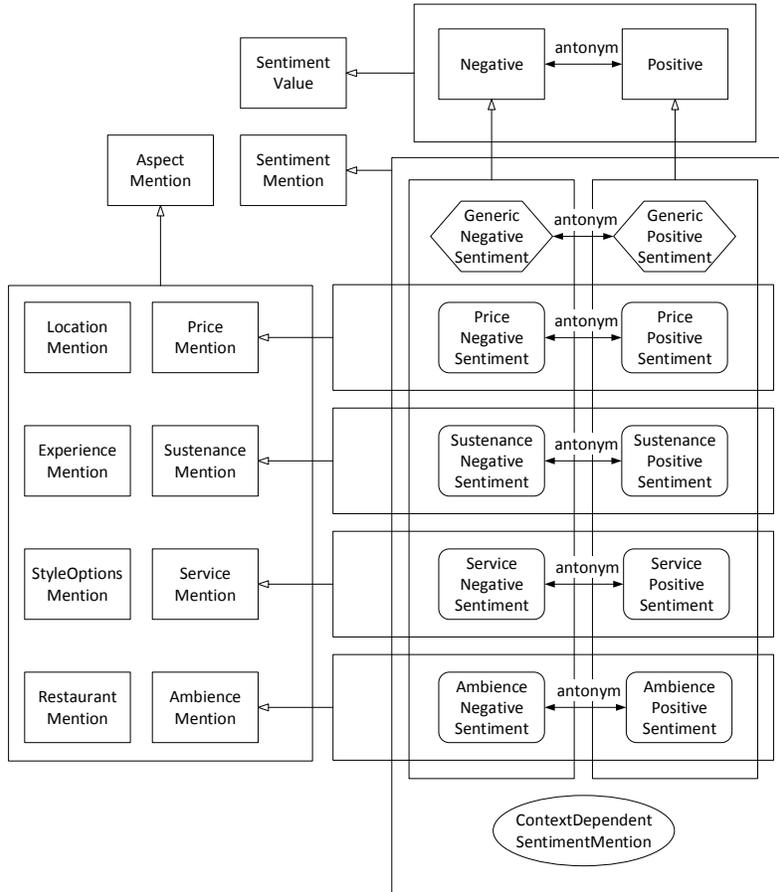


Fig. 3: A schematic overview of the main ontology classes

Mention (e.g., “a fresh decor”), but not with subclasses of, e.g., *PriceMention* or *LocationMention*.

When a type-1 **SentimentMention** is encountered, its sentiment value is used for the classification of all aspects within scope (i.e., the sentence). While the scope of the complete sentence can be considered too broad, as generic sentiment words usually apply to just one aspect, not all of them, in preliminary experiments, it was shown that limiting the scope to a word window or to steps over the grammatical graph is sub-optimal. A type-2 **SentimentMention** is only used for the classification of aspects that belong to the implied aspect category. For type-3 **SentimentMentions**, a new class is created that is a subclass of both the property class and the aspect class. If the ontology provides any information on that combination, its sentiment value can be inferred. Otherwise, the ontology does not provide any sentiment information for that combination of aspect and property.

The ontology is lexicalized by attaching annotations of type *lex* to each concept. A concept can have multiple lexicalizations, and since this ontology is designed to work within a single domain, there are not many ambiguous words that would point to more than one concept in the ontology. Furthermore, some concepts have one or more *aspect* properties, which link a concept to one of the aspect categories in the data annotations. This means that such a concept, and all of its subclasses fit within that aspect category. For instance, the **Ambience** concept has an *aspect* property with the value “AMBIENCE#GENERAL”. Last, concepts that are a subclass of **SentimentValue** have an *antonym* property that links that concept to its antonym (e.g., **Positive** has antonym **Negative**). This is used when found ontology concepts are subject to negation.

For this research, a domain ontology is manually constructed using the OntoClean methodology [6], and represented in OWL. To demonstrate the usefulness of ontologies, a choice is made for a relatively small, but focused ontology. Hence, it contains about 365 concepts, predominantly *AspectMentions*, but also including 53 type-1 *SentimentMentions*, 38 type-2 *SentimentMentions*, and 15 type-3 *SentimentMentions*. The maximum depth of the class hierarchy, not counting *owl:Thing* at the top, is 7.

