

Ontology-Enhanced Aspect-Based Sentiment Analysis

Kim Schouten, Flavius Frasinca, and Franciska de Jong

Erasmus University Rotterdam
P.O. Box 1738, NL-3000 DR Rotterdam, the Netherlands
{schouten, frasinca}@ese.eur.nl, f.m.g.dejong@eshcc.eur.nl

Abstract. With many people freely expressing their opinions and feelings on the Web, much research has gone into modeling and monetizing opinionated, and usually unstructured and textual, Web-based content. Aspect-based sentiment analysis aims to extract the fine-grained topics, or aspects, that people are talking about, together with the sentiment expressed on those aspects. This allows for a detailed analysis of the sentiment expressed in, for instance, product and service reviews. In this work we focus on knowledge-driven solutions that aim to complement standard machine learning methods. By encoding common domain knowledge into a knowledge repository, or ontology, we are able to exploit this information to improve classification performance for both aspect detection and aspect sentiment analysis. For aspect detection, the ontology-enhanced method needs only 20% of the training data to achieve results comparable with a standard bag-of-words approach that uses all training data.

1 Introduction

With many people freely expressing their opinions and feelings on the Web, much research has gone into modeling and monetizing opinionated, and usually unstructured and textual, Web-based content [14]. A popular option to extract information from Web texts is to perform sentiment analysis. Given a certain unit of text, for instance a document or a sentence, the task of sentiment analysis is to compute the overall sentiment expressed by the author of the text. Text can be tagged for sentiment by using labels for emotions, or by assigning a polarity value to the processed unit of text, which is a more commonly adopted method. Aspect-based sentiment analysis goes a step deeper. Rather than labeling a document or a sentence, it aims to extract and tag semantic units. It captures the topics, or aspects, that are being talked about, together with the sentiment expressed about those aspects [20]. Relating the expressed sentiment directly to certain topics enables the extraction of opinions expressed in product and service reviews in a much more focused way. Instead of an overall score in which both positive and negative aspects are combined, a breakdown can now be provided, showing the aspects for which the reviewer said positive things and the aspects he or she was less enthusiastic about.

Generally speaking, we can define two sub-problems that together comprise aspect-based sentiment analysis: aspect detection and aspect sentiment classification. We define aspect detection as capturing the topics, or aspects, that are being talked about. This can be done within the textual unit of choice, for instance per sentence or per document. Most of the aspects will be explicitly mentioned in the textual unit, and the exact phrase that mentions the aspect is defined as the target expression of the aspect. Aspects without a target expression are not explicitly mentioned but rather are implied by a, usually larger, portion of text. Since target expressions are often very specific, it can be informative to group aspects together in aspect categories. Implicit aspects, even though lacking a specific target expression, can be categorized in the same manner. Even within sentences, multiple aspects, both explicit and implicit, can occur, as shown in Example 1. This sentence contains two explicit aspects: “chow fun” and “pork shu mai”, both belonging to the broader ‘food’ category, as well as an implicit aspect about sharing the table with a loud and rude family, which can be categorized under ‘ambiance’.

Example 1 *“Chow fun was dry; pork shu mai was more than usually greasy and had to share a table with loud and rude family.”*

Aspect sentiment analysis, the second sub-problem of aspect-based sentiment analysis, can be defined as determining the sentiment expressed on that sentiment in the text where the aspect is mentioned. For explicit aspects, the target expression indicates where in the text the aspect is mentioned and this information can be useful in determining the relevance of each piece of sentiment carrying text as textual units, such as sentences, can contain multiple aspects that have differing sentiment values. This is illustrated in Example 2 in which one sentence contains two aspects, one about the food and one about the service, but the expressed sentiment on these two aspects is completely different. For implicit aspects, where target expressions are not available, the complete textual unit can be relevant, but the aspect category usually provides some information on which part of the sentence might be relevant.

Example 2 *“The food was great, if you’re not put off by the rude staff.”*

Current approaches for aspect-based sentiment analysis rely heavily on machine learning methods because this yields top performance [17]. Deep learning approaches are especially popular and include techniques such as word embeddings [16], convolutional neural networks [11], recursive neural tensor networks [21], and long short-term memory networks [22]. While the above methods have shown very promising results in various natural language processing tasks, including sentiment analysis, there are some downsides as well. For example, while the methods learn their own features, often better than what an individual researcher could come up with, they do so at the cost of requiring much training data. While this may not be a problem for resource-rich languages, such as English, and resource-rich domains such as reviews or tweets, it is a real issue for many other languages and domains where training data are not abundant or simply unavailable.

With the previous argumentation in mind, we propose a knowledge-driven approach to complement traditional machine learning techniques. By encoding some common domain knowledge into a knowledge repository, or ontology, we limit our dependence on training data [3]. The idea is that, compared to using only information from the text itself, relating the text to the concepts described in a knowledge repository will lead to stronger signals for the detection of aspects as well as the prediction of sentiment. Consequently, having stronger signals limits the amount of necessary training data, as the relation between input and desired output is easier to discover.

While knowledge repositories, such as ontologies, have to be adjusted for every domain and language, this is a less resource-intensive task than manually annotating a big enough corpus for the training of a deep neural network. Furthermore, since ontologies are designed to be reused, for example within the Linked Open Data cloud, it is easy to share knowledge. Last, with ontologies, being logically sound structures, it is possible to reason with the available data, and to arrive at facts not directly encoded in the ontology. For example, if meat is a type of food, and food is edible, we can state that meat is edible, even without directly specifying this. Furthermore, inferencing functionality can help to disambiguate sentiment carrying phrases given the context they appear in, for example by taking into account that “cold” is positive for “beer”, but negative for “pizza”. This opens up some exciting possibilities when performing sentiment analysis.

