# A Hybrid Approach for Aspect-Based Sentiment Analysis Using a Lexicalized Domain Ontology and Attentional Neural Models

Olaf Wallaart and Flavius Frasincar(✉)

Erasmus University Rotterdam,
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
olafwallaart@student.eur.nl, frasincar@ese.eur.nl

**Abstract.** This work focuses on sentence-level aspect-based sentiment analysis for restaurant reviews. A two-stage sentiment analysis algorithm is proposed. In this method, first a lexicalized domain ontology is used to predict the sentiment and as a back-up algorithm a neural network with a rotatory attention mechanism (LCR-Rot) is utilized. Furthermore, two features are added to the backup algorithm. The first extension changes the order in which the rotatory attention mechanism operates (LCR-Rot-inv). The second extension runs over the rotatory attention mechanism for multiple iterations (LCR-Rot-hop). Using the SemEval-2015 and SemEval-2016 data, we conclude that the two-stage method outperforms the baseline methods, albeit with a small percentage. Moreover, we find that the method where we iterate multiple times over a rotatory attention mechanism has the best performance.

## 1 Introduction

Since the enormous increase of unstructured review data on the Web, the interest in sentiment analysis [8] has risen as well. The main goal of sentiment analysis is to extract the sentiment and opinions of content creators and combine this information into useful results for companies, researchers, and users. Where for small companies it might be manually possible to obtain their costumers' opinion, this task becomes labor-intensive and time-expensive for large companies. Hence, sentiment analysis can be a useful tool. A subtask of sentiment analysis is aspect-based sentiment analysis [15]. Here, instead of computing a sentiment score (usually positive or negative) for the entire review or sentence, the task is to identify different aspects or characteristics in the review and compute the sentiment of the reviewer towards these specific aspects.

The main task at hand is to create a method that accurately and efficiently predicts the sentiment about a given aspect. Methods can generally be classified as knowledge-based or as machine learning approaches [15]. Both methods have their strong and weak points and recent research shows that the two types of


P. Hitzler et al. (Eds.): ESWC 2019, LNCS 11503, pp. 363–378, 2019.
https://doi.org/10.1007/978-3-030-21348-0_24

methods are complementary to each other [16]. It is shown that a hybrid method, using both statistical learning and a knowledge-based approach, outperforms many of the existing methods that use one type of method [16].

The problem still at hand is what combination of these types of methods will yield the best performance, both in terms of accuracy and efficiency. This work will extend and try to improve the results obtained in [16] by implementing state-of-the-art machine learning methods. Precisely we will make use of a combination between available domain knowledge, in the form of an ontology as described in [16] and a neural rotatory attention model as introduced in [21]. In addition, we investigate two extensions of the used neural network: changing the order in the rotary attention mechanism and iterating several times over the rotary attention mechanism.

The paper is organized as follows. In Sect. 2 we provide and discuss the relevant literature that is available with respect to sentiment analysis using an ontology and neural models. Section 3 gives an overview and explanation of the data used in this work. Next, a description of the proposed framework is given in Sect. 4 and results of the proposed methods are evaluated in Sect. 5. Finally, in Sect. 6 conclusions are made and suggestions for future research are provided. The source code used to implement our methods is written in `Python` and can be found at https://github.com/ofwallaart/HAABSA.

## 2 Related Work

In [15] an overview of aspect-based sentiment analysis is given. The goal of aspect-based sentiment analysis is to find the sentiment of a group of people with regard to a certain topic. A sentiment or aspect can be mentioned explicitly but also remain implicit. For instance the sentence 'You can't go wrong here' has an implicit aspect and an explicit sentiment, since the aspect is not literally in the text but a positive sentiment can be derived from the sentence directly. In this work, we will ignore implicitly mentioned aspects since the methods proposed rely on the presence of predefined aspects. Implicitly mentioned sentiments are taken into consideration since they pose no problem for the proposed methods. There are three different categories of algorithms for aspect-based sentiment analysis [15]: knowledge-based approaches, machine learning approaches, and hybrid approaches.

Knowledge-based algorithms often use a sentiment dictionary that finds a sentiment score for a specific word. Subsequently, these sentiment scores are combined by a method to determine the sentiment of all the relevant words with respect to an aspect [15]. Using the SenticNet knowledge base, in [5] a polarity detection method is developed by means of sentic patterns. Sentic patterns are linguistic patterns which allow sentiments to flow from concept to concept based on the dependency relation of the input sentence. First, each sentence is processed to find the expressed concepts. The discovered concepts are then linked to the SenticNet knowledge base by using sentic patterns to make an inference of the sentiment value linked to the sentence.

In [16], a common domain knowledge in the form of an ontology is used to construct a knowledge-based method. Sentiment-indicating words are split into three types. The first type contains expressions that, regardless of the aspect, always indicate the same sentiment. The second type of sentiment expressions only belong to one specific aspect category, whereas the third type of words are words of which the expressed sentiment depends on their context. The ontology approach used in this work will be similar to this procedure, due to the high performance of this solution [16].

