SASOBUS: Semi-automatic Sentiment Domain Ontology Building Using Synsets

Ewelina Dera, Flavius Frasincar$^{(\boxtimes)}$, Kim Schouten, and Lisa Zhuang

Erasmus University Rotterdam,
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
ewelina.dera@live.com, {frasincar,schouten}@ese.eur.nl,
lisa.zhuang@hotmail.com

Abstract. In this paper, a semi-automatic approach for building a sentiment domain ontology is proposed. Differently than other methods, this research makes use of synsets in term extraction, concept formation, and concept subsumption. Using several state-of-the-art hybrid aspect-based sentiment analysis methods like Ont + CABASC and Ont + LCR-Rot-hop on a standard dataset, the accuracies obtained using the semi-automatically built ontology as compared to the manually built one, are slightly lower (from approximately 87% to 84%). However, the user time needed for building the ontology is reduced by more than half (from 7 h to 3 h), thus showing the usefulness of this work. This is particularly useful for domains for which sentiment ontologies are not yet available.

Keywords: Semi-automatic ontology building · Sentiment domain ontology · Aspect-based sentiment analysis

1 Introduction

With the growth of review data on the Web, as well as its importance, it is no wonder that 80% of consumers read online reviews and 75% of those people consider these reviews important [9]. Currently, the amount of online reviews, as well as other Web-based content, is tremendous. It is nearly impossible for a human to go through even a fraction of those reviews. As a result, it is not surprising that there was and still is, an increased interest in extracting, filtering and summarizing all the available reviews. Consequently, sentiment analysis and the more specific, aspect-based sentiment analysis (ABSA) [22] are very crucial and relevant tasks in the current business world.

This paper focuses on ABSA. ABSA is especially useful since in comparison with sentiment analysis, it gives more in-depth sentiment breakdown. There are many different approaches to conduct ABSA. However, there are two main types of methods, namely, knowledge representation (KR)-based and machine learning (ML)-based. Despite different advantages and disadvantages of those two methods, they both have a relatively good performance [22]. Nevertheless, a hybrid approach, combing ML with KR, was recently found to have an even

better performance than the two methods on their own [23]. Therefore, it is not surprising that many researchers tried combining these methods.

The authors of [25] proposed a hybrid model with better performance than other state-of-the-art approaches, including [17,23]. Therefore, [25] will be used as a base for this paper. The aim of this research is to further improve the performance of the methods proposed in [25] by enhancing the employed domain ontology. A new semi-automatic domain ontology is built based on synsets. Employing synsets instead of words should enable a fair and reliable comparison of words, while simultaneously capturing their meaning. Moreover, what is particularly worth emphasizing is the fact that as semi-automatic ontologies save considerable amounts of time, they are already considered to be successful if they have a similar performance to the manual ontologies. This is particularly true for new domains for which sentiment ontologies have not been yet devised.

There are many papers concerned with semi-automatic ontology building, e.g., [8,11,14]. However, these ontologies are neither sentiment ontologies nor are they built specifically for the task of ABSA. Furthermore, the majority of ontologies does not utilise synsets in any of the ontology building steps.

This paper follows with a review of the relevant literature in Sect. 2. Then, in Sect. 3 the used data is briefly explained. Afterwards, in Sect. 4 the used methodology is described. Then, the paper follows with Sect. 5 where the obtained results are presented. This paper concludes with Sect. 6 giving conclusions and suggestions for future work.

2   Related Work

As mentioned in [2], the KR-based techniques for ABSA have a rather good performance if a few major difficulties are overcome. The performance of a KR mainly depends on the quality of the used resource. For it to be extensive, it would need to be built automatically. However, for it to also be precise, it would need to be created manually, thus taking significant amounts of time. Hence, semi-automatic ontologies seem to be the best solution, where automatically extracted information is curated by users. Moreover, ML approaches, such as SVMs or neural networks, have a relatively good performance on their own. Unfortunately, they also need a lot of training data in order to learn properly [2]. That is why hybrid approaches are a good option for ABSA or other text classification tasks [22]. Hybrid approaches combine KR with ML, thus also exploiting the strengths of each of these two methods.

Seeing the potential of ontology as a base model, the authors of [25] decided to implement hybrid approaches. The authors used the same data and the same domain sentiment ontology as in [23]. However, for the ML part they decided to replace the SVM from [23] with neural networks. First, they combined the ontology with the Content Attention Based Aspect-based Sentiment Classification (CABASC) model [13]. By using a context attention mechanism, the model is able to take into consideration correlations between words and, at the same time, also the words' order. Furthermore, the authors also combined the sentiment domain ontology with a Left-Center-Right (LCR) separated neural network

[27]. They used three different variants of the LCR model, namely, LCR-Rot, LCR-Rot-inv and LCR-Rot-hop. The first approach has a rotatory attention mechanism, which first finds the most indicative words from the left and right context. These are, in turn, used to find the most indicative words in the target phrase (for the considered aspect). The second model, LCR-Rot-inv, is very similar - the only difference is that it inverses the order of the rotatory attention mechanism. Finally, the third model just repeats the rotatory attention mechanism of LCR-Rot multiple times. With the above described approaches, the authors obtained an even better performance than [23], with Ont + LCR-Rot-hop having the highest (out-of-sample) accuracy equal to 88% for the 2016 dataset [25]. Furthermore, while not directly compared, based on the reported performance results, [25] also has better results than [17] on the very same dataset.

