Explaining a Neural Attention Model for Aspect-Based Sentiment Classification Using Diagnostic Classification

Lisa Meijer
Delft University of Technology
Delft, the Netherlands
lhmeijer@tudelft.nl

Flavius Frasincar
Erasmus University Rotterdam
Rotterdam, the Netherlands
frasincar@ese.eur.nl

Maria Mihaela Truşcă
Bucharest Univ. of Economic Studies
Bucharest, Romania
maria.trusca@csie.ase.ro

ABSTRACT

Many high performance machine learning models for Aspect-Based Sentiment Classification (ABSC) produce black box models, and therefore barely explain how they classify a certain sentiment value towards an aspect. In this paper, we propose explanation models, that inspect the internal dynamics of a state-of-the-art neural attention model, the LCR-Rot-hop, by using a technique called Diagnostic Classification. Our diagnostic classifier is a simple neural network, which evaluates whether the internal layers of the LCR-Rot-hop model encode useful word information for classification, i.e., the part of speech, the sentiment value, the presence of aspect relation, and the aspect-related sentiment value of words. We conclude that the lower layers in the LCR-Rot-hop model encode the part of speech and the sentiment value, whereas the higher layers represent the presence of a relation with the aspect and the aspect-related sentiment value of words.

CCS CONCEPTS

• **Information systems** → **Sentiment analysis**; *Information extraction*; Web mining;

KEYWORDS

aspect-based sentiment classification, neural rotatory attention model, diagnostic classification

1 INTRODUCTION

Aspect-based Sentiment Classification (ABSC) is a subtopic of the Textual Classification task whose main aim is the assignment of sentiment orientations (positive, negative, or neutral) at the aspect-level of the entity of interest. In the recent years, many models have been designed for ABSC, among which machine learning together with its deep learning subset have proved to be the most

effective. However, the performance of machine learning models for the ABSC is usually traded off for a high level of uncertainty and vagueness in the logic of models [7]. The reason behind the unsuccess in clarifying the model's inner working [14] is especially by the large numbers of parameters that need to be optimised. In addition to the benefits to businesses, the regulation of General Data Protection Regulation (GDPR) [16] has shown us that it has become a societal need to make machine learning models more understandable, and to transform black boxes into transparent, white boxes.

In this paper, we use a state-of-the-art ABSC model, the Left-Center-Right Rotatory Attention model with multiple hops (LCR-Rot-hop), proposed by Wallaart and Frasincar [17]. Given the LCR-Rot-hop model, it is unclear which input information gets encoded while processing an opinion, and whether the model has any understanding about which word(s) determine(s) the sentiment polarity towards an aspect. Therefore, our research aim is to shed light on the inner working of the LCR-Rot-hop model by means of Diagnostic Classification [5]. Precisely, we use Diagnostic Classification to examine whether the internal layers of the LCR-Rot-hop model encode for each word of an opinion the *part of speech*, the *presence of aspect relation*, the *sentiment value*, and the *aspect-related sentiment value* information that it is useful for the current classification task. By means of the LCR-Rot-hop model, we provide a methodology to understand the inner information process specific to the ABSC problem that can be easily applied to any other neural network designed for this purpose. Our implementation can be found at https://github.com/lhmeijer/ABSCEM.

The structure of this paper is as follows. In Section 2, we discuss the relevant literature of ABSC using neural networks, and of Diagnostic Classification. Section 3 gives an overview of the used dataset. Section 4 gives a description of the LCR-Rot-hop model, and our proposed diagnostic classifiers, which we evaluate in Section 5. In Section 6 we present our conclusion and future work.

2 RELATED WORK

The following Section 2.1 and Section 2.2 review the literature dedicated to ABSC and Diagnostic Classification, respectively.

2.1 Aspect-Based Sentiment Classification

Since a writer could express multiple opinions towards different specific aspects in one document or sentence, only looking at a document or a sentence level as in Sentiment Classification is often insufficient [7]. Therefore, we focus on ABSC, a finer-grained sentiment classification, which computes the sentiment for an aspect of an entity of interest, at the sentence level [12].