4.2 Sentiment Computation

An overview of the sentiment computation method is shown in Alg. 1, outlining the three cases for type-1, type-2, and type-3 sentiment expressions, respectively. The input for sentiment prediction is an ontology, an aspect, and whether or not a bag-of-words model is used as a backup method in case the ontology does not specify a single sentiment value for this aspect. The PREDICTSENTIMENT method starts by retrieving all the words that are linked to the ontology with a URI and that are in the sentence containing the aspect. It also checks whether the current word is negated or not. For this we look for the existence of a *neg* relation in the dependency graph, or the existence of a negation word in a window of three words preceding the current word [7].

In the next step, the type of the concept is retrieved from the ontology and, depending on its type, the algorithm executes one of three cases. As mentioned before, if the concept is a type-2 sentiment expression, then its inferred aspect category has to match with the current aspect, otherwise it is ignored. For example, when encountering the word “delicious”, it leads to the concept **Delicious** due to the lexical property, which is a subclass of **SustenancePositiveProperty**.

1. **Delicious** $\equiv \exists lex.\{\text{“delicious”}\}$
2. **Delicious** \sqsubseteq **SustenancePositiveProperty**
3. **SustenancePositiveProperty** \sqsubseteq **Sustenance** \sqcap **Positive**

Furthermore, the **Sustenance** concept is linked to several aspect categories that exist in the annotated dataset by means of an *aspect* property.

4. **Sustenance** $\equiv \exists aspect.\{\text{“FOOD#QUALITY”}\}$
5. **Sustenance** $\equiv \exists aspect.\{\text{“FOOD#STYLE_OPTIONS”}\}$

Hence, when the current aspect for which we want to compute the sentiment is annotated with either one of those two categories, the word “delicious” is considered to be of positive sentiment. For aspects with a different category, the same word is considered to be neutral.

Algorithm 1 Pseudocode for computing aspect sentiment

```

1: function PREDICTSENTIMENT(Ontology o, Aspect a, boolean useBOW) : SentimentValue
2:   Set<String> foundURIs =  $\perp$ 
3:   Set<Word> words = GETWORDSWITHURI(GETSENTENCE(a))
4:   for all Word w  $\in$  words do
5:     boolean negated = ISNEGATED(w)
6:     String URI = GETURI(o, w)
7:     if ISTYPE1(o, URI) then
8:       foundURIs = foundURIs  $\cup$  GETSUPERCLASSES(o, URI, negated)
9:     end if
10:    if ISTYPE2(o, URI)  $\wedge$  CATEGORYMATCHES(a,URI) then
11:      foundURIs = foundURIs  $\cup$  GETSUPERCLASSES(o, URI, negated)
12:    end if
13:    if ISTYPE3(o, URI) then
14:      for all String relURI  $\in$  GETRELATEDASPECTMENTIONS(w) do
15:        String newURI = ADDSUBCLASS(o, URI, relURI)
16:        foundURIs = foundURIs  $\cup$  GETSUPERCLASSES(o, newURI, negated)
17:      end for
18:    end if
19:  end for
20:  boolean foundPositive = (PositiveSentimentURI  $\in$  foundURIs)
21:  boolean foundNegative = (NegativeSentimentURI  $\in$  foundURIs)
22:  if foundPositive  $\wedge$   $\neg$  foundNegative then
23:    return Positive
24:  else if  $\neg$  foundPositive  $\wedge$  foundNegative then
25:    return Negative
26:  else if useBOW then
27:    return GETBOWPREDICTION(a)
28:  else
29:    return GETMAJORITYCLASS // this is Positive in our data sets
30:  end if
31: end function

```

If the current **SentimentMention** is generic (type-1) or matching aspect-specific (type-2), then all superclasses are added to the set of *foundURIs*. If the current concept is a type-3, or context-dependent, **SentimentMention**, we need to check if it is related to an **AspectMention** and whether the combination of those two triggers a class axiom or not. Hence, we create a subclass with both the **SentimentMention** and the **AspectMention** as its direct superclasses, and add all (inferred) superclasses to the set of *foundURIs*. If there is a class axiom covering this combination, then the set of all inferred superclasses of this new subclass will include either **Positive** or **Negative**. When the current word was determined to be negated, the **GETSUPERCLASSES** method will add the antonym of each superclass instead, provided the ontology has an antonym for that class.