Based on [25] it can be seen that a sentiment domain ontology is a very useful tool for ABSA. The neural back-up models in the mentioned paper have a high performance and improving them would be a tedious and strenuous task that might give an improvement at the level of a fraction of a percent. Therefore, it is decided that the best way to further enhance the performance of hybrid models for ABSA, is to concentrate on the ontology. Any further improvements to the KR would only make it more reliable and thus, decrease the number of cases when the back-up model has to be used.

Regarding previous efforts in extracting information in relation to aspects and their associated sentiments from text we would like to mention the following works. First, there are works that exploit dependency relations and a sentiment lexicon for finding aspects and their associated sentiment in text [7,21]. Second, there are advanced solutions that make use of an argumentation framework [6] or the rhetorical structure of text [10] in conjunction with a sentiment lexicon for determining aspects and/or the sentiment associated to these. Nevertheless, these works adopt a linguistic approach and not a KR one as considered here.

3   Data

For the purpose of this paper, different datasets are used. The Yelp dataset is used as a domain corpus for building the ontology. It comes from the Yelp Dataset Challenge 2017[1] and it contains 5,001 restaurant-related reviews and 47,734 sentences in total. Except the text representing the opinion of the reviewer, each review also contains a star rating. This rating is represented by an integer value between zero and five.

In addition, some contrastive corpora are also used for ontology learning, namely, six popular and freely available English books obtained from Project Gutenberg[2] as text files. These books are first pre-processed. Each book goes through the NLP pipeline from Stanford CoreNLP 3.8.0[3] toolkit. The following

---

[1] https://www.yelp.nl/dataset/challenge.
[2] http://www.gutenberg.org/wiki/Main_Page.
[3] https://stanfordnlp.github.io/CoreNLP/.

steps are performed: sentence splitting, tokenization, lemmatization and part-of-speech (POS) tagging.

Our proposed approach builds in a semi-automatic manner a sentiment domain ontology that is tested using the methods from [25]. While [25] used two datasets, i.e., SemEval-2015 and SemEval-2016, we only evaluate the afore-mentioned methods with the SemEval-2016 dataset. There is no need for the SemEval-2015 data as it is contained in the SemEval-2016 dataset. In 2016 Task 5[4] of SemEval was performed on ABSA. The used dataset contains reviews regarding different domains. However, as [25] used only the restaurant domain, to enable a reliable comparison, we also only focus on the restaurant domain. The SemEval-2016 data is already split into training and test datasets. The former contains 350 reviews and the latter only 90 reviews. Each review consists of sentences. In total, there are 2,676 sentences (2000 in the training dataset and 676 in the test dataset) and each one of them holds one or more opinions. Each one of the 3,365 opinions has a target word, aspect, and sentiment polarity. The aspect name consists of an entity and an attribute separated by a hash symbol. Furthermore, in Fig. 1 an example sentence in XML format is given. Here, a target word is 'atmosphere' (spans from the '12'th character to the '22'nd character in the text), the aspect category is 'AMBIENCE#GENERAL', and the sentiment polarity is 'positive'. There might be cases where, e.g., word 'meat' implies a food aspect; this is an explicit aspect because it has a clear target word. Nevertheless, there are also situations, when there is no target word. For instance, the sentence 'everything was cooked well' also implies a food aspect but there is no clear target word. In order to stay consistent with [25], all opinions with implicit aspects are removed. Consequently, there are 2,529 opinions remaining (1,870 in the training dataset and 650 in the test dataset). Moreover, all the words are tokenized and lemmatized using the NLTK platform [1] and WordNet.

```
<Sentence id="en_BlueRibbonSushi_478218900:2">
    <Text>I liked the atmosphere very much but the food was not worth the price.</Text>
    <Opinions>
        <Opinion to="22" from="12" polarity="positive" category="AMBIENCE#GENERAL"
        target="atmosphere"/>
        <Opinion to="45" from="41" polarity="negative" category="FOOD#QUALITY" target="food"/>
        <Opinion to="45" from="41" polarity="negative" category="FOOD#PRICES" target="food"/>
    </Opinions>
</Sentence>
```

Fig. 1. An example sentence from the SemEval-2016 dataset.

Moreover, it can be seen from Fig. 2a that for both train and test data, the positive sentiment is most frequently expressed. It accounts for 65–70% of the cases. Negative sentiment is found considerably less (with frequency of 25–30%).

Furthermore, when it comes to the number of opinions expressed per sentence, it can be seen in Fig. 2b that almost all of the respondents have between 0–3 opinions in a sentence.
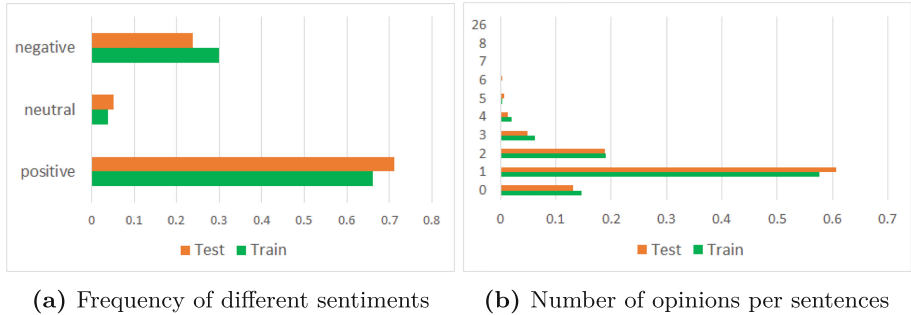

(a) Frequency of different sentiments   **(b)** Number of opinions per sentences

Fig. 2. Descriptive statistics for the SemEval-2016 dataset.