This paper is structured as follows. In the next section, some of the related work is presented, followed by a discussion of the problem and the used data set in Sect. 3. In Sect. 4, an overview of the proposed method is given, and in Sect. 5, its performance is compared and evaluated. This paper concludes with Sect. 6, providing both conclusions and possible avenues for future work.

2 Related Work

In [3], a short overview is given of the field of affective computing and sentiment analysis, and the author makes a case for the further development of hybrid approaches that combine statistical methods with knowledge-based approaches. The leading idea in that field is that the intuitive nature and explanatory power of knowledge-based systems should be combined with the high performance of machine learning methods, which also forms the research hypothesis of our current work.

Sentic computing is presented in [1], which combines statistical methods 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 method that only uses the bag of words is employed to determine the

sentiment for that sentence. Note that this is a sentence level approach and not an aspect based approach.

In [5], a multi-domain approach to sentence-level sentiment analysis is presented, where the task is to assign sentiment to each sentence, but where the sentences could come from a variety of domains. The proposed method is designed in such a way that sentiment words can be disambiguated based on the recognized domain the sentence originates from. Similar to what is typical of our approach, the used knowledge graph is split into two main parts: a semantic part in which the targets are modeled, and a sentiment part in which the links between concepts and sentiment are described. A big difference with our approach is that while we opt for a focused domain ontology, in [5], due to the multi-domain nature of the problem, a very broad knowledge graph is created that combines several resources such as WordNet [7] and SenticNet [2]. Another difference is the use of fuzzy membership functions to describe the relations between concepts and domains, as well as between concepts and sentiment. This gives more flexibility in terms of modeling, but makes it harder to reason over the knowledge graph.

In [18], an extended version of the Sentilo framework is presented that is able to extract opinions, and for each opinion its holder, expressed sentiment, topic, and subtopic. The framework converts natural language to a logical form which in turn is translated to RDF that is compliant with Semantic Web and Linked Data design principles. Then, concepts with relations that signal sentiment are identified. The sentiment words receive a sentiment score based on SentiWordNet [6] and SenticNet [2].

A typical problem for which external knowledge can be very useful is the issue of sentiment ambiguity: a certain expression can be positive in one context and negative in the other (e.g., the cold pizza and cold beer example from the previous section). This problem is tackled in [24] by means of a Bayesian model that uses the context around a sentiment carrying word, in particular the words that denote the aspect, to determine the polarity of the sentiment word. When there is not enough information in the context to make the decision, a backup method is to retrieve inter-opinion data, meaning that if the previous opinion was positive and there is a conjunction between that one and the current one, it is very likely that the current opinion is positive too.

In contrast to the previous approaches, high performing SemEval submissions on the aspect-based sentiment analysis task [17], are typically limited in their use of external knowledge or reasoning. For instance, in the top performing system for aspect category classification [23], a set of binary classifiers is trained, one for each aspect category, using a sigmoidal feedforward network. It uses words, bigrams of words, lists of opinion target words extracted from the training data, syntactic head words, and Brown word clusters as well as k-mean clusters from word2vec [16]. The highest performing system in the sentiment classification task [19] also exclusively focuses on using lexical features. Besides the information that can be directly extracted from the text, a number of lexical resources such as sentiment lexicons were used to detect the presence of negation and sentiment

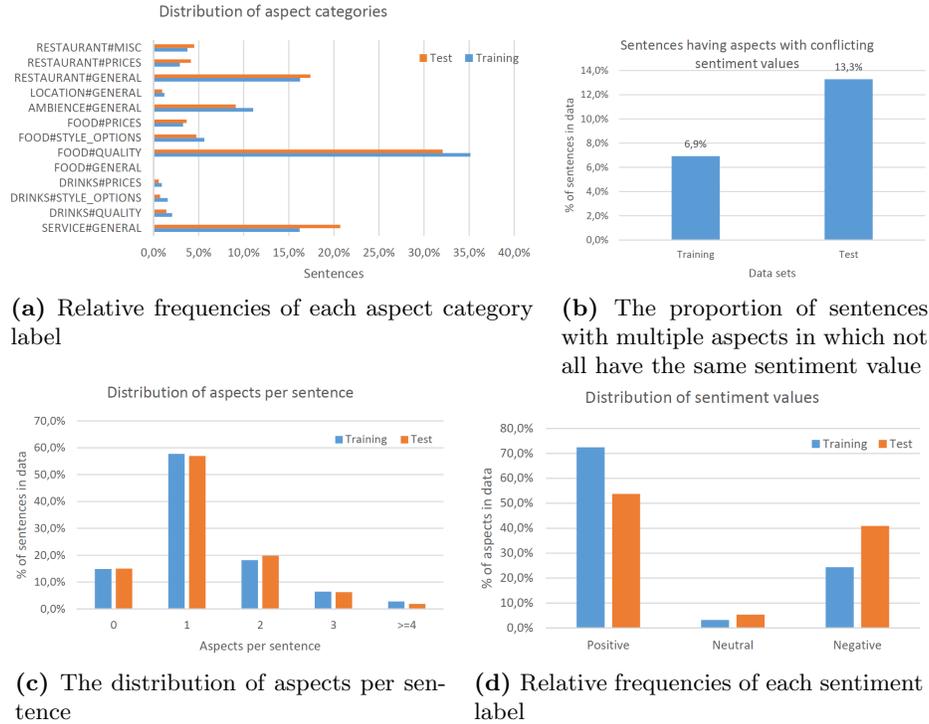


Fig. 1: Some statistics related to the used data set

words. While lexical resources can be seen as being external knowledge, they are limited in functionality and do not, for example, support reasoning.