Nowadays Machine Learning approaches are popular ABSC models. Frequently, these models are based on neural networks, such as Long Short-Term Memory (LSTM) networks [4] and neural attention models [8]. Tang et al. [15] proposed a target-dependent LSTM (TD-LSTM) approach, which models the relatedness of an aspect with its context words, and selects the relevant parts from the context to deduce the sentiment value towards the aspect. Liu et al. [8] developed the Content Attention Based Aspect-Based Sentiment Classification (CABASC) model, which consists of two attention modeling mechanisms, one at the sentence level and one at the context level. While the sentence level content attention mechanism considers the whole meaning of the full sentence, the context attention mechanism takes the correlations between the words and the aspect into account.

Zheng and Xia [18] suggested a Left-Center-Right separated neural network with a rotatory attention mechanism (LCR-Rot) to effectively represent an aspect composed of multiple words, and to utilize the interaction between the aspect and its context. Additionally, Wallaart and Frasincar [17] proposed the LCR-Rot-hop model as an extension of the the work of Zheng and Xia [18]. The LCR-Rot-hop model iterates the rotatory attention mechanism to better indicate the relation between the aspect and the context of an opinion. Given that the LCR-Rot-hop model is a state-of-the-art approach to ABSC with higher rates of effectiveness than TD-LSTM, LCR-Rot, and CABASC models, we only consider this model for our diagnostic analysis.

2.2 Diagnostic Classification

Although, neural networks are good at predicting the sentiment value of an opinion, they are not highly transparent predictors. Therefore, Hupkes et al. [5] proposed a Diagnostic Classifier that attempts to predict information from the (hidden) states of a neural network. If this information is predicted accurately, it indicates that the information is indeed computed or represented by the network at the level of the given hidden states.

The idea of a diagnostic classifier is based on a prediction task proposed by Adi et al. [1]. They trained a classifier to predict specific sentence properties (e.g., number of words, word context, and word order) based on the vector representations obtained from LSTM autoencoders or simple continuous bag-of-words representations. Adi et al. stated that a property of a sentence is not encoded in the representation, if we cannot train a classifier to predict this property.

Initially, Hupkes et al. [5] proposed a diagnostic classifier to study how recurrent neural networks process hierarchical structures by using an arithmetic language, but Hupkes et al. mainly showed that diagnostic classification is a useful technique for opening up the black box of neural networks. For instance, Jumelet and Hupkes [6] used the diagnostic classification to predict whether a word in a sentence is inside the licensing scope. Jumelet and Hupkes concluded that their neural language model is able to encode this information.

Belinkov et al. [2] evaluated the representation quality of neural machine translation models on part-of-speech and morphological tagging in various languages by using a neural classifier. Belinkov

et al. [2] concluded that lower layers of the neural machine translation models are better at capturing morphology than the higher layers.

3 SPECIFICATION OF THE DATA

In this paper, we use the widely employed SemEval-2016 Task 5 Subtask 1 dataset [11] to train, and evaluate the LCR-Rot-hop model. The SemEval-2016 dataset consists of hundreds of restaurant reviews. Each review is divided into various sentences, and each sentence has one or multiple opinions.

Figure 1 presents a review in the XML markup language. This sentence includes three opinions. For each opinion, a `polarity` (sentiment value) is specified, which expresses the sentiment of the reviewer towards the predefined aspect or `target`. Moreover, an aspect `category` is stated, which classifies the `target`.

```
<Review rid="404464">
<sentences>
<sentence id="404464:0">
<text>Thalia is a beautiful restaurant with
beautiful people serving you, but the food
doesn't quite match up.</text>
<Opinions>
<Opinion target="people"
category="SERVICE#GENERAL" polarity="positive"
from="48" to="54"/>
<Opinion target="food" category="FOOD#QUALITY"
polarity="negative" from="76" to="80"/>
<Opinion target="Thalia"
category="AMBIENCE#GENERAL" polarity="positive"
from="0" to="6"/>
</Opinions>
</sentence>
</sentences>
</Review>
```

Figure 1: A sentence from the SemEval-2016 dataset

Since the model relies on the presence of explicitly stated opinions, we remove all implicitly stated opinions from the dataset. Moreover, we replace all long dashes (−) by a short dash (-), and all textual apostrophes (') by a digital typesetting (') to encode these similar characters in the same way. We use the Stanford CoreNLP package [9] to tokenize and lemmatize all the words within a sentence.