4   Methodology

All the text pre-processing and ontology learning is performed in the Semi-automatic Sentiment Domain Ontology Building Using Synsets (SASOBUS)[5] framework in Java. Furthermore, the HAABSA[6] framework in Python is used to evaluate the created ontology. Moreover, the Java API for WordNet Searching (JAWS)[7] library is used for obtaining synsets from WordNet.

In order to identify a sense of each word for both, the domain corpus and the contrastive corpora for ontology learning, the Simplified Lesk algorithm [12] is used. The reason behind such choice is that out of all the variants of the Lesk algorithm, this one has the best trade-off between accuracy and speed [5,12,24]. Besides, despite its simplicity it is hard to beat by other more advanced algorithms. The general idea behind the algorithm is that the ambiguous word and its context words are compared based on their glosses. The sense (or synset) having the highest overlap is returned by the algorithm.

4.1   Semi-automatic Ontology Learning

The approach chosen for the ontology building process is based on methods using ordinary words. However, these methods are modified in such a way that words are replaced with their corresponding synsets. Such an approach enables not only the comparison of the manually built ontology from [25] with the semi-automatically built ontology in this paper, but it also facilitates a comparison of two semi-automatically built ontologies: one with ordinary words and one

---

with synsets. Using synsets enables capturing the meaning of words better, thus enabling a more reliable comparison of words.

To the extent of authors' knowledge there is no research so far on ontology learning with synsets as terms. The term extraction method that is used has a score based on domain pertinence (DP) and domain consensus (DC) [16]. There are also other methods for term suggestion, such as, e.g., Term Frequency Inverse Document Frequency (TF-IDF) method based on frequency count [19]. In [3] the authors used TF-IDF and replaced terms by synsets, thus creating Synset Frequency Inverse Document Frequency (SF-IDF) method. The authors obtained better results with terms being synsets rather than ordinary words. Even though, [3] used SF-IDF for news item recommendation rather than for ABSA, there is still reason to believe that synsets as terms have a large potential, for instance, in other term extraction methods such as the DP and DC-based approach. These above mentioned reasons complement the motivation behind using synsets as terms not only for term extraction but also for the whole ontology building process.

Ontology Structure. The built ontology will have the same structure as in [23]. What is important to know is the fact that there are different types of sentiments. Type-1 sentiments are the words that have only one polarity, i.e., positive or negative, irrespective of the context and aspect. Type-2 sentiments are aspect-specific. These are words such as 'delicious' that can only relate to one aspect, i.e., sustenance in this case. When it comes to Type-3 sentiments, these words can have different polarity depending on the mentioned aspect. For instance, 'cold' combined with 'beer' has a positive sentiment, while 'cold' and 'soup' has a negative meaning.

Skeletal Ontology. The skeletal ontology contains two main classes, namely *Mention* and *Sentiment*. The first class encloses all the classes and concepts that represent the reviewed aspect, while the second one encompasses all concepts that relate to the sentiment polarity. The *Mention* class incorporates three subclasses: *ActionMention*, *EntityMention* and *PropertyMention*, which consist only of verbs, nouns and adjectives, respectively. The *Sentiment* class also has three subclasses: *Positive*, *Neutral* and *Negative*, which refer to the corresponding sentiment word. Each of the *Mention* classes has two subclasses called *GenericPositive<Type>* and *GenericNegative<Type>*. Type denotes one of the three types of mention classes, i.e., *Action*, *Entity* and *Property*. Those *Generic<Positive/Negative><Type>* classes are also subclasses of the corresponding *<Positive/Negative>* class.

The first performed step in ontology building is adding some general synsets representing words such as 'hate', 'love', 'good', 'bad', 'disappointment' and 'satisfaction' for each of the *GenericPositive<Type>* and *GenericNegative<Type>* classes. For each of those classes two general properties are added. Each word/synonym in a given synset is added to the concept as a *lex* property. Moreover, the synset ID is added as a *synset* property. However, to make the

name of the concept more human-readable and -understandable, the synset ID
is not used as the name. Instead the first word contained in the associated synset
denotes the name of the given concept. For instance, the synset 'verb@1778057'
is added as a (subclass) concept to *GenericNegativeAction*. All the synonyms
in this synset, i.e., 'hate' and 'detest' are added as a *lex* property, the ID is
added as a *synset* property and the name of this concept is the first synonym of
this synset, namely *Hate*. The synset ID has a format of POS@ID, where POS
stands for the part-of-speech tag and ID denotes the unique synset ID number
from WordNet. What is important to note is that '@' is replaced by '#' because
'@' has its own meaning in the RDFS language used for ontology building. An
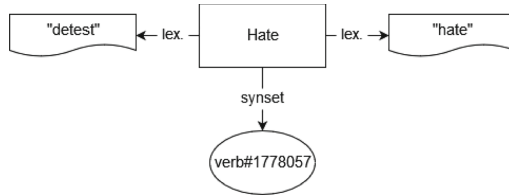example concept and its properties can be seen in Fig. 3.