3 Specification of Data and Tasks

The data set used in this research is the widely known set of restaurant reviews from SemEval [17], with each review containing one or more sentences: it contains 254 reviews with in total 1315 sentences. Each sentence is annotated with zero, one, or multiple aspects, and each aspect is put into a predefined aspect category and is labeled as either positive, neutral, or negative. For explicit aspects, the target expression is also provided. Some statistics related to aspects and sentiment can be found in Fig. 1. In Fig. 1a, the number of times each category label appears is presented and in Fig. 1b, the proportion of sentences with multiple aspects is shown in which not all aspects have the same sentiment label. This is related to Fig. 1c, which shows the distribution of aspects per sentence. Fig. 1d presents the distribution of sentiment values over aspects, showing that this data set, especially the training data set, is unbalanced with respect to sentiment.

```

<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. 2: A snippet from the used dataset showing an annotated sentence from a restaurant review.

A snippet of the used data set is shown in Fig. 2. The data set is already organized by review and by sentence, and each sentence is annotated with zero or more opinions, which represent the combination of an aspect and the sentiment expressed on that aspect.

For the aspect detection task, only the sentence is given. The task is to annotate the sentence with aspects but the `polarity` field can be left empty. While some variations of the task exist, we limit ourselves to predicting only the `category` field of each aspect. Hence, the `target` field and corresponding `to` and `from` fields are ignored. The category labels themselves consist of an entity and an attribute that are separated by a hash symbol. In this work, however, we regard the category as just a single label. In the evaluation, every category that is in the data and is also predicted is a true positive, every category that is predicted but is not in the data is a false positive and every category that is not predicted, even though it is in the data, is a false negative. From the number of positives and negatives, the standard evaluation metrics of precision, recall, and F_1 score can be computed.

For the sentiment classification task, the sentence with the aspects are given. Thus, we get everything in Fig. 2, except for the values of the `polarity` fields. Every correctly predicted polarity is a true positive and every incorrectly predicted polarity is both a false positive and a false negative, so precision, recall, and F_1 have the same value for this task.

4 Method

Since both detecting aspects and determining their sentiment can be seen as a classification task, we choose an existing classifier to work with. In this case, we choose to use a linear Support Vector Machine (SVM), since it has shown good performance in the text classification domain, with relatively many input

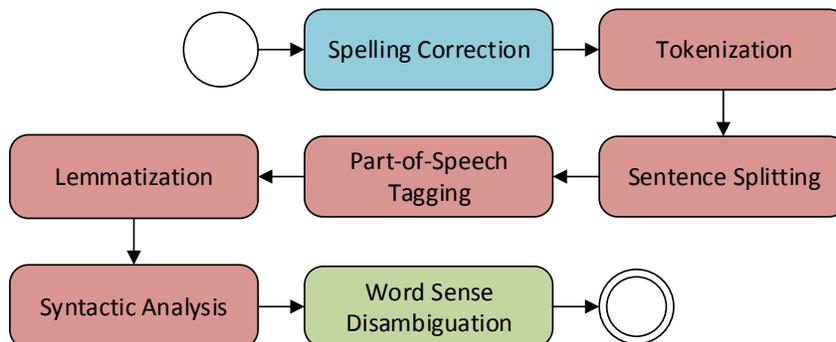


Fig. 3: The NLP pipeline used at the basis of the methods

features compared to the number of training examples [4]. For aspect detection, we train an independent binary classifier for each different aspect category. In this way, per sentence, each classifier will determine whether that aspect is present or not, enabling us to find zero, one, or more aspects per sentence. For sentiment classification, we train only one (multiclass) model, that is able to predict one of three outcomes: positive, neutral, or negative. We use the libsvm [4] implementation of the SVM classifier.

Using natural language processing (NLP) techniques, we gather information from the review texts that will comprise the input vector for the SVM. In Fig. 3, the components of the NLP pipeline are shown. First, an automated spelling correction is performed, based on the JLanguageTool library [10]. Given the nature of consumer-written reviews, this is a very helpful step. Next, the text is split into tokens, which are usually words, but also includes punctuation. Then, these tokens are combined into sentences. With sentence boundaries defined, the words in each sentence can be tagged with Part-of-Speech tags, which denote the word types (e.g., noun, verb, adjective, etc.). Once that is known, words can be lemmatized, which means extracting the dictionary form of a word (e.g., reducing plurals to singulars). The syntactic analysis then finds the grammatical relations that exist between the words in a sentence (e.g., subject, object, adjective modifier, etc.). All these components are provided by the Stanford CoreNLP package [15]. The last step connects each word with a specific meaning called a synset (i.e., set of synonyms), given the context in which it appears, using a Lesk [13] variant and WordNet semantic lexicon [7]. This particular version of the Lesk algorithm is provided by DTU [9].

4.1 Ontology Design

The ontology consists of three main parts, modeled as top-level classes. The first is a *Sentiment* class, which has individuals representing the various values of