To symbolise the meaning of a word, such that we can use it as input in our model, we represent a lemmatized word as a point in a 300 dimensional semantic space. We use the pre-trained algorithm GloVe [10] to obtain these vector representations (embeddings). If words do not appear in the GloVe vocabulary, we randomly initialise these words by using a normal distribution $N(0, 0.05^2)$ [17].

4 METHOD

Below, we first give an overview of the LCR-Rot-hop model in Section 4.1, and then present the diagnostic classification in Section 4.2.

4.1 Neural Rotatory Attention model with multiple hops

According to the LCR-Rot-hop model [17], we initially split the sentences in a left, a center (target), and a right part depending on the position of the target. The left part consists of L words, the right part has R words, and the target phrase consists of T words belonging to the aspect.

The model starts with three bi-directional Long-Short-Term-Memory (Bi-LSTM) networks for each of the three parts of the input sentence (left context, target content, and right context). These Bi-LSTMs use, as input, the embeddings of the words in the left context $[e_1^l, ..., e_L^l]$, the embeddings of the words in the target phrase $[e_1^t, ..., e_T^t]$, and the embeddings of the words in the right context $[e_1^r, ..., e_R^r]$ to compute the corresponding hidden states $[h_1^l, ..., h_L^l]$ for the left context, $[h_1^t, ..., h_T^t]$ for the target phrase, and $[h_1^r, ..., h_R^r]$ for the right context.

To capture the most sentiment indicative words in a sentence, we apply a rotatory attention mechanism to all hidden states. First, we use an attention scoring function for both the left $f(h_i^l, r^{t_p})$ and right $f(h_i^r, r^{t_p})$ contexts to represent the most sentiment indicative words in the left and right contexts, respectively. Both the attention scoring functions are defined with respect to their hidden states, and the average target representation r^{t_p}, calculated by feeding the target hidden states into an average pooling layer. Given the symmetry of the rotatory attention mechanism, we present the computational process only for one context. As a result the attention scoring function for the left context is defined as follows:

$$f(h_i^l, r^{t_p}) = \tanh(h_i^l \times W_c^l \times r^{t_p} + b_c^l) \quad (1)$$

where W_c^l is a weight matrix, and b_c^l is a bias term. We feed these context attention scores $f(h_i^l, r^{t_p})$ and $f(h_i^r, r^{t_p})$ into a softmax function for normalization. With the normalised attention scores α_i^l and α_i^r, we compute the left r^l and right r^r context representations. Considering the left context again, the r^l vector is computed as:

$$r^l = \sum_{i=1}^{L} \alpha_i^l \times h_i^l \quad (2)$$

Second, we use these context representations r^l and r^r, and an attention scoring function for both the left-aware target phrase $f(h_i^t, r^l)$ and the right-aware target phrase $f(h_i^t, r^r)$ to represent the most sentiment indicative words in the target phrase. The attention scoring function defined with respect to the left context is:

$$f(h_i^t, r^l) = \tanh(h_i^t \times W_t^l \times r^l + b_t^l) \quad (3)$$

where W_t^l is a weight matrix and b_t^l is a bias term. These target attention scores $f(h_i^t, r)$ are as well fed into a softmax function to obtain the normalised attention scores $\alpha_i^{t_l}$ and $\alpha_i^{t_r}$. We use these scores to compute the left-aware target r^{t_l} and right-aware target r^{t_r} representations. Similar to the r^l vector defined above (eq. 2), the r^{t_l} representation is computed as:

$$r^{t_l} = \sum_{i=1}^{T} \alpha_i^{t_l} \times h_i^t \quad (4)$$

Next, we repeat the above mentioned steps of applying the rotatory attention mechanism by feeding the left- and right-aware target representations into the left and right attention functions, respectively, instead of the average target representation (r^{t_p}), and we compute again all context and target representations. Moreover, the rotatory attention mechanism is repeated over multiple iterations to better represent the relation between the aspect and its context.