A good example of a Type-3 **SentimentMention** is **Small**, for which the ontology contains two sentiment-defining class axioms in the ontology, as well as a property that links the concept to the lexical representation “small”.

1. $\text{Small} \equiv \exists \text{lex}.\{\text{“small”}\}$
2. $\text{Small} \sqcap \text{Price} \sqsubseteq \text{Positive}$
3. $\text{Small} \sqcap \text{Serving} \sqsubseteq \text{Negative}$

Furthermore, $\text{Portion} \sqsubseteq \text{Serving}$ and we assume the review text contains a phrase like “small portions”, “portions are small”, or something similar. First, the words “small” and “portions” are linked to their respective ontology concepts by means of the `lex` attribute. Then, since, **Small** is neither a generic type-1 **SentimentMention**, nor an aspect-specific type-2 **SentimentMention**, it is paired with related words in the sentence to see if there are any class axioms to take advantage of. In this case, *small* is directly related to *portions*, so a new class is created called **SmallPortion**, that is a direct subclass of **Small** and **Portion**:

4. $\text{SmallPortion} \sqsubseteq \text{Small} \sqcap \text{Portion}$

This triggers the class axiom defined earlier, leading to

5. $\text{SmallPortion} \sqsubseteq \text{Negative}$

Hence, **Negative** is added to the list of found classes, as all the other superclasses were already known as superclasses from the two individual classes.

The last step is to check whether the previous inferences have resulted in finding **Positive** or **Negative**. If we find one but not the other, the aspect is determined to have the found sentiment value. If either no sentiment value is found, or both sentiment values are found, the ontology does not give a definitive answer. In that case, if we opt to use a bag-of-words backup model, then it is used here. If bag-of-words is not used, we default to predicting **Positive** as that is the majority class.

4.3 Bag-of-words model

The bag-of-words model is used both as a baseline, and as a backup model in case the ontology cannot decide which sentiment to assign. For the most part, it is a classical bag-of-words model with binary features for each lemma in the review that contains the current aspect. In preliminary experiments, this gave better results than using the lemmas from the sentence only. We hypothesize that this might be due to the fact that with more words, it is easier to get the overall sentiment of the review correctly, while for sentences, being a lot smaller, this would be harder. Given that the majority of the aspects follow the overall sentiment of the review, the effect of having more words to work with is larger than the effect of missing out on those aspects with a sentiment value different from the overall review. Furthermore, there is a set of dummy features to encode the aspect category as well as a numerical feature denoting the sentiment of the sentence. This sentiment score is computed by a sentiment component [15] in the

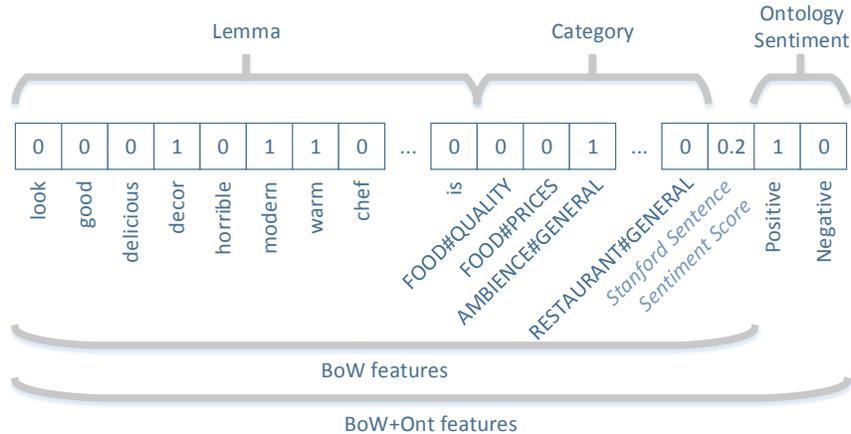


Fig. 4: Feature vector example for BoW+Ont model

Stanford CoreNLP package and falls roughly in the range of $[-1,1]$. The model is trained as a multi-class Support Vector Machine that is able to predict positive, negative, and neutral. These last two features are aspect-specific and sentence-specific, so the model is technically not bound to predict the same sentiment for all aspects within the same review. The feature vector is illustrated in Fig. 4.