The general increase in interest of machine learning methods such as neural networks has also caused an increase of their usage for the purpose of sentiment analysis. Especially the usage of neural attention models has been a field of high interest lately [6,20]. [9] uses a neural attention model with an attention mechanism that can enforce a model to pay more attention to the important parts of a sentence. The mechanism is able to focus on a certain region with 'high attention' while perceiving the surrounding with 'low attention' and adjusting the focal point over time [20].

The model introduced in [9], is a so-called Content Attention Based Aspect-based Sentiment Classification (CABASC) model. It uses a weighted memory module by introducing a context attention mechanism in the model that is responsible for simultaneously taking the order of the words and their correlations into account. When considering the SemEval-2014 data [14], it is shown that CABASC outperforms widely-known methods, such as a support vector machine (SVM) and a Long Short-Term Memory model (LSTM). CABASC even outperforms RAN, which is a state-of-the-art model that uses a multi-hop attention mechanism, a deep bidirectional LSTM, and a position attention mechanism to provide tailor-made memories for different aspects in a sentence [6].

In [21] an approach called a Left-Center-Right separated neural network with Rotatory attention (LCR-Rot) is proposed. This model is able to better represent the sentiment aspect, especially when the aspect contains multiple words, and improve the interaction between aspect and left/right contexts to capture the most important words in the aspect and context. The model consists of three separated LSTMs with 300 hidden units each, corresponding to the three parts of a review (left context, target phrase, and right context). Furthermore, it uses a rotatory attention mechanism which models the relation between aspect and left/right contexts. This is done by letting the left/right context and target phrase both use information of the other part to capture indicative sentiment words. Obtained results again show an improved performance over an SVM and an LSTM model. Furthermore, results indicate that LCR-Rot also outperforms CABASC. However, a direct comparison between the two models has not yet been made and would give a more profound conclusion on which model performs better. This paper aims to also answer this question by directly comparing these two methods.

The proposed idea of using a hybrid method, combining a knowledge-driven approach and a statistical method has been marked as the most promising way to improve the effectiveness of affective computing, which includes sentiment analysis [4]. [16] propose two hybrid methods for aspect-based sentiment analysis where their previously mentioned ontology-based approach is used as a knowledge-based method. As a machine learning approach, a bag-of-words model combined with an SVM classifier is used. The first approach proposed by [16] uses this bag-of-words model with the addition of two binary features that indicate the sentiment prediction from the ontology method. The second approach uses a two-step procedure where first the ontology method is used to predict a sentiment and if this method is not able to make a prediction, the bag-of-words model is used as a backup method. The latter model performs the best and has a state-of-the-art performance of 85.7% on the SemEval-2016 [13] data. This two-step approach is superior to the global vector approach as proposed by [17], which also combines ontology and non-ontology-based features.

## 3 Specification of Data and Tasks

SemEval (Semantic Evaluation) is a widely used evaluation method for computational semantic analysis systems [9,16,21]. In this paper we use the SemEval-2015 Task 12 [12] and SemEval-2016 Task 5 Subtask 1 [13] datasets to train and evaluate our models. By using these datasets we are able to compare our outcomes with that of other methods using the same datasets and thus making it a convenient choice.

The datasets consist of restaurant reviews with one or multiple sentences. These sentences contain one or multiple opinions about a certain aspect category. Such an opinion consists of the aspect category where an opinion is given about and the actual aspect. In the dataset this aspect is marked as `target` and refers to the word or words in the sentence that is/are opinionated with respect to an aspect category. In addition, for each opinion a polarity is given that expresses whether the reviewer is positive, negative, or neutral towards a specific aspect. In Fig. 1 a sentence from the SemEval-2016 dataset in the XML markup language is shown as an example. The dataset consists of training data, that will be used

```
<sentence id="430342:0">
  <text>delicious simple food in nice outdoor atmosphere.</text>
  <Opinions>
      <Opinion from="17" to="21" polarity="positive" category="FOOD#
          QUALITY" target="food"/>
      <Opinion from="17" to="21" polarity="positive" category="FOOD#
          STYLE_OPTIONS" target="food"/>
      <Opinion from="30" to="48" polarity="positive" category="AMBIENCE#
          GENERAL" target="outdoor atmosphere"/>
  </Opinions>
</sentence>
```

**Fig. 1.** A sentence from the SemEval-2016 data set.

**Table 1.** Polarity frequencies in the data set.

| | Negative | | Neutral | | Positive | | Total | |
|---|---|---|---|---|---|---|---|---|
| | Freq. | % | Freq. | % | Freq. | % | Freq. | % |
| SemEval-2016 train data | 488 | 26.0 | 72 | 3.8 | 1319 | 70.2 | 1879 | 100 |
| SemEval-2016 test data | 135 | 20.8 | 32 | 4.9 | 483 | 74.3 | 650 | 100 |

to train our proposed machine learning methods, and test data that is used to evaluate our methods.

The obtained XML data is preprocessed so that it can be used efficiently by our algorithms. To make our results as verifiable and as comparable as possible we will use similar procedures as in [20] and [21]. First of all, all opinions where the aspect is implicit are removed from the dataset. The remaining sentences are then processed using the NLTK platform [3]. The data is tokenized and all words are lemmatized using the WordNet lexical database [10].