Fig. 3. An example concept from the ontology.

Furthermore, as it was already mentioned in Sect. 3, each aspect has the
format of ENTITY#ATTRIBUTE. In this context 'entity' just represents a cer-
tain category for a particular aspect. Moreover, as it was also already mentioned
ActionMention, *EntityMention* and *PropertyMention* classes can only consist of
(concepts with) verbs, nouns and adjectives, respectively. Consequently, 'entity'
in *EntityMention* means noun. In order not to confuse those two meanings of
'entity', i.e., category or noun, from now on, an aspect has the format of CAT-
EGORY#ATTRIBUTE and it consists of a category and attribute. In other
words, word 'entity' is replaced with 'category' for this particular context.

The next step in the ontology building process is adding all the classes repre-
senting different aspects to the ontology. Just as in [23], for each *<Type>Mention*
class, a set of subclasses is added, namely, all the possible *<Category><Type>-
Mention* and *<Attribute><Type>Mention* classes are added. For *Attribute* there
are only three possible choices, i.e., prices, quality and style&options. Gen-
eral and miscellaneous attributes are skipped as, e.g., *MiscellaneousEntityMen-
tion* would be a too generic class. However, what is worth noting is the fact
that *Food<Type>Mention* and *Drinks<Type>Mention* classes are not added
directly as children of the respective *<Type>Mention* class. Just as in [23],
these classes have a parent called *Sustenance<Type>Mention*, which in turn
has *<Type>Mention* as a parent.

In the next step, for each *<Category/Attribute><Type>Mention* class, two new subclasses are created, namely *<Category/Attribute>Positive<Type>* and *<Category/Attribute>Negative<Type>* class. These classes also have the respective *Positive* or *Negative* classes as parents. An example with just a few possible classes can be seen in Fig. 4. It can be seen there that, e.g., *PropertyMention* has two subclasses, namely *ServicePropertyMention* (which represents one of the *<Category>PropertyMention* classes) and *PricesPropertyMention* (which represents one of the *<Attribute>PropertyMention* classes). Furthermore, *ServicePropertyMention* has two children: *ServicePositiveProperty* and *ServiceNegativeProperty*. These classes also have another parent: *Positive* and *Negative*, respectively. The situation is the same for all the remaining categories, attributes and types.

Furthermore, each of the discussed *<Category/Attribute><Type>Mention* classes has a *synset* property (with the synset ID), *lex* properties (with the synonyms from a given synset) and *aspect* properties. The last property has the format of CATEGORY#ATTRIBUTE. For each class, all the aspects that contain a certain category or attribute (as given in the class name) are added as the *aspect* property. For instance, in Fig. 5 there is a *LocationEntityMention* class. *Location* is a category so all the possible aspects that contain this category are added as an *aspect* property. Furthermore, location has a meaning of 'a determination of the place where something is'[8] so the corresponding synset 'noun@27167' is added as a *synset* property. All of the synonyms in this synset, i.e., 'location', 'localization' and 'localisation' are added as lexicalisations with the *lex* property.

Additionally, what is also important to know is the fact that there is a *disjointWith* relation between all the *<Category/Attribute>Positive<Type>* and all the *<Category/Attribute>Negative<Type>* classes.

Term Selection. To extract useful terms, the relevance score from [16] is used. The first step of this method is related to finding terms that are relevant only for a particular domain but not for another (irrelevant) domain. The DP score is calculated the following way:

$$DP_D(t) = \frac{freq(t/D)}{max_i(freq(t/C_i))},$$

(1)

where $freq(t/D)$ denotes the frequency of term t in the domain corpus D and $freq(t/C_i)$ denotes the frequency of the same term t in the contrastive corpus C_i. Index i stands for a particular contrastive corpus [16].

Furthermore, another measure that forms the relevance score is DC, which is defined as the consensus of a term across the domain corpus. The DC score is calculated as follows [16]:

$$DC_D(t) = -\sum_{d \in D} n\_freq(t,d) \times log(n\_freq(t,d)),$$
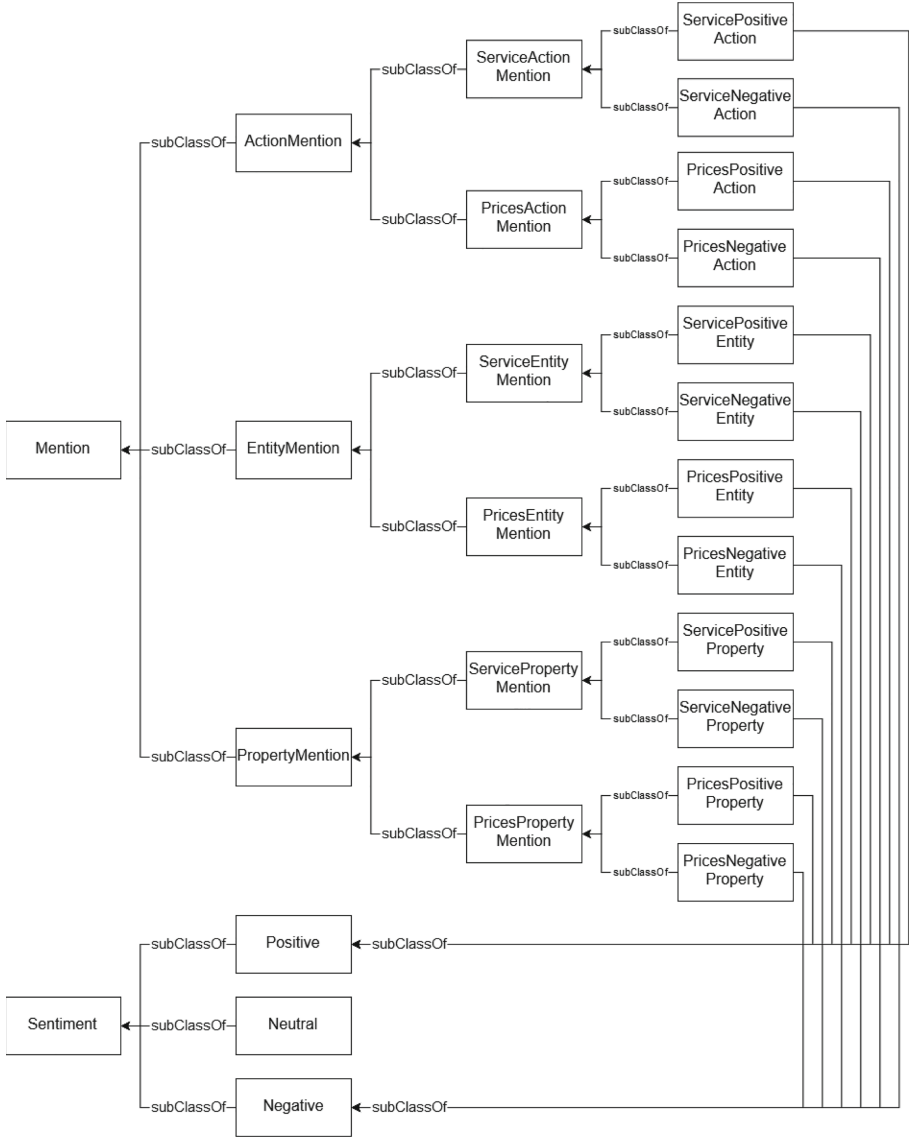
(2)

---

[8] https://wordnet.princeton.edu/.

Fig. 4. An excerpt from the ontology with a few example classes.

where $n\_freq(t, d)$, the normalized frequency of term t in document d is defined as follows:

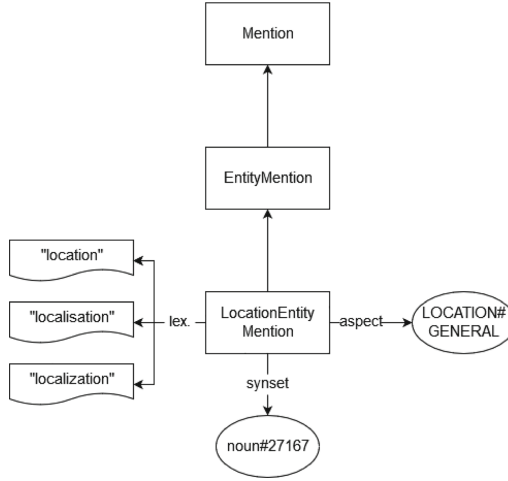$$n\_freq(t, d) = \frac{freq(t, d)}{max_{d \in D}(freq(t, d))}. \tag{3}$$

Fig. 5. A simplified example class from the ontology.

Ultimately, the final relevance score is defined as:

$$relevance\_score(t, D) = \alpha \frac{DP_D(t)}{max_t(DP_D(t))} + \beta \frac{DC_D(t)}{max_t(DC_D(t))}, \qquad (4)$$

where α and β are weights [16]. They are determined with a grid search algorithm. Furthermore, only a fraction of terms with the highest score is suggested to the user. These fractions are determined with the same algorithm.

However, terms are substituted by either synsets or lemma. If there exists a synset for a particular word, its frequency is calculated. Consequently, this frequency score is more reliable than just the word frequency. For instance, the noun 'service' has 15 possible senses in WordNet. With ordinary words as terms all the occurrences of this word in completely different contexts are counted together. With synset terms, however, these occurrences are context-dependent. Furthermore, if there is no synset for a word, it is replaced by its lemma. Consequently, in this paper a term is either a synset or a word.

Once all the frequencies and relevance scores are calculated, the (fraction of) extracted terms is suggested to the user. The user can reject the term or accept it. If it is the latter, then the user has to chose whether the term is an aspect concept or sentiment concept. The former just encompasses all the words/synsets relating to a certain aspect but with no polarity in their meaning. The latter are also aspect-related but they have a sentiment as well. For instance, 'sushi' is an aspect concept because it is related to an aspect, specifically to the food category. Also, 'yummy' is aspect-related; however, this word also carries a positive sentiment in its meaning. Therefore, it is a sentiment concept (related to the food aspect).