4.4 Bag-of-words model with ontology features

Besides the rule-based ontology method using the bag-of-words model as a backup, it also makes sense to use the bag-of-words model as the leading model and add ontology information in the form of additional features. Hence, we add two binary features to the bag-of-words model, one to denote that the presence of the **Positive** concept and one to denote the presence of the **Negative** concept (see Fig. 4). Furthermore, to keep it in line with the rule-based ontology method, when both **Positive** and **Negative** are present, this is regarded as having no information so both features will be zero.

5 Evaluation

To evaluate the performance of the proposed method and the baselines, all methods are trained on the training data and tested on the official test data. To determine the required (hyper)parameters, such as C and γ for the SVM, about 20% of the training data is reserved as a validation set. After the optimal values have been found, the model is trained using those settings on the whole training data. This is done for both the 2015 and 2016 editions of Restaurant data set from the SemEval ABSA task and the results are shown in Table 1 and Table 2, respectively. From the results, we can conclude that the ontology method on its own is not sufficient, which is caused by the fact that it only works for roughly 50% of the aspects and defaults to predicting the majority

class for the other half. However, as evidenced by the increased performance for both hybrid methods, the ontology method is able to provide information that is complementary to the information contained in the bag-of-words.

Table 1: Comparison of the four methods on the 2015 data, using out-of-sample, in-sample, and 10-fold cross-validation performance.

	out-of-sample	in-sample	cross-validation	st.dev	paired two-sided t-test		
	accuracy	accuracy	accuracy		Ont	BoW	Ont+BoW
Ont	63.3%	79.4%	79.3%	0.0508			
BoW	80.0%	91.1%	81.9%	0.0510	5.98E-05		
Ont+BoW	82.5%	89.9%	84.2%	0.0444	4.67E-14	3.70E-09	
BoW+Ont	81.5%	91.7%	83.9%	0.0453	6.74E-09	2.95E-05	0.453

Table 2: Comparison of the four methods on the 2016 data, using out-of-sample, in-sample, and 10-fold cross-validation performance.

	out-of-sample	in-sample	cross-validation	st.dev	paired two-sided t-test		
	accuracy	accuracy	accuracy		Ont	BoW	Ont+BoW
Ont	76.1%	73.9%	74.2%	0.0527			
BoW	82.0%	90.0%	81.9%	0.0332	4.96E-14		
Ont+BoW	86.0%	89.3%	84.3%	0.0319	3.52E-20	1.83E-11	
BoW+Ont	85.7%	90.4%	83.7%	0.0370	7.63E-17	5.37E-06	0.0304

Since the results on the official test data sets are comparable with previous SemEval submissions, an overview of the top 6 best performing systems is given in Table 3 with our proposed system listed in bold. Note that the proposed system did not participate in SemEval together with these systems, so the sole function of Table 3 is to provide context for the listed performances.

For various amounts of training data, the accuracies of all four methods are plotted in Fig. 5 and Fig. 6. Since the Ont method does not depend on training data, its performance remains constant. However, we can see that both hybrid methods consistently outperform the BoW baseline and that the difference in performance widens with less training data, especially on the 2015 data. Since the performance of both methods would depend on which part of the training data is randomly selected, the reported numbers are the average of 5 runs.