The word embedding vectors used in this paper will have a dimension (vector size) of 300. In theory higher dimensions can store more information and perform better. In practice however, the benefit from vectors with a dimensionality higher than 300 is small [11]. We use a pre-trained word vector vocabulary from the GloVe framework [11], with a vocabulary size of 1.9 million words. We choose GloVe because it is superior to CBOW and skip-gram methods [11]. Words that do not appear in the GloVe vocabulary are randomly initialized by a normal distribution $N(0, 0.05^2)$ as in [21].

Since the SemEval-2015 data is contained in the SemEval-2016 data, it has similar properties and therefore we will not discuss it separately and only provide insights for the SemEval-2016 data. Table 1 shows how the opinions are distributed after preprocessing, considering the opinion polarities. The majority of the opinions, around 70% have a 'positive' polarity. From the relative frequencies it can be observed that, with respect to polarity frequencies, the test and training datasets are similar.

# 4 Method

In this section we present the used methods in this research. First, in Sect. 4.1 we describe the ontology-based approach. Then, in Sect. 4.2 we depict the employed neural attention model. Last, in Sects. 4.3, 4.4, and 4.5 we give our proposed models that build on the previous ones.

## 4.1 Ontology-Based Approach

By employing an ontology approach for aspect-based sentiment analysis we are able to predict sentiment by using predefined classes, relations between these classes, and axioms that entail either a positive or negative opinion in a sentence. The approach used in this paper is similar to the ontology reasoning proposed

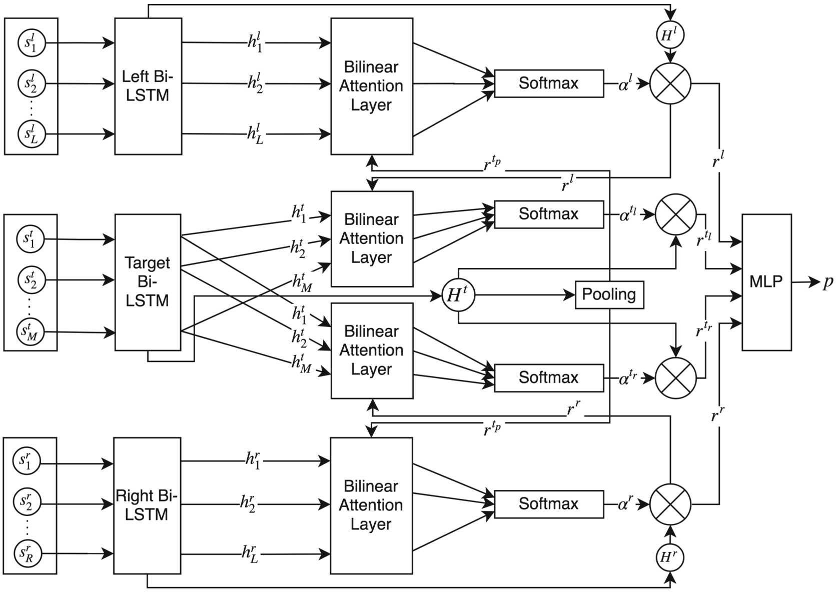



**Fig. 2.** Visual representation of the LCR-Rot models architecture.

by [16]. Note that the ontology does not contain any classes or relations for the neutral sentiment. Hence, the only possible outcomes are either positive or negative.

The ontology consists of three main classes: *SentimentValue*, *AspectMention*, and *SentimentMention*. *SentimentValue* contains the subclasses `Positive` and `Negative` and simply assigns these classes to, respectively, positive and negative expressions. The *AspectMention* class describes the mentions of aspects by linking the lexical representation of a word to the corresponding concept in the ontology. For example, the word 'atmosphere' is linked to the concept `Ambience` by the following axiom: `Ambience` $\equiv\ \exists\, lex.\{$'atmosphere'$\}$.

The *SentimentMention* class models the expressions of sentiment and is divided into three types of subclasses. A type-1 *SentimentMention* contains concepts that express the same sentiment for every aspect. For instance, the word 'good' always implies a positive expression and does not depend on the given aspect. Type-2 concepts contain expressions that always have the same sentiment but only belong to a unique set of aspects. The word 'delicious' is a good example since it always expresses a positive sentiment towards the aspect categories `FOOD#QUALITY` and `DRINKS#QUALITY`, but does not apply towards other categories and is therefore ignored towards these other aspects. The final, type-3 *SentimentMention* class contains expressions that do not belong to any of the above mentioned types because the sentiment of the expression depends on the context around the aspect. For instance the expression 'cheap' is positive when combined with the word 'price' but is negative with regard to 'atmosphere'.

### 4.2 Left-Center-Right Separated Neural Network with Rotatory Attention

The LCR-Rot base model used in this paper is described by [21], and we propose two extensions and compare these extensions with the base model. The architecture of the LCR-Rot model is illustrated in Fig. 2.