Hierarchical Relations. Hierarchical methods are derived with the subsumption method [20]. This method is based on co-occurrence and determines potential parents (subsumers) with the following formula:

$$P(x|y) \geq c, P(y|x) < c, \tag{5}$$

where c is a co-occurrence threshold, x is the potential parent and y the potential child [16,20]. In other words, the parent appears in more than a fraction of c documents, where the child also occurs, and the child occurs in less than a fraction of c documents, where the parent also occurs. Just as it was suggested in [20], c is replaced with a value of 1 in the second inequality, and set empirically to 0.2 in the first inequality.

Furthermore, multiple parents can be found by Eq. 5 so only one is chosen based on a parent score defined as:

$$parent\_score(x,y) = P(x|y). \tag{6}$$

All the potential parents are ranked by the score (from highest to lowest). The potential parents for verbs, nouns and adjectives that are aspect concepts are the respective $<Category/Attribute><Type>Mention$ classes. However, the potential parent classes for terms that are sentiment concepts are the corresponding $<Category/Attribute><Polarity><Type>$ classes. The *Polarity* here denotes the positive or negative sentiment of a concept.

Furthermore, an additional step to calculate the sentiment score of a given concept is added. We adapt the score from [4] as:

$$sentiment\_score(y) = \frac{\sum\limits_{d \in D} \left(rating(d) \times \frac{n(y,d)}{\sum\limits_{sent \in sentiments(D)} n(sent,d)} \right)}{\sum\limits_{d \in D} \frac{n(y,d)}{\sum\limits_{sent \in sentiments(D)} n(sent,d)}}, \tag{7}$$

where $rating(d)$ is a (Min-Max) normalized score of the Yelp star rating of a review d, $n(y,d)$ stands for the number of times concept y is used in review d and *sent* is a sentiment concept in $sentiments(D)$, the sentiment concepts in D. The polarity is negative if the score is smaller than 0.5, otherwise it is positive.

Consequently, as possible parent classes for aspect concepts are suggested based only on score from Eq. 6, there are two scores that are taken into account when suggesting possible parents for sentiment concepts. The score from Eq. 6 suggests a possible $<Category/Attribute><Polarity>Type$, while the score from Eq. 7 suggests a certain polarity value first. For instance, if Eq. 6 calculates the highest score for *FoodMention* class, Eq. 7 suggests a positive sentiment and the word form is verb then, *FoodPositiveAction* is suggested first, followed by *FoodNegativeAction*.

Additional Steps. What is worth being explicitly mentioned are the Type-3 sentiments. The proposed methods allow the user to accept multiple parents for a concept. Consequently, for instance, concept *Cheap* can have parents *PricesPositiveProperty* and *AmbienceNegativeProperty*.

4.2   Evaluation

In order to evaluate the quality of the created ontology, the same methods as used in [25] are utilised. The hybrid approach Ont + LCR-Rot-hop is performed with the manual ontology and the semi-automatic ontology for the SemEval-2016 dataset. This approach was chosen as it was found to have the best performance by the authors of [25]. Furthermore, similarly to [25] the Ont + CABASC approach is used as a baseline.

5   Results

This section provides all the results related to the ontology building process. First, the parameter optimisation results are described in Sect. 5.1. Then, the effectiveness of the semi-automatically built ontology is evaluated in Sect. 5.2 with three methods: Ont, Ont + LCR-Rot-hop and Ont + CABASC. Furthermore, each of these methods is evaluated with an in-sample, out-of-sample and average (based on 10-fold cross-validations) in-sample accuracy.

5.1   Parameter Optimisation

As it was already mentioned, the parameters α, β and the fraction of suggested verbs, nouns and adjectives were optimised. Let us call these ratios f_v, f_n and f_a, respectively. What is especially worth mentioning, is the fact that in Eq. 4 only the relative ratio between α and β is crucial. Consequently, the restriction of $\alpha + \beta = 1$ is imposed. Furthermore, an important goal of the ontology building process is to extract and suggest terms that the user accepts. Consequently, the grid search has the objective of maximising the term acceptance ratio. However, the user also does not want to go through all the possible terms. Therefore, to keep the number of suggested terms at a reasonable amount and to maximise the amount of accepted terms, the grid search for f_v, f_n, f_a and the respective values of α and β, maximises the harmonic mean between acceptance ratio and the amount of accepted terms. This mean is defined as:

$$objective_{pos} = \frac{2}{\frac{1}{acceptance\_ratio_{pos}} + \frac{1}{accepted\_terms_{pos}}}, \tag{8}$$

where *pos* stands for verbs, nouns and adjectives. The step size for the values of α and β is equal to 0.1 on a range from 0 to 1 and the step size for f_v, f_n and f_a is 0.01 on a range from 0.1 to 0.2 (due to the large number of terms). The resulting parameters can be seen in Table 1.

5.2   Ontology Building Evaluation

The number of added properties and classes in the built ontology can be seen in the left part of Table 2. Based on those numbers it can be observed that the ontology based on synsets as terms (sOnt) has more lexicalisations and classes.

Table 1. Table with the best performing parameters.