sentiment. In our case, that is only positive, neutral, and negative, but one can imagine a more fine grained classification using basic emotion classes like anger, joy, etc. The second major class is *Target*, which is the representation of an aspect. The higher level concepts correspond to aspect categories, while the subclasses are often target expressions of an aspect. Subclasses of *Target* are domain specific, and for our restaurant domain we use *Ambience*, *Sustenance*, *Service*, *Restaurant*, *Price*, *Persons*, and *Quality*. Some of these have only a handful of individuals, such as *Quality* since quality is more expressed in evaluative words and not in concepts, while *Sustenance*, unsurprisingly, has many subclasses and individuals. Because we want to use object relations, all subclasses of *Target* are modeled as having both the class role and the individual role, much like classes in OWL FULL. For every subclass of *Target*, there is an individual with the same (resource) identifier that is of that same type. Hence, there is a subclass *Beer*, and an individual *Beer* that is of type *Beer*. This duality allows us to use the powerful subclass relation and corresponding reasoning, as well as descriptive object relations. The latter are mainly used for the third part of the ontology, which is the *SentimentExpression* class.

The *SentimentExpression* class only has individuals describing various expressions of sentiment that can be encountered. Each sentiment expression is linked to a *Sentiment* value by means of an object relation called *hasSentiment*, and to a *Target* with an object relation called *hasTarget*. In most cases, the *hasTarget* relation points to the top-level concept *Target*, since the word “good” is positive regardless of the target. However, the word “cold”, when linked to the concept *Pizza* has the *negative* sentiment value, while it has the *positive* value when linked to *Beer*.

The ontology is lexicalized by means of a data property that is added to each concept. The targets have a *targetLexicalization* property, and the sentiment expressions have a *sentimentLexicalization* property. By means of these lexicalizations, which can be one or more words, the concepts in the ontology can be linked to words or phrases in the text.

In Fig. 4, the sentiment expression for “cold beer” is shown with its related concepts. Note that the ellipse around the *Beer* class and the *Beer* individual denotes the fact that those are two roles of the same concept.

This ontology design allows us to perform two types of reasoning, one for aspect detection and one for sentiment classification. The first is that if we encounter a sentiment word, we know that its target is also in this sentence. For example, when we find the word “delicious”, we will find the *SentimentExpression* with the same *sentimentLexicalization*. This concept has a target, namely the *Sustenance* concept. Because of this, we know that the sentence where we find “delicious”, the target aspect is something related to food or drinks. The second type of reasoning is that when we encounter a sentiment word in a sentence and that word is linked to a *SentimentExpression* in the ontology, the aspect for which we want to determine the sentiment has to be of the same type as the target of that *SentimentExpression* in order for its sentiment value to be relevant. For example, we again find the word “delicious” in the text, but

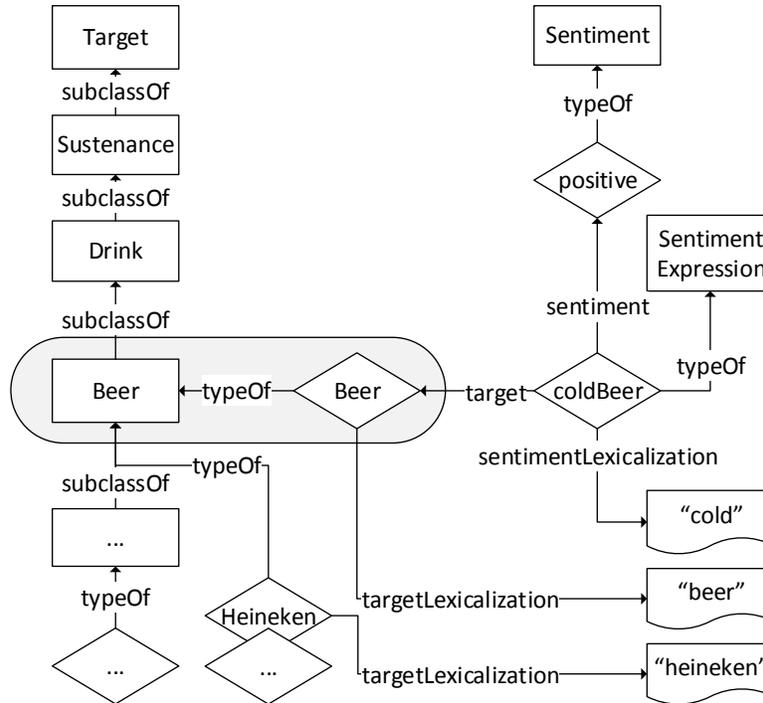


Fig. 4: Snippet of the ontology showing a sentiment expression and its related concepts

we want to determine the sentiment for the aspect `FOOD#PRICE`, we should not take the positive value of “delicious” into account, since it is not relevant to the current aspect we are classifying. This is especially useful when a sentence has more than one aspect.

The ontology is created manually, using the OntoClean methodology [8], so it is guaranteed to fit with the restaurant domain of the reviews. To keep the ontology manageable, we have deliberately opted for a relatively small, but focused, ontology. As such, it contains 56 sentiment expressions, 185 target concepts, and two sentiment concepts: positive and negative. The maximum depth of its class hierarchy is 5.

4.2 Features

Since the aspect detection task is defined as predicting zero or more aspect category labels per sentence, we extract the following features from each sentence: the presence or absence of lemmatized words, the presence or absence of WordNet synsets, and the presence or absence of ontology concepts. For the latter, we use words and phrases in the sentence to find individuals of top-level class *Target*

that have a matching *targetLexicalization*. When a concept is found, we include all its types as features. For example, when we find the concept *Steak*, we also include the concepts *Meat*, *Food*, *Sustenance*, and *Target*. Furthermore, when a word or phrase matches with a *sentimentLexicalization*, we include the *target* of that SentimentExpression as being present as well. All these features are binary, so the input vector for the SVM will contain 1 for features that are present, and 0 otherwise. The same features are used for each binary aspect classifier

This process of gathering features can be formalized as follows.

$$L_W = \{l | l = \text{lemma}(w), w \in W\} \quad (1)$$

$$Z_W = \{z | z = \text{synset}(w), w \in W\} \quad (2)$$