We define $S = [s_1, s_2, \ldots, s_N]$ as a sentence that contains $N$ words. We first split this sentence into three parts, namely: left context $[s_1^l, s_2^l, \ldots, s_L^l]$, target phrase $[s_1^t, s_2^t, \ldots, s_M^t]$ (the relevant aspect), and right context $[s_1^r, s_2^r, \ldots, s_R^r]$. $L, M, R$ are the lengths of the three parts, respectively.

Next, we build the model by adding three bi-directional long-short-term-memory (Bi-LSTM) modules with 300 hidden units each. Namely, a left-, center-, and right-Bi-LSTM that respectively model left context, target phrase, and right context in the sentence. LSTMs are specialized in remembering information for a long period of time. Moreover, the bidirectional property keeps the contextual information in both directions. The input of each Bi-LSTM is a word embedding representation of the words for that specific part. We use the GloVe vocabulary where all the words are represented in a matrix $K \in \mathbb{R}^{d \times |V|}$, where $d$ is the dimension of the word embedding and $|V|$ is the total vocabulary size. After using the initial word embeddings as an input, the Bi-LSTMs give back hidden states $[h_1^l, h_2^l, \ldots, h_L^l]$ for left context, $[h_1^t, h_2^t, \ldots, h_t^l]$ for target phrase, and $[h_1^r, h_2^r, \ldots, h_R^r]$ for right context as initial representations.

Next we apply a rotatory attention mechanism to the hidden state outputs of the Bi-LSTMs to capture the most indicative words in the left/right context and the target phrase. The mechanism is divided into two steps, where the first step will try to capture the most indicative words in the left/right context. In the second step the left/right representations from the previous step are used to capture the most indicative words in the target phrase. These four representations together form the final sentence representation.

**Step 1: Target2Context Attention Mechanism.** To obtain a better representation of the left and right contexts we use an average representation of the target phrase. To achieve this average target representation we use an average pooling layer, which works well in this context [1]:

$$\underset{2d\times 1}{r^{t_p}} = \text{pooling}([\underset{2d\times 1}{h_1^t}, \underset{2d\times 1}{h_2^t}, \ldots, \underset{2d\times 1}{h_M^t}]). \tag{1}$$

We then define an attention function $f$ to obtain a representation of the left and rights contexts. This is done by using the average target pooling $r^{t_p}$ and the hidden states of each word in the context. For example, when considering the left context, the hidden states are $h_i^l$, for $i = 1, \ldots, L$, and $f$ is defined as:

$$\underset{1\times 1}{f(h_i^l, r^{t_p})} = \tanh(\underset{1\times 2d}{h_i^{l\prime}} \times \underset{2d\times 2d}{W_c^l} \times \underset{2d\times 1}{r^{t_p}} + \underset{1\times 1}{b_c^l}), \tag{2}$$

where $W_c^l$ is a weight matrix and $b_c^l$ is a bias. The obtained context attention scores $f$ are then fed into a softmax function that scales the scores on an interval between 0 and 1. Taking again the left context as an example, normalized attention scores $\alpha_i^l$ are computed by:

$$\alpha_i^l = \frac{\exp(f(h_i^l, r^{t_p}))}{\sum_{j=1}^{L} \exp(f(h_j^l, r^{t_p}))}. \tag{3}$$

At last we retrieve a left context representation by taking a weighted combination of the word hidden states:

$$\underset{2d\times1}{r^l} = \sum_{i=1}^{L} \underset{1\times1}{\alpha_i^l} \times \underset{2d\times1}{h_i^l}. \tag{4}$$

By following Eqs. (2)–(4) in a similar way we can obtain $r^r$ for right context.

**Step 2: Context2Target Attention Mechanism.** The left and right context representations obtained in step 1 are now used to construct an improved representation of the target phrase. Again, we first define an attention function $f$ by using the context representations $r^l/r^r$ of the left and right contexts and the hidden states of each word in the target $h_i^t$, for $i = 1, \ldots, M$. If we take the left context as an example:

$$\underset{1\times1}{f(h_i^t, r^l)} = \tanh(\underset{1\times2d}{h_i^{t\prime}} \times \underset{2d\times2d}{W_t^l} \times \underset{2d\times1}{r^l} + \underset{1\times1}{b_t^l}), \tag{5}$$

where $W_t^l$ is a weight matrix and $b_t^l$ is a bias. The obtained target attention scores $f$ are then fed into a softmax function that scales the scores on an interval between 0 and 1. Normalized attention scores $\alpha_i^r$ are computed by:

$$\alpha_i^{t_l} = \frac{\exp(f(h_i^t, r^l))}{\sum_{j=1}^{M} \exp(f(h_j^t, r^l))}. \tag{6}$$

At last we retrieve a target phrase representation by taking a weighted combination of the word hidden states:

$$\underset{2d\times1}{r^{t_l}} = \sum_{i=1}^{M} \underset{1\times1}{\alpha_i^{t_l}} \times \underset{2d\times1}{h_i^t}, \tag{7}$$

which we call left-aware target representation, since it denotes the amount that words in the target phrase are influenced by the left context. By following Eqs. (5)–(7) in a similar way we can obtain the right-aware target representation, $r^{t_r}$.