At the end of the iterative rotatory attention mechanism, we concatenate the left-context representation r^l, the two side-target representations, r^{t_l} and r^{t_r}, and the right-context representation r^r, to obtain the sentence representation s. Further, the new representation of the input sentence is converted by a dense layer with a softmax activation function to compute the sentiment prediction vector with a score for each of the three *sentiment values*: positive, negative, and neutral.

The LCR-Rot model is trained in a supervised manner by minimising the cross-entropy loss function with an L_2-norm regularisation term. To update the weights and biases we use stochastic gradient descent with momentum, and to prevent overfitting we apply the dropout technique to all hidden layers. For a more in-depth discussion of the training process, we refer to [17].

4.2 Diagnostic Classification

Diagnostic Classification is based on the idea that if a model is keeping track of certain information, we should be able to extract this information from the internal layers of the model. We propose several diagnostic classifiers to examine whether the internal layers of the LCR-Rot-hop model are keeping track of the *part of speech information*, the presence of a *relation with the aspect*, the *sentiment value*, and the *sentiment value related to the aspect*, all information deemed useful at the word level for ABSC. If the diagnostic classifiers are able to predict accurately, it indicates that the considered information is indeed encoded by the layers of the LCR-Rot-hop model.

Since our diagnostic classifiers try to predict from the internal layers of the LCR-Rot-hop model, we first train the LCR-Rot-hop model on the SemEval-2016 training dataset to obtain the internal layers of all words in the opinions of the training and test dataset. For each word i, we obtain its embedding (e_i); its hidden state (h_i); and its $j \in \{1, ..., n\}$ context representations (r_{ij}), where j represents the hop number. Since the context representation r is computed as a single value for a group of words (eq. 2), the r_{ij} vector representation at the word-level is defined as follows:

$$r_{ij} = \alpha_{ij} \times h_i \quad (5)$$

where α_{ij} is either the left attention score (α_{ij}^l) or the right attention score (α_{ij}^r) for the j iteration of the rotatory attention mechanism. In order to predict from the hidden layers ($e_i, h_i, r_{i1}, ..., r_{in}$) the *part-of-speech*, the *presence of a relation with the aspect*, the *sentiment value*, and the *sentiment value in relation to the aspect*, we define below the information that makes up the class labels.

To predict the *part-of-speech* tags, we categorise the words as: noun, verb, adjective (adj.), or adverb (adv.). If a word is none of those four, we categorize it as remaining *part-of-speech* (rem.). We simply extract these labels from each individual word by using

the Stanford CoreNLP package [9]. Table 1 represents the number of words per *part-of-speech* class. The diagnostic classifiers are trained only on the correctly predicted instances of the training data because it does not make sense to use the incorrectly predicted training data that might cause the learning of inconsequential patterns. Therefore, we choose to not present the distribution per classes of the words in the training instances that are incorrectly predicted by the LCR-Rot-hop in Tables 1-4.

Table 1: Number of words per *part-of-speech* class.

	Noun	Verb	Adj.	Adv.	Rem.
Correctly predicted training set	4586	3631	2058	4661	14148
Correctly predicted test set	1211	1099	569	2338	4848
Incorrectly predicted test set	278	182	129	264	908

To predict the *presence of a relation* with the aspect, we label a word as "Yes" if it is related to the aspect, and as "No", if it is not related to the aspect. We determine the existence of a relation by using the Stanford CoreNLP Dependency Parser [9], and the ontology described by Schouten and Frasincar [13] that connects aspects to words. A word is related to the aspect in the opinion if there exists a dependency between them, or if they are connected according to the ontology. The number of words per *presence of a relation* class is given in Table 2.

Table 2: Number of words per *presence of a relation* class.