$$C_W = \{c | k : c, (k, \text{lemma}(w)) : \text{targetLexicalization}, w \in W\} \cup \\ \{c | (k, c) : \text{target}, (k, \text{lemma}(w)) : \text{sentimentLexicalization}, w \in W\} \quad (3)$$

where W represents a set of words, given as a parameter, $i : c$ represents an individual k of type c , and $(k, c) : \text{target}$ represents that k is related to c through relation type *target*. Then, let W' be the set of all words in the data set. Every word has its own unique representation in this set, so the same word appearing in three different places will have three entries in this set. And let S be the indexed set of all sentences in the data set, with functions $g : I \rightarrow S$, and $g' : S \rightarrow I$, so that $i \rightarrow s_i$, $s \rightarrow i_s$ and $I = \{i \in \mathbb{N} | i \geq 0, i < |S|\}$, resulting in a unique one-to-one mapping between I and S . Then W_i is defined as the set of words in sentence s_i .

Using W' , we gather all possible features from the full data set into set $F_{W'}$.

$$F_{W'} = L_{W'} \cup Z_{W'} \cup C_{W'} \quad (4)$$

Similar to S , set $F_{W'}$ is indexed with a one-to-one mapping: $h : F_{W'} \rightarrow J$ and $h' : J \rightarrow F_{W'}$, so that $j \rightarrow f_j$ and $f \rightarrow j_f$ with $J = \{j \in \mathbb{N} | j \geq 0, j < |F_{W'}|\}$.

Given this mapping between J and $F_{W'}$, the index numbers of only the features that are present in a given sentence i are retrieved through $h(F_{W_i})$. This leads to defining the input matrix \mathbf{X} as having

$$x_{ij} = \begin{cases} 1, & \text{if } j \in h(F_{W_i}) \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Where i specifies the row in the matrix, representing the current sentence and j specifies the column in the matrix, representing the current feature.

For sentiment classification, the process of gathering features is similar, so due to the page limit, the formalization is omitted here. The difference is that the scope here is a single aspect for which we want to determine the sentiment. An aspect already has the category information given, together with its position in the sentence, if applicable. Besides the features for lemmas, synsets, and ontology concepts, we also include the aspect category information of an aspect as a feature (e.g., **FOOD#QUALITY** or **FOOD#PRICE**). In addition, we include some sentiment information, using existing sentiment tools or dictionaries together

with our own ontology. Utilizing the Stanford Sentiment tool [21], which assigns sentiment scores (decimals between -1 and 1) to every phrase in the parse tree of a sentence, we add a feature that represents the sentiment of the whole sentence, as well as a feature that represents the sentiment of the smallest phrase containing the whole aspect. The latter is only available for explicit aspects, whereas the former is always available. Since the sentence sentiment score is additional information that can be useful, for instance when the aspect sentiment, as determined by this tool, is incorrect, we chose to always add this feature.

Since the lowest level of the parse tree is comprised of the words in a sentence, we use the same tool to get sentiment values for each word. A special review sentiment dictionary [12] is used to retrieve sentiment values for some of the words as well, and as a third source of sentiment information we use the ontology to find sentiment information for any word that can be linked to a *SentimentExpression* in the ontology. As explained in the previous section, we only take the latter into account when the aspect for which we want to determine the sentiment can be linked to a concept in the ontology that matches with the target concept of the sentiment expression. The *positive* concept is translated to a value of 1, and the *negative* concept is translated to a value of -1. All these sentiment values are averaged to arrive at a single sentiment value for a given word or phrase. However, when we do find an applicable sentiment expression in the ontology, preliminary experiments suggest to use double the weight for this value in the average computation. Assigning a higher weight is an intuitive course of action, since we are sure this is relevant information.

Some of the aspects have location information provided, so we know which words in the sentence are describing that particular aspect. When this information is available, we construct a scope around the aspect so words that are closer to the aspect are more valuable than words further away. The distance is measured in terms of grammatical steps between words. In this way, words that are grammatically related are close by, even though there might be many words in between them in the actual sentence. Based on some preliminary experiments, we compute the distance correction value as:

$$distanceCorrection = \max(0.1, 3 - grammaticalSteps) \quad (6)$$

Instead of having binary features (cf. Eq. 5), denoting just the presence of features, we multiply the distance correction value with the average sentiment value and use this as the weight for each word-based feature. The aspect category features, as well as the sentence sentiment and aspect sentiment features are not affected by this because distance and sentiment are irrelevant seeing that these features are not directly linked to specific words.

5 Evaluation