**Sentiment Prediction.** The final representation for a sentence is acquired by concatenating the left-context representation $r^l$, right-context representation $r^r$, and both the two side-target representations, $r^{t_l}$ and $r^{t_r}$:

$$\underset{8d\times1}{v} = [\underset{2d\times1}{r^l}; \underset{2d\times1}{r^{t_l}}; \underset{2d\times1}{r^{t_r}}; \underset{2d\times1}{r^r}]. \tag{8}$$

The sentence representation vector is converted by a linear layer to compute the sentiment prediction vector $p$ of length $|C|$, where $C$ is the number of different sentiment categories. The vector is then fed into a softmax layer to predict the sentiment polarity of the target phrase:

$$\underset{|C|\times 1}{p} = \text{softmax}(\underset{|C|\times 8d}{W_c} \times \underset{8d\times 1}{v} + \underset{|C|\times 1}{b_c}), \tag{9}$$

where $p$ is a conditional probability distribution, $W_c$ is a weight matrix, and $b_c$ is a bias.

**Model Training.** The model is trained in a supervised manner by minimizing a cross-entropy loss function. The loss function is defined as:

$$\underset{1\times 1}{L} = -\sum_j \underset{|C|\times 1}{y_j} \times \log(\underset{|C|\times 1}{\hat{p_j}}) + \lambda\|\Theta\|^2, \tag{10}$$

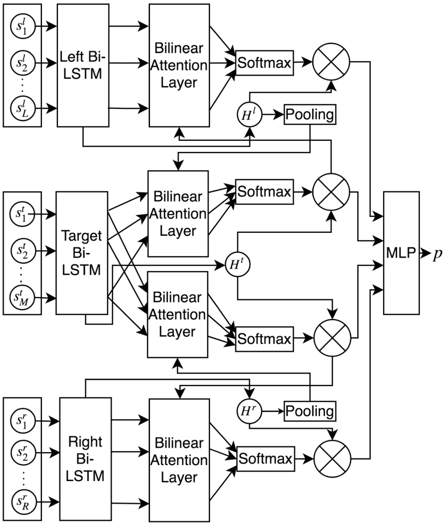
where $y_j$ is a vector that contains the true sentiment value for the $j$-th training opinion, $\hat{p}_j$ is a vector containing the predicted sentiment for the $j$-th training opinion, $\lambda$ is the weight of the $L_2$-regularization term, and $\Theta$ is the parameter set which contains $\{W_c^l, b_c^l, W_c^r, b_c^r, W_t^l, b_t^l, W_t^r, b_t^r, W_c, b_c\}$ and the LSTM parameters.

For loss minimization we use backward propagation where we initialize the weight matrices by a uniform distribution $U(-0.1, 0.1)$ and all bias are set to zero, as is done by [21]. To update the weights and biases we use stochastic gradient descent with momentum. Furthermore, the dropout technique is applied to all hidden layers to avoid overfitting [19]. The dropout technique randomly drops units from the neural network during training to prevent units from co-adapting too much on the training data.

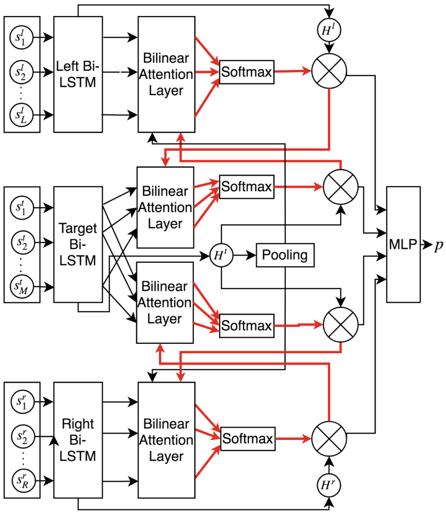
Before training, the required hyperparameters of the proposed models are tuned. Parameters that are tuned include the learning rate, the $L_2$-norm regularization term ($\lambda$ in Eq. 10), the dropout rate, and the momentum term. 80% of the training data is used for tuning and the other 20% is used for validation to test hyperparameter configurations. For a fast convergence speed we use a tree-structured Parzen estimators (TPE) algorithm [2] for tuning. TPE allows to learn from the training history and hence is able to give better estimations for the next set of parameters.

### 4.3 Two-Step Approach

The algorithm proposed in this paper will combine the ontology-based approach and LCR-Rot method into a hybrid method. The algorithm will iterate over all the opinions in the test dataset and predict the sentiment towards the aspects mentioned. The algorithm framework is explained in detail in [16] and hence we refer to this work for further details. In short, the algorithm will first use the previously described ontology to predict a positive or negative sentiment. If the ontology is not able to give a conclusive result, the algorithm will use a machine learning method (in our case the LCR-Rot algorithm) as a backup method.