	α	β	f
Verbs (v)	1.0	0.0	0.10
Nouns (n)	0.5	0.5	0.11
Adjectives (a)	0.1	0.9	0.14

Furthermore, there are more *synset* properties than concepts, which means that there are some concepts that have more than one meaning. For instance, concept *Atmosphere* has two *synset* properties (and consequently, two meanings), namely, 'a particular environment or surrounding influence' and 'a distinctive but intangible quality surrounding a person or thing'. Moreover, sOnt does not have considerably more concepts than mOnt, however, the number of lexicalisations is significantly higher. While each concept in mOnt, on average, has one *lex* property, in sOnt there are, on average, three *lex* properties.

As can be seen in the right part of Table 2, the total time taken to create sOnt is higher than for mOnt. When it comes to the system time this is due to WSD. However, when it comes to the user time, it is lower by more than half when comparing sOnt to mOnt. In general, the system time cannot be reduced. However, when comparing the user time, it can be seen that sOnt takes considerably less time, while having substantially more concepts and lexicalisations.

Table 2. Table comparing some general statistics between different ontologies.

	lex property	*aspect* property	*synset* property	Classes	Concepts	System time	User time	Total time
sOnt	1324	45	483	558	456	5 h	3 h	8 h
mOnt	374	16	–	365	328	–	7 h	7 h

sOnt denotes the semi-automatic ontology based on synsets that was built using the methods from this paper and mOnt stands for the manual ontology from [23]. Furthermore, word 'classes' denotes here all the ontology classes, while 'concepts' stands for the classes containing the accepted terms.

The upper part of Table 3 shows the KR-based method's results. Unfortunately, sOnt has lower, both in-sample and out-of-sample accuracy than mOnt. However, this difference is rather small (only around 2%). Moreover, another semi-automatic ontology with words as terms (wOnt) has a slightly lower both in-sample and out-of-sample accuracy than sOnt. Therefore, the performance of sOnt is slightly worse than mOnt but it is simultaneously considerably better than a similar semi-automatic ontology but built on words rather than synsets.

Furthermore, it can also be seen in Table 3 that each hybrid method with sOnt has also around 2% lower performance. In addition, for both ontologies the benchmark approach (based on CABASC) has the worst performance when it comes to hybrid methods. The Ont + LCR-Rot-hop approach is significantly better than the benchmark, thus confirming the findings of [25].

Moreover, what is also interesting to see is that the benchmark approach, as well as the KR one has higher out-of-sample than in-sample accuracy for both

types of ontology. However, the LCR-Rot method has the accuracy values the other way around. In other words, the KR and the benchmark approach tend to underfit the data, while the LCR-Rot method rather leans towards overfitting.

Table 3. Table comparing the performance of different methods based on SemEval-2016 dataset with in-sample, out-of-sample and 10-fold cross-validation accuracy.

	Out-of-sample	In-sample	Cross-validation	
	Accuracy	Accuracy	Accuracy	St. dev.
mOnt	78.31%	75.31%	75.31%	0.0144
wOnt	72.80%	70.80%	70.90%	0.0504
sOnt	76.46%	73.92%	73.87%	0.0141
mOnt + CABASC	85.11%	82.73%	80.79%	0.0226
mOnt + LCR-Rot-hop	86.80%	88.21%	82.88%	0.0224
sOnt + CABASC	83.16%	79.53%	72.04%	0.1047
sOnt + LCR-Rot-hop	84.49%	86.07%	79.73%	0.0348

wOnt stands for a similar semi-automatic ontology built with the same methods as sOnt but with words (instead of synsets) as terms.

Each of the components used in the implementation is subject to errors due to various reasons. First, the proposed method depends on the domain corpus (as well as the contrastive corpora) for building the sentiment domain ontology (that affects both coverage and precision). Second, the method is sensitive to the errors made by the used NLP components. Given that we mainly use the Stanford CoreNLP 3.8.0 toolkit, which reports very good performance in the considered tasks [15], we expect the number of errors to be limited. The component that gives the largest number of errors is the word sense disambiguation implementation based on the Simplified Lesk algorithm, which obtained an accuracy of 67.2% on the SemCor 3.0 dataset [18] (word sense disambiguation is considered a hard task in natural language processing). Fortunately, some of the errors made by the implementation components can be corrected by the user as we chose for a semi-automatic approach instead of a fully automatic one.

6   Conclusion

This paper's aim was to propose a semi-automatic approach for ontology construction in ABSA. The main focus was on exploiting synsets for term extraction, concept formation and concept subsumption during the ontology learning process. A new semi-automatic ontology was built with synsets as terms. Its accuracy was slightly lower (about 2%) than the accuracy of the manual ontology but the user time is significantly lower (about halved). This result is particularly useful for new domains for which a sentiment ontology has not been devised yet. It can be concluded that the created ontology is successful. It can also be stated

that employing synsets in the term extraction, concept formation and taxonomy building steps of the ontology learning process results in better performance than just employing words. As future work it is planned to apply the proposed approach to other domains than restaurants, e.g., laptops. Also, it is desired to experiment with alternative methods to build the concept hierarchy, for instance like the one proposed in [26].