Because the ontology-based method so clearly distinguishes between different choices based on whether the positive and/or negative class is detected in the sentence, an overview is given in Table 4 of the performance of the ontology-based method (without BoW backup) as well as the bag-of-words model (without Ont features), split out per scenario. From this, it is evident that the knowledge-based approach complements the traditional machine learning method. When able to make a decision, the ontology-based method performs better than the bag-of-words model (top two lines in Table 4), but the reverse is true when the ontology does not have the information to come to a conclusion (bottom two lines in Table 4). In that case, it is better to use the bag-of-words model than to default to the majority class, which is the default behavior of the ontology-based method without BoW backup. Interestingly, the aspects for which both a positive

Table 3: Ranks of the proposed method in top of SemEval-2015 and SemEval-2016 ranking

2015		2016	
Team	Accuracy	Team	Accuracy
Ont+BoW	82.49	XRCE	88.1%
sentiue	78.7%	IIT-T	86.7%
ECNU	78.1%	Ont+BoW	86.0%
Isislif	75.5%	NileT	85.5%
LT3	75.0%	IHS-R	83.9%
UFRGS	71.7%	ECNU	83.6%
wnlp	71.4%	AUEB	83.2%

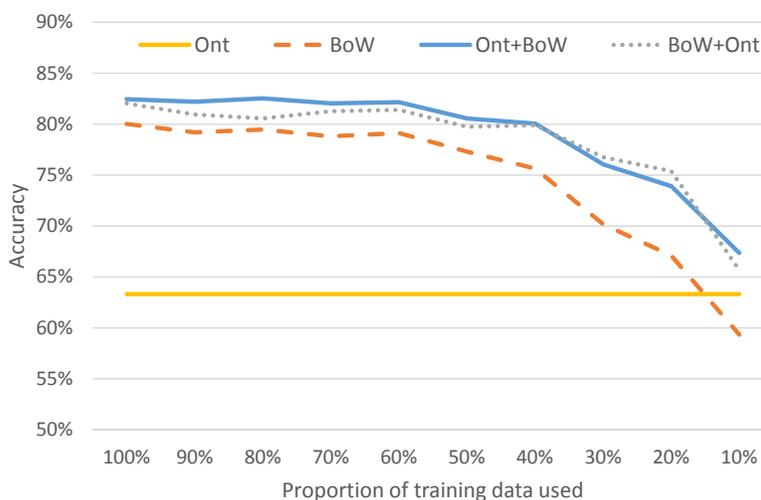


Fig. 5: The accuracy of all four methods at different amounts of training data (SemEval-2015)

and a negative sentiment is detected in the sentence are harder for the bag-of-words model to predict the sentiment for than sentences where the ontology did not find any sentiment expressions. The fact that the bag-of-words model does relatively well on the latter suggests omissions in the ontology. Clearly, the bag-of-words model is able to find some clues as to what sentiment to predict, even though these are not present in the ontology.

6 Conclusion

In this paper, an ontology-based method for aspect sentiment analysis is presented. It utilizes domain information, encoded in an ontology, to find cues for positive and negative sentiment. When such cues are either not found, or when both positive and negative cues are present, a bag-of-words model is used as

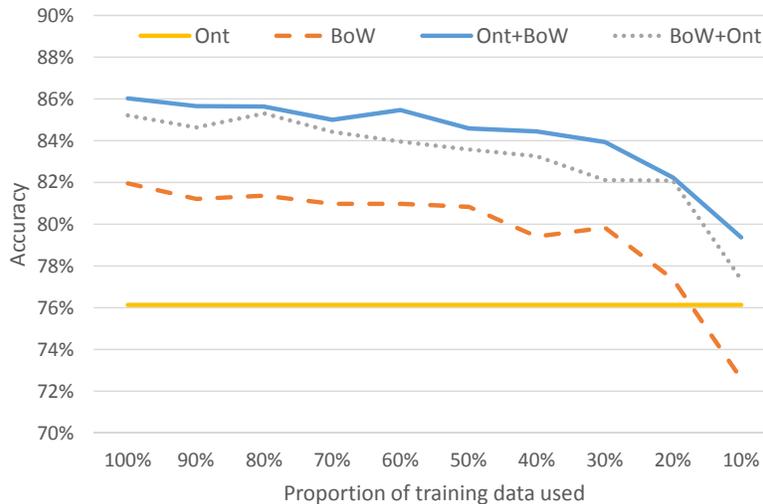


Fig. 6: The accuracy of all four methods at different amounts of training data (SemEval-2016)