(a) LCR-Rot-inv




(b) LCR-Rot-hop

**Fig. 3.** Visual representations of the LCR-Rot-inv and LCR-Rot-hop architectures.

## 4.4 Inversed LCR-Rot

A possible alteration of the LCR-Rot algorithm as described in Sect. 4.2 is to inverse the rotatory attention mechanism. We will call this method LCR-Rot-inv. Instead of first applying a target2context attention algorithm and subsequently applying a context2target attention algorithm, it is also possible to first perform the context2target algorithm and then use the target2context algorithm. Instead of only one target pooling layer (for the target sentence), now two context pooling layers (for the left and right context) are used. By altering the order of the rotatory attention mechanism, the algorithm might be able to give important words more weight and/or catch semantic relations better. An illustration of this altered model is provided in Fig. 3a.

## 4.5 Multi-hop LCR-Rot

Another possible alteration of the LCR-Rot method is to repeat the rotatory attention mechanism for $x$ times. This method is called LCR-Rot-hop since multiple hops are performed over the attention weights. By increasing the number of rotatory attention iterations, the model might be able to better represent aspects and contexts by improving the interaction between aspects and contexts. Furthermore, the iterative nature of the method might let the attention weights converge to their optimal values and achieve a higher accuracy. Figure 3b gives an illustration of the architecture of this method. The attention mechanism, that is iterated over for a certain amount of times, is marked with bold, red arrows. The main difference from the LCR-Rot method is that after step 2 the representation outputs are fed back into the rotatory attention mechanism.

## 5 Evaluation

To evaluate the performance of the models proposed in this paper against the baseline methods, training is performed on the training data and testing is done on the official test data. The evaluation metric is classification accuracy. Our models are compared to the following baseline methods:

**Ont** [16]**:** Uses a domain knowledge, as encoded in an ontology, to determine aspect sentiment.

**BoW** [16]**:** Bag-of-words model combined with an SVM classifier to determine sentiment. Hyperparameters $C$ and $gamma$ for the SVM are tuned in a similar way as described in Sect. 4.2 using a TPE algorithm.

**CABASC** [9]**:** Neural network that contains a context attention-based memory module for aspect-based sentiment analysis.

**Ont+BoW** [16]**:** Two step approach where an ontology method is first used and a backup bag-of-words method is used.

**Ont+CABASC:** Two step approach where first an ontology method is used [16] and as a backup method the CABASC model [9] is used.

In order to guarantee a fair comparison, we opt to program the baseline models instead of copying results reported in other papers. Since our code is written in `Python`, the Stanford CoreNLP package [18] that is used in [16] to find a sentence sentiment score for the BoW model is not available. Hence, we use the VADER sentiment score [7] as an alternative.

Accuracy results for the SemEval-2015 and SemEval-2016 dataset are given in Table 2. For each dataset, the first two columns show out-of-sample and in-sample results, respectively, of a single run. For this, the in-sample accuracy uses the complete training dataset and the out-of-sample accuracy uses the test dataset. The last two columns of each dataset present the average results of a 10-fold cross-validation procedure using the training data. For the LCR-Rot-hop model, preliminary results on the training data show that when the number of iterations $x$ is set to three attention rotations, the highest accuracy is achieved.

We conclude that the ontology method by itself does not perform well. This is not surprising since it is only able to make a prediction in around 60% of all sentiment opinions. In the other 40%, the majority class is predicted, which is not a very good predictor. Furthermore, we can conclude that, in the case of restaurant reviews, LCR-Rot indeed outperforms CABASC by 1.8%–2.3%. LCR-Rot-inv also outperforms CABASC but by a smaller percentage of 0.5%–1.9%. Remarkably for the SemEval-2016 dataset the LCR-Rot-hop method performs around 0.8% better then the LCR-Rot method. This improved performance is however not apparent in the 2015 dataset, here the two methods perform equally well. Hence, LCR-Rot-hop is the best performing machine learning method for this specific task. By inversing the rotatory attention mechanism, we cannot

**Table 2.** Comparison of the models using out-of-sample, in-sample, and 10-fold cross-validation accuracy

| | SemEval-2015 | | | | SemEval-2016 | | | |
|---|---|---|---|---|---|---|---|---|
| | Out-of-sample | In-sample | Cross-validation | | Out-of-sample | In-sample | Cross-validation | |
| | Acc. | Acc. | Acc. | St. dev. | Acc. | Acc. | Acc. | St. dev. |
| Ont | 65.8% | 79.7% | 79.7% | 0.0183 | 78.3% | 75.3% | 75.3% | 0.0152 |
| BoW | 76.2% | 91.0% | 87.9% | 0.0311 | 83.2% | 89.3% | 84.5% | 0.0254 |
| CABASC | 76.6% | 85.8% | 87.1% | 0.0138 | 84.6% | 79.2% | 84.0% | 0.0218 |
| LCR-Rot | **78.4%** | 86.2% | 88.0% | 0.0144 | 86.9% | 92.9% | 85.8% | 0.0214 |
| LCR-Rot-inv | 77.1% | 85.2% | 88.1% | 0.0146 | 86.5% | 93.9 | 85.5% | 0.0161 |
| LCR-Rot-hop | **78.4%** | 88.6% | 87.6% | 0.0181 | **87.7%** | 86.3 | 85.6% | 0.0169 |
| Ont+BoW | 79.5% | 86.9% | 83.5% | 0.0308 | 85.6% | 86.7% | 85.7% | 0.0329 |
| Ont+CABASC | 79.6% | 84.3% | 83.2% | 0.0138 | 85.9% | 82.3% | 85.5% | 0.0298 |
| Ont+LCR-Rot | **80.6%** | 84.5% | 83.7% | 0.0144 | 87.0% | 88.3% | 86.3% | 0.0323 |
| Ont+LCR-Rot-inv | 79.9% | 89.0% | 83.7% | 0.0146 | 86.6% | 88.7% | 86.2% | 0.0296 |
| Ont+LCR-Rot-hop | **80.6%** | 85.7% | 83.5% | 0.0298 | **88.0%** | 86.7% | 86.2% | 0.0308 |