References

1. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python, Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Inc., Sebastopol (2009)
2. Cambria, E.: Affective computing and sentiment analysis. IEEE Intell. Syst. **31**(2), 102–107 (2016)
3. Capelle, M., Frasincar, F., Moerland, M., Hogenboom, F.: Semantics-based news recommendation. In: 2nd International Conference on Web Intelligence, Mining and Semantics (WIMS 2012), p. 27. ACM (2012)
4. Cesarano, C., Dorr, B., Picariello, A., Reforgiato, D., Sagoff, A., Subrahmanian, V.: OASYS: an opinion analysis system. In: AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs (CAAW 2006), pp. 21–26. AAAI Press (2006)
5. Craggs, D.J.: An analysis and comparison of predominant word sense disambiguation algorithms. Edith Cowan University (2011)
6. Dragoni, M., da Costa Pereira, C., Tettamanzi, A.G.B., Villata, S.: Combining argumentation and aspect-based opinion mining: the SMACk system. AI Commun. **31**(1), 75–95 (2018)
7. Federici, M., Dragoni, M.: A knowledge-based approach for aspect-based opinion mining. In: Sack, H., Dietze, S., Tordai, A., Lange, C. (eds.) SemWebEval 2016. CCIS, vol. 641, pp. 141–152. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46565-4_11
8. Fortuna, B., Mladenič, D., Grobelnik, M.: Semi-automatic construction of topic ontologies. In: Ackermann, M., et al. (eds.) EWMF/KDO 2005. LNCS (LNAI), vol. 4289, pp. 121–131. Springer, Heidelberg (2006). https://doi.org/10.1007/11908678_8
9. Gretzel, U., Yoo, K.H.: Use and impact of online travel reviews. In: O'Connor, P., Hopken, W., Gretzel, U. (eds.) Information and Communication Technologies in Tourism 2008. Springer, Vienna (2008). https://doi.org/10.1007/978-3-211-77280-5_4
10. Hoogervorst, R., et al.: Aspect-based sentiment analysis on the web using rhetorical structure theory. In: Bozzon, A., Cudre-Maroux, P., Pautasso, C. (eds.) ICWE 2016. LNCS, vol. 9671, pp. 317–334. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-38791-8_18
11. Kietz, J.U., Maedche, A., Volz, R.: A method for semi-automatic ontology acquisition from a corporate intranet. In: 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2000) (2000)
12. Kilgarriff, A., Rosenzweig, J.: English senseval: report and results. In: 2nd International Conference on Language Resources and Evaluation (LREC 2000). ELRA (2000)
13. Liu, Q., Zhang, H., Zeng, Y., Huang, Z., Wu, Z.: Content attention model for aspect based sentiment analysis. In: 27th International World Wide Web Conference (WWW 2018), pp. 1023–1032. ACM (2018)

14. Maedche, A., Staab, S.: Semi-automatic engineering of ontologies from text. In: 12th International Conference on Software Engineering and Knowledge Engineering (SEKE 2000), pp. 231–239 (2000)
15. Manning, C.D.: Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In: Gelbukh, A.F. (ed.) CICLing 2011. LNCS, vol. 6608, pp. 171–189. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19400-9_14
16. Meijer, K., Frasincar, F., Hogenboom, F.: A semantic approach for extracting domain taxonomies from text. Decis. Support Syst. **62**, 78–93 (2014)
17. Meskele, D., Frasincar, F.: ALDONA: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalised domain ontology and a neural attention model. In: 34th Symposium on Applied Computing (SAC 2019), pp. 2489–2496. ACM (2019)
18. Mihalcea, R.: SemCor 3.0 (2019). https://web.eecs.umich.edu/~mihalcea/downloads.html#semcor
19. Ramos, J., et al.: Using TF-IDF to determine word relevance in document queries. In: 1st instructional Conference on Machine Learning (iCML 2003), vol. 242, pp. 133–142 (2003)
20. Sanderson, M., Croft, W.B.: Deriving concept hierarchies from text. In: 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999), pp. 206–213. ACM (1999)
21. Schouten, K., Frasincar, F.: The benefit of concept-based features for sentiment analysis. In: Gandon, F., Cabrio, E., Stankovic, M., Zimmermann, A. (eds.) SemWebEval 2015. CCIS, vol. 548, pp. 223–233. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25518-7_19
22. Schouten, K., Frasincar, F.: Survey on aspect-level sentiment analysis. IEEE Trans. Knowl. Data Eng. **28**(3), 813–830 (2016)
23. Schouten, K., Frasincar, F.: Ontology-driven sentiment analysis of product and service aspects. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 608–623. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_39
24. Vasilescu, F., Langlais, P., Lapalme, G.: Evaluating variants of the lesk approach for disambiguating words. In: 4th International Conference on Language Resources and Evaluation (LREC 2004). ELRA (2004)
25. Wallaart, O., Frasincar, F.: A hybrid approach for aspect-based sentiment analysis using a lexicalized domain ontology and attentional neural models. In: Hitzler, P., et al. (eds.) ESWC 2019. LNCS, vol. 11503, pp. 363–378. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21348-0_24
26. Zafar, B., Cochez, M., Qamar, U.: Using distributional semantics for automatic taxonomy induction. In: 14th International Conference on Frontiers of Information Technology (FIT 2016), pp. 348–353. IEEE (2016)
27. Zheng, S., Xia, R.: Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention. arXiv preprint arXiv:1802.00892 (2018)