Table 4: The performance of the ontology and bag-of-words based on whether the **Positive** and/or **Negative** concept from the ontology was found or inferred for an aspect.

	out-of-sample 2015 data			out-of-sample 2016 data		
	size	acc. ontology	acc. BoW	size	acc. ontology	acc. BoW
Found only Positive	42.7%	88.1%	83.7%	55.3%	93.1%	87.6%
Found only Negative	9.8%	94.0%	85.5%	8.4%	73.2%	62.0%
Found both	4.3%	47.2%	52.8%	4.1%	62.9%	68.6%
Found none	43.2%	33.4%	77.3%	33.8%	50.7%	79.7%

a backup method. The ontology-based method and the bag-of-words model are shown to complement each other, resulting in a hybrid method that outperforms both. Since the ontology-based model does not need any training data, the performance of the hybrid method also depends less on having sufficient training data, and this effect was illustrated empirically as well.

For future work, we suggest looking into expanding the ontology, as there is still a large group of aspects for which no sentiment expression could be found. This process could be automated by scraping restaurant reviews from the Web and using the assigned star rating, or something similar, as sentiment information to classify found expressions as being positive or negative.

References

1. Cambria, E., Hussain, A.: Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis. Springer (2015)

2. Cambria, E., Olsher, D., Rajagopal, D.: SenticNet 3: A Common and Common-Sense Knowledge Base for Cognition-driven Sentiment Analysis. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI 2014). pp. 1515–1521. AAAI (2014)
3. Cambria, E.: Affective Computing and Sentiment Analysis. *IEEE Intelligent Systems* 31(2), 102–107 (2016)
4. Dragoni, M., Tettamanzi, A., da Costa Pereira, C.: A Fuzzy System for Concept-Level Sentiment Analysis. In: Semantic Web Evaluation Challenge at Extended Semantic Web Conference. pp. 21–27. Springer (2014)
5. Frank, E., Hall, M.A., Witten, I.H.: The WEKA Workbench. Online Appendix for “Data Mining: Practical Machine Learning Tools and Techniques”, Fourth Edition. Morgan Kaufmann (2016)
6. Guarino, N., Welty, C.: Evaluating Ontological Decisions with OntoClean. *Communications of the ACM* 45(2), 61–65 (2002)
7. Hogenboom, A., van Iterson, P., Heerschop, B., Frasincar, F., Kaymak, U.: Determining Negation Scope and Strength in Sentiment Analysis. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2011). pp. 2589–2594. IEEE (2011)
8. Lau, R.Y., Li, C., Liao, S.S.: Social Analytics: Learning Fuzzy Product Ontologies for Aspect-Oriented Sentiment Analysis. *Decision Support Systems* 65(C), 80–94 (2014)
9. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. In: Proceedings of 52nd Annual Meeting of the ACL: System Demonstrations. pp. 55–60. ACL (2014)
10. Mullen, T., Collier, N.: Sentiment Analysis using Support Vector Machines with Diverse Information Sources. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004). vol. 4, pp. 412–418. ACL (2004)
11. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., Clercq, O.D., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S.M., Eryiğit, G.: SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016). ACL (2016)
12. Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., Androutsopoulos, I.: SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In: Proceedings of the Ninth International Workshop on Semantic Evaluation (SemEval 2015). pp. 486–495. ACL (2015)
13. Schouten, K., Frasincar, F.: Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* 28(3), 813–830 (2016)
14. Schouten, K., Frasincar, F., de Jong, F.: Ontology-Enhanced Aspect-Based Sentiment Analysis. In: Proceedings of the 17th International Conference on Web Engineering (ICWE 2017). Lecture Notes in Computer Science, vol. 10360. Springer (2017)
15. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.P.: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: Proceedings of the 2013 Conference on Empirical Methods on Natural Language Processing (EMNLP 2013). pp. 1631–1642. ACL (2013)
16. Xia, Y., Cambria, E., Hussain, A., Zhao, H.: Word Polarity Disambiguation Using Bayesian Model and Opinion-Level Features. *Cognitive Computation* 7(3), 369–380 (2015)