derive more information about relevant words. However, when iterating over the rotatory attention mechanism for three times, the model is able to improve the attention and interaction between aspect and context.

When analyzing the two-stage approaches we can observe an improved performance with respect to the base ontology approach. For each case, the ontology is able to provide additional information that the backup methods do not capture. Again, we observe that the hybrid method that uses LCR-Rot-hop performs the best and this is also the only hybrid method that outperforms the individual LCR-Rot-hop method, albeit with a very small percentage of 0.3%.

In order to find an explanation why LCR-Rot-hop and LCR-Rot perform better then CABASC, and why LCR-Rot-hop performs better then LCR-Rot, we will analyze the differences between the attention weights. Figure 4 gives a visualization of a sentence where LCR-Rot and LCR-Rot-hop make a correct prediction and CABASC makes an incorrect prediction. The analyzed sentence

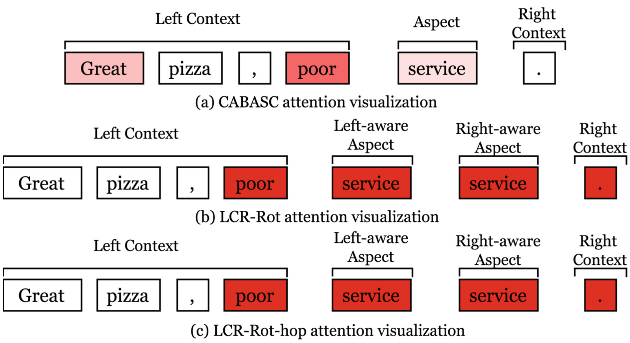



**Fig. 4.** Attention visualizations of the LCR-Rot, LCR-Rot-hop, and CABASC models for the phrase ‘Great pizza, poor service’.

is 'Great pizza, poor service.' The red color denotes words to which the model pays attention. The darker the color, the higher the attention weight and the more important a word is for the representation.

The LCR-Rot model (Fig. 4b) and LCR-Rot-hop model (Fig. 4c) are both able to capture the most indicative sentiment word with respect to the aspect, i.e., 'poor'. The CABASC (Fig. 4a) model is also able to capture the words in the sentence that indicate a sentiment. However, it pays attention to both the words 'great' and 'poor', where the former word belongs to the aspect 'pizza' and only the latter is relevant for the aspect 'service'. In this simple case, where two different sentiments are expressed with respect to two aspects, CABASC finds it difficult to address which sentiment indicating words belong to which aspect. This might be due the fact that the sentences are short and both the aspects and sentiment indicating words are close to each other. On the contrary, both rotatory attention models are able to make this distinction, by better capturing the relevant sentiment words that belong to a specific aspect.

Figure 5 graphically shows attention weights for the sentence 'The food in here is incredible, though the quality is inconsistent during lunch'. Again, two sentiments are expressed, positive and negative, respectively related to two aspects, 'food' and 'lunch'. However, regarding sentiment, the sentence has a more difficult structure, since both aspects are closely related to each other. Similar as in Fig. 4a CABASC (Fig. 5a) is not able to detect which sentiment indicating words belong to which aspect and hence pays equal attention to both the words 'incredible' and 'inconsistent'. This also holds for the LCR-Rot method