In this section, we will evaluate the proposed method and discuss the results. First, the used data sets are described, followed by a comparative overview of the performance of the method. Then, to test our hypothesis that less data is needed for knowledge-driven approaches, a series of experiments is performed

Table 1: The performance on the aspect detection task

	avg. F_1	st.dev.	p-values of two-sided t-test			in-sample F_1	out-of-sample F_1
			base	+S	+O	(training data)	(test data)
base	0.5749	0.0057	-	-	-	0.803	0.5392
+S	0.6317	0.0039	<0.0001	-	-	0.896	0.5728
+O	0.6870	0.0026	<0.0001	<0.0001	-	0.858	0.6125
+SO	0.6981	0.004	<0.0001	<0.0001	<0.0001	0.920	0.6281

Table 2: The performance on the aspect sentiment classification task

	avg. F_1	st.dev.	p-values of two-sided t-test			in-sample F_1	out-of-sample F_1
			base	+S	+O	(training data)	(test data)
base	0.7823	0.0079	-	-	-	0.831	0.7372
+S	0.7862	0.0049	0.0294	-	-	0.847	0.7349
+O	0.7958	0.0069	0.0002	0.0008	-	0.863	0.7479
+SO	0.7995	0.0063	<0.0001	<0.0001	0.0029	0.884	0.7527

with varying amounts of training data. This is followed by a feature analysis, showing which features are useful and demonstrating that the output of the algorithm can be explained.

5.1 Performance

To test the performance of the proposed method, we compare the full ontology-enhanced method (+SO) against more basic versions of the same algorithm, having the exact same setup except for some missing features: a version without synsets, but with ontology features (+O); a version with synsets but without ontology features (+S); and a version without both synsets and ontology features (base). The two tasks of the algorithm are tested separately, so the aspect sentiment classification is performed with the gold input from the aspect detection task. This prevents errors in aspect detection from influencing the performance analysis of the sentiment classification. The two performances on the two tasks are given in Table 1 and Table 2, respectively. The reported F_1 scores are averages over 10 runs where each run is using 10-fold cross-validation with randomly assigned folds. The standard deviation is also reported, together with the p-values of the two-sided t-test.

For aspect detection the picture is most clear, showing that every step towards including more semantic features, starting with synsets and going to ontology concepts, is a significant improvement over not including those features. Note that most of the improvement with respect to the base algorithm comes from the ontology. The synsets, while showing a solid improvement on their own, are able to increase the performance with much less when the ontology is also used (i.e., going from +O to +SO).

For aspect sentiment classification, the results are less pronounced than for aspect detection, but it still shows the same overall picture. Adding more semantic features improves the performance, and while the improvement is less, it is statistically significant. A key observation here is that for sentiment analysis,

Table 3: Ranks of the proposed methods in top of SemEval-2015 ranking

Aspect Detection		Aspect Sentiment Classification	
Team	Performance	Team	Performance
+SO	0.628	sentieue	0.787
NLANGP	0.627	ECNU	0.781
NLANGP	0.619	Isislif	0.755
UMDuluth-CS8761-12	0.572	+SO	0.752
UMDuluthTeamGrean	0.572	LT3	0.750
SIEL	0.571	UFRGS	0.717
sentieue	0.541	wnlp	0.714

we already employ a number of features that convey sentiment values, and as such, it is a strong signal that in spite of all that information, the ontology is still able to boost the performance.

This work is mainly focused on showing how ontologies have an added value for aspect-based sentiment analysis. Nevertheless, our methods show a competitive performance. In Table 3, an overview of the top performances of SemEval submissions on the same task are given [17]. These methods have been tested on the exact same test data, so their reported F_1 scores are directly comparable. For ease of reference, our +SO method is shown in bold, together with the top 6 out of 15 submissions for both tasks.

5.2 Data Size Sensitivity

Since we hypothesize that an ontology-enhanced algorithm needs less data to operate than a traditional machine learning method, we perform the following experiment. Taking a fixed portion of the data as test data, we train on an ever decreasing part of the total available training data. In this way, the test data remains the same, so results can be easily compared, while the size of the training data varies. This maps the sensitivity of the algorithms to training data size and the results are shown in Fig. 5 for the aspect detection task and in Fig. 6 for the aspect sentiment classification task.

When looking at the sensitivity of the aspect detection method, we can see that the base algorithm is quite sensitive to the size of the data set, dropping the fastest in performance when there are fewer training data. With synsets, the performance is more stable, but with less than 40% of the original training data, performance drops significantly. The two versions that include ontology features are clearly the most robust when dealing with limited training data. Even at 20% of the original training data, the performance drop is less than 10%.