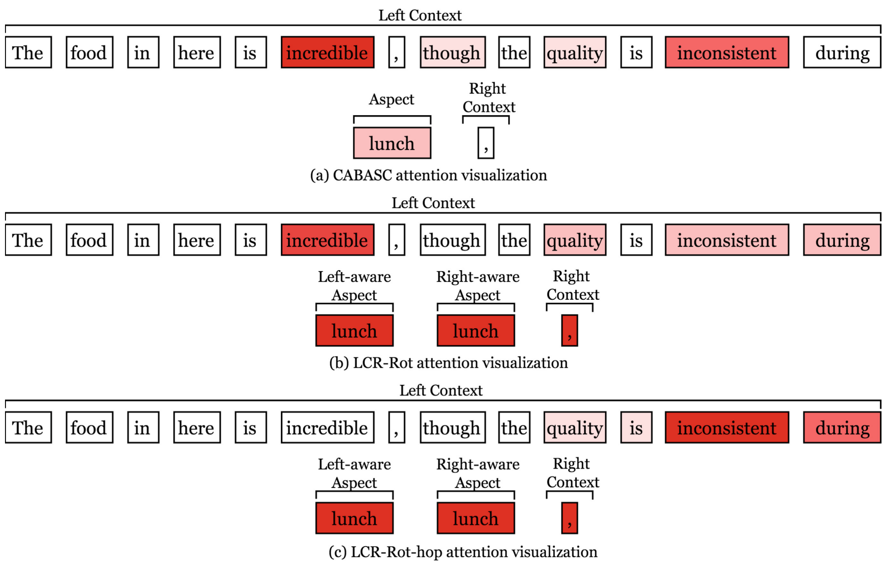



**Fig. 5.** Attention visualizations of the LCR-Rot, LCR-Rot-hop, and CABASC models for the phrase 'The food in here is incredible, though the quality is inconsistent during lunch'.

(Fig. 5b), it is not able to notice the irrelevance of the word ‘incredible’. However, it is able to detect the relation between the word ‘inconsistent’ and the aspect, by paying attention to the preposition ‘during’.

LCR-Rot-hop (Fig. 5c) is the only method that fully focuses attention to the relevant sentiment word ‘inconsistent’ and capture the connecting preposition ‘during’. For these reasons it makes a correct prediction. When the sentiment expressed in a sentence is constructed in a more complex manner, repeating the rotatory attention mechanism helps to separate irrelevant sentiment words from relevant sentiment words.

## 6 Conclusion

This paper focuses on aspect-based sentiment analysis of restaurant reviews on a sentence level. We employ an ontology-driven hybrid approach, following the main idea of [16] and using the LCR-Rot method [21] as a backup algorithm. The two-stage algorithms outperform one-stage baseline models, albeit with a small percentage. Our best performing method (Ont+LCR-Rot-hop) outperforms the Ont+BoW model proposed by [16] by 2% point when considering the SemEval-2016 dataset.

We conclude that all the machine learning methods are able to effectively find words that carry sentiment. However, the models differ in how much they are able to focus on sentiment words that are only relevant for the given aspect. By repeating the rotatory attention mechanism, the LCR-Rot-hop method is able to make this distinction the best and therefore has the highest performance.

A suggestion for future work is to further improve the models used in this paper. With regard to the neural models, one can experiment further with the rotatory attention mechanism by looping until a convergence is reached or by hopping through the LCR-Rot-inv method. Considering the ontology used in this paper, it can be enlarged by including more lemmas, classes, and axioms. A final suggestion for future research is the usage of different word embeddings.

## References


1. Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., Goldberg, Y.: Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In: Proceedings of the 2017 International Conference on Learning Representations (ICLR 2017) (2017). CoRR abs/1608.04207
2. Bergstra, J.S., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS 2011), pp. 2546–2554. Curran Associates (2011)
3. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Sebastopol (2009)
4. Cambria, E.: Affective computing and sentiment analysis. IEEE Intell. Syst. **31**(2), 102–107 (2016). https://doi.org/10.1109/mis.2016.31

5. Cambria, E., Hussain, A.: Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis. SC, vol. 1. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23654-4
6. Chen, P., Sun, Z., Bing, L., Yang, W.: Recurrent attention network on memory for aspect sentiment analysis. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pp. 452–461. Association for Computational Linguistics (2017)
7. Hutto, C.J., Gilbert, E.: VADER: a parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the 8th International Conference on Weblogs and Social Media (ICWSM 2014). AAAI Press (2014)
8. Liu, B.: Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press, New York (2015)
9. Liu, Q., Zhang, H., Zeng, Y., Huang, Z., Wu, Z.: Content attention model for aspect based sentiment analysis. In: Proceedings of the 27th International World Wide Web Conference (WWW 2018), pp. 1023–1032. ACM Press (2018). https://doi.org/10.1145/3178876.3186001
10. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995). https://doi.org/10.1145/219717.219748
11. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of The 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1532–1543. Association for Computational Linguistics (2014)
12. Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., Androutsopoulos, I.: Semeval-2015 task 12: aspect based sentiment analysis. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 486–495. Association for Computational Linguistics (2015)
13. Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., Androutsopoulos, I.: Semeval-2016 task 5: aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), pp. 19–30. Association for Computational Linguistics (2016)
14. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: Semeval-2014 task 4: aspect based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 27–35. Association for Computational Linguistics and Dublin City University, August 2014
15. Schouten, K., Frasincar, F.: Survey on aspect-level sentiment analysis. IEEE Trans. Knowl. Data Eng. (TKDE) **28**(3), 813–830 (2016)
16. 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
17. Schouten, K., Frasincar, F., de Jong, F.: Ontology-enhanced aspect-based sentiment analysis. In: Cabot, J., De Virgilio, R., Torlone, R. (eds.) ICWE 2017. LNCS, vol. 10360, pp. 302–320. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60131-1_17
18. Socher, R., et al.: 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), vol. 1631, pp. 1631–1642. ACL, January 2013
19. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)

20. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods on Natural Language Processing (EMNLP 2016), pp. 606–615. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/d16-1058
21. Zheng, S., Xia, R.: Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention (2018). arXiv preprint arXiv:1802.00892