For sentiment classification, it shows that all methods are quite robust with respect to the amount of training data. This might be because all variants are using external information in the form of the sentiment dictionaries, already alleviating the need for training data to a certain extent. The gap between ontology-enhanced methods and the ones without ontology features does not widen, so contrary to our hypothesis, ontology features do not reduce the required number of training data for aspect sentiment analysis. On the other hand,

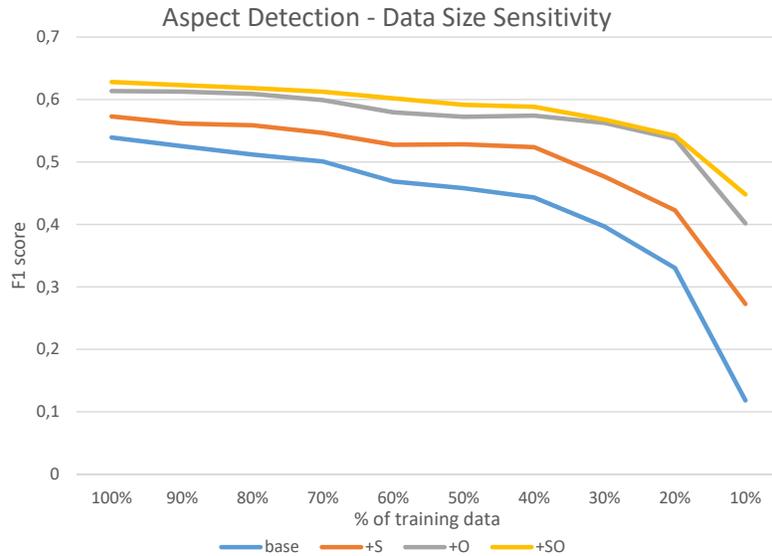


Fig. 5: The data size sensitivity for the aspect detection task

Table 4: Top 4 features for DRINKS#STYLE_OPTIONS according to weight assigned by SVM classifier

- | | |
|------------------------------------|---------------------------|
| 1. 0.369 - Ontology Concept: Menu | 3. 0.307 - Synset: list |
| 2. 0.356 - Ontology Concept: Drink | 4. 0.265 - Synset: enough |

the ontology-enhanced methods consistently outperform the other two methods, so the ontology features stay relevant even at smaller numbers of training data.

5.3 Feature Analysis

To investigate whether the ontology concepts are indeed useful for the SVM model, we take a look under the hood by investigating the internal weights assigned by the SVM to each feature. Since we use a binary classifier for each aspect category label, there are too many trained models to show these weights for all of them. Hence, we will only look at the top-weighted features for some of the more illustrative ones.

A very nice example is the trained SVM for the DRINKS#STYLE_OPTIONS category, which deals with the variety of the available drinks. The four most important features are listed in Table 4.

Clearly, the top two features perfectly describe this category, but the synsets for “list” and “enough”, as in “enough options on the list”, are also very appropriate. Another good example is DRINKS#PRICES, for which the top 10 features are shown in Table 5. Here, again, the top two concepts are typical of this aspect category. However, we see that certain lemmas and synsets are also high on the list, but with a lower weight. Note the word “at” which is often used to denote

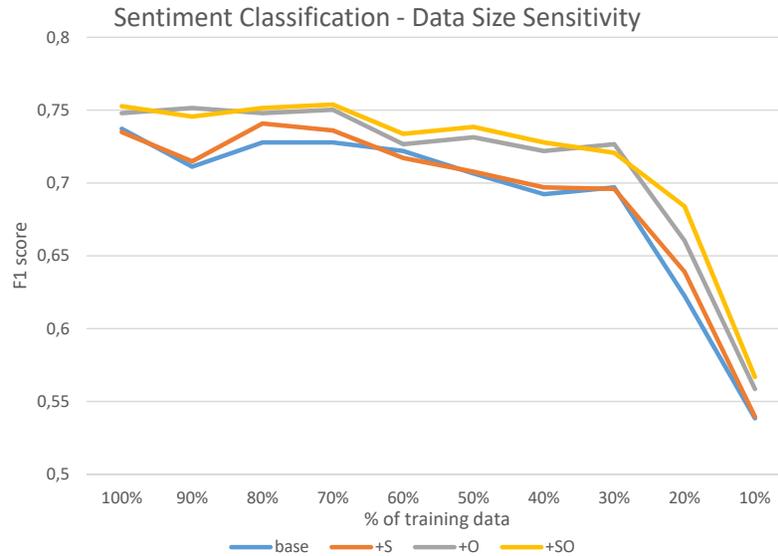


Fig. 6: The data size sensitivity for the aspect sentiment classification task (note that the y-axis does not start at 0 to improve readability)

Table 5: Top 10 features for DRINKS#PRICES according to weight assigned by SVM classifier

- | | |
|------------------------------------|-----------------------------------|
| 1. 0.428 - Ontology Concept: Price | 6. 0.180 - Lemma: price |
| 2. 0.303 - Ontology Concept: Drink | 7. 0.176 - Synset: wine |
| 3. 0.232 - Synset: drink | 8. 0.175 - Lemma: wine |
| 4. 0.204 - Lemma: drink | 9. 0.165 - Ontology Concept: Wine |
| 5. 0.184 - Lemma: at | 10. 0.157 - Synset: value |

a price, but which, being a function word, is not associated with a synset or an ontology concept.

An example where the ontology could make less of a difference is the category RESTAURANT#MISC, where, due to the miscellaneous nature of this aspect, no ontology concepts were applicable. Another category with interesting weights is FOOD#QUALITY, where next to the obvious ontology concept *Food*, a lot of adjectives were useful for the SVM since they convey the quality aspect. The fact that people write about quality is demonstrated by the strong use of sentiment words, such as “amazing” and “authentic”. Hence, it is rather difficult to define an ontology concept *Quality*, and while this concept is present in the ontology, it is not very useful, being ranked the 44th most useful feature here.

Looking at the RESTAURANT#GENERAL category, we find that some concepts are really missing from the ontology, namely the idea of someone liking a restaurant. This is often expressed as wanting to go back to that place, recommending it to others, or that it is worth a visit or worth the money. The top 10 features for this category are listed in Table 6 below to illustrate this.

Table 6: Top 10 features for RESTAURANT#GENERAL according to weight assigned by SVM classifier

1. 0.636 - Lemma: worth	6. 0.402 - Lemma: up
2. 0.505 - Lemma: love	7. 0.399 - Lemma: again
3. 0.487 - Lemma: back	8. 0.368 - Lemma: overall
4. 0.419 - Lemma: wrong	9. 0.360 - Lemma: favorite
5. 0.406 - Lemma: return	10. 0.355 - Lemma: recommend

At a first glance, the word “wrong” looks a bit out of place here, but at closer inspection of the data, it seems this word is often used in the phrase “you can’t go wrong ...”, which is indeed a positive remark about a restaurant in general.

For sentiment classification, a clear feature analysis is not very feasible, since the features are not binary, but are weighted according to distance and sentiment dictionary values, when applicable. Furthermore, the SVM model is trained on three sentiment values, which means that internally, a 1-vs-all strategy is performed, so the weights would be a combination of three trained models and not so descriptive anymore.

6 Conclusion

In this paper we presented a method for aspect-based sentiment analysis that utilizes domain ontology information, both for aspect detection and for sentiment analysis of these aspects. The performance of the SVM classifier is improved with this additional knowledge, although the improvement for aspect detection is more pronounced.

For aspect detection, it can indeed be concluded that fewer training data are needed, because performance drops much less when fewer training data are available compared to the same method without ontology features. This is not the case for the sentiment analysis method applied, where due to the fact that all methods use external resources in the form of sentiment dictionaries, the added value of the ontology remains limited. However, the ontology-enhanced method keeps outperforming the basic methods on the sentiment analysis task with about the same difference, even at smaller amounts of available training data.

When interpreting the internal weights assigned by the SVM, we see that for aspect detection the ontology features are in most cases the most informative. Only when the aspect categories themselves are not clearly defined (e.g., RESTAURANT#MISC), or when that category is not described in the ontology (e.g., RESTAURANT#GENERAL), do we see the SVM using mostly word-based features.

This leads us to conclude that ontology concepts are useful, especially for aspect detection, but also for sentiment analysis, and that ontology-enhanced aspect-based sentiment analysis is a promising direction for future research.

In terms of future work, we would suggest expanding the ontology by including more domain-specific sentiment expressions. Given the fact that there are more than three times as many target concepts in the current ontology than

sentiment expressions, focusing as much on sentiment expressions could lead to a more pronounced increase in performance on the sentiment analysis task as well. Furthermore, this process could be automated, scraping the information from the Web, where this type of data is readily available. Linking the ontology to others in the Linked Open Data cloud is also a direction that could be further explored.

While we propose methods for both aspect detection and sentiment analysis, there is still a subtask that is not yet covered, which is determining the target of an aspect within the sentence. Even though we use the target location information for the sentiment analysis task, we currently do not determine this information, predicting only the aspect category label. To create a complete method that can be deployed on real-life data, this missing link will need to be dealt with.

Acknowledgments

The authors of this paper are supported by the Dutch national program COMMIT.

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] Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [5] Dragoni, M., Tettamanzi, A., da Costa Pereira, C.: A Fuzzy System for Concept-Level Sentiment Analysis. In: Semantic Web Evaluation Challenge. pp. 21–27. Springer (2014)
- [6] Esuli, A., Sebastiani, F.: SentiWordNet: A publicly available lexical resource for opinion mining. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006). vol. 6, pp. 417–422. European Language Resources Association (ELRA) (2006)
- [7] Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)
- [8] Guarino, N., Welty, C.: Evaluating Ontological Decisions with OntoClean. Communications of the ACM 45(2), 61–65 (2002)
- [9] Jensen, A., Boss, N.: Textual Similarity: Comparing Texts in Order to Discover How Closely They Discuss the Same Topics. Bsc. thesis, Technical University of Denmark (2008), http://etd.dtu.dk/thesis/220969/bac08_15.pdf
- [10] JLanguageTool: wiki.languagetool.org/java-api (2016)

- [11] Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014. pp. 1746–1751. The Association for Computer Linguistics (2014)
- [12] Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). pp. 437–442. Association for Computational Linguistics and Dublin City University (2014)
- [13] Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proceedings of the Fifth Annual International Conference on Systems Documentation (SIGDOC 1986). pp. 24–26. ACM (1986)
- [14] Liu, B.: Sentiment Analysis and Opinion Mining, Synthesis Lectures on Human Language Technologies, vol. 16. Morgan & Claypool (2012)
- [15] 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 Association for Computational Linguistics: System Demonstrations. pp. 55–60. Association for Computational Linguistics (2014)
- [16] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
- [17] 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. Association for Computational Linguistics (2015)
- [18] Reforgiate Recupero, D., Presutti, V., Consoli, S., Gangemi, A., A.G., N.: Sentilo: Frame-Based Sentiment Analysis. *Cognitive Computing* 7(2), 211–225 (2015)
- [19] Saias, J.: Sentiue: Target and aspect based sentiment analysis in semeval-2015 task 12. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). pp. 767–771. Association for Computational Linguistics (2015)
- [20] Schouten, K., Frasincar, F.: Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* 28(3), 813–830 (2016)
- [21] 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. Association for Computational Linguistics (2013)
- [22] Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015. pp. 1556–1566. The Association for Computer Linguistics (2015)
- [23] Toh, Z., Su, J.: Nlangp: Supervised machine learning system for aspect category classification and opinion target extraction. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). pp. 496–501. Association for Computational Linguistics (2015)
- [24] 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)