	Yes	No
Correctly predicted training set	6220	22864
Correctly predicted test set	2070	7995
Incorrectly predicted test set	364	1397

To predict the *sentiment value*, we assign each word to the class label "Positive", if the word has a positive *sentiment value*, "Negative", if the word has a negative *sentiment value*, and "No Sentiment", if we cannot define the *sentiment value*. Table 3 represents the classes' frequencies in the data sets. We specify the *sentiment value* of a word by using the ontology described by Schouten and Frasincar [13], and NLTK SentiWordNet [3]. The ontology classifies a word positive (negative), if the word has a superclass *Positive* (*Negative*). If the ontology is not able to classify the *sentiment value*, we use SentiWordNet [3] that defines the sentiment score of the word's most frequently used form. A word has a positive *sentiment value*, if the positive sentiment score is larger than the negative sentiment score, and vice versa. We classify the word as "No Sentiment", if both the ontology and SentiWordNet are not able to identify the *sentiment value*. Table 3 lists the number of words per *sentiment value* class.

Table 3: Number of words per *sentiment value* class.

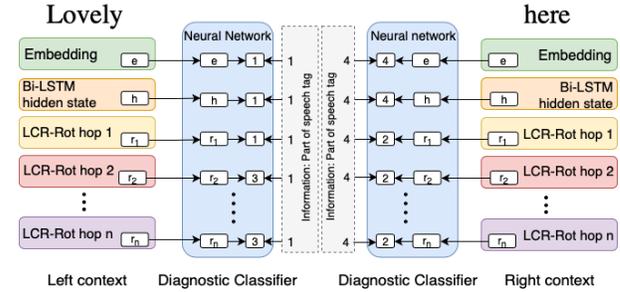|	Positive	Negative	No Sentiment
Correctly predicted training set	5583	1774	21727
Correctly predicted test set	1677	489	7899
Incorrectly predicted test set	264	120	1377

Figure 2: Diagnostic classifier applied for the detection of *part-of-speech* of the words "Lovely" and "here"

To predict the *sentiment value of a word related to the aspect* in the opinion, we classify a word as "Positive" if it has both a positive *sentiment value* and a relation towards the aspect, as "Negative" if it has both a negative *sentiment value* and a relation towards the aspect, and as "No Aspect Sentiment" if the *sentiment value* or the relation are undefined. We specify the *sentiment value* in the same way, as we did for predicting the *sentiment value*, and we determine the existence of a relation in the same way, as we did for predicting the *presence of a relation*. Table 4 gives the number of words per *aspect-related sentiment value* class.

Table 4: Number of words per *aspect-related sentiment value* class.

	Pos.	Neg.	No Aspect Sentiment
Correctly predicted training set	1425	439	27220
Correctly predicted test set	450	91	9524
Incorrectly predicted test set	57	38	1666

Since the number of words per class label for any information is rather unequally distributed, we run the LCR-Rot-hop and select sub-samples from the correctly predicted opinions of the SemEval-2016 training set to train the diagnostic classifiers. We randomly draw $\min(q_c, q^{mean})$ words per class, where q_c is the number of word with class label c, and q^{mean} is the mean of all q_c's except from the class that has the greatest number of words. The reason behind this configuration is the need to reduce the differences between the classes in terms of size.

After composing multiple training sets from the internal layers of the randomly selected words, and their corresponding class labels, we train the various diagnostic classifiers on their related training set. Figure 2 gives an illustration of all $2 + n$ diagnostic classifiers, which try to predict the *part-of-speech* of the words "Lovely" and "here" from their various hidden layers ($e, h, r_1, ..., r_n$). The words "Lovely" and "here" represent the left and right contexts of the imaginary opinion "Lovely place here", where the "place" word is the aspect. "Lovely" is an adjective, and should be predicted as 1, while "here" has a remaining tag, and should be predicted as 4 (where the class labels are ordered from 0 to 4). Since we select a sub-sample to train the diagnostic classifiers on, a single run is insufficient to draw a conclusion. Therefore we train our diagnostic classifiers on 10 subsets of the correctly predicted instances by LCR-Rot-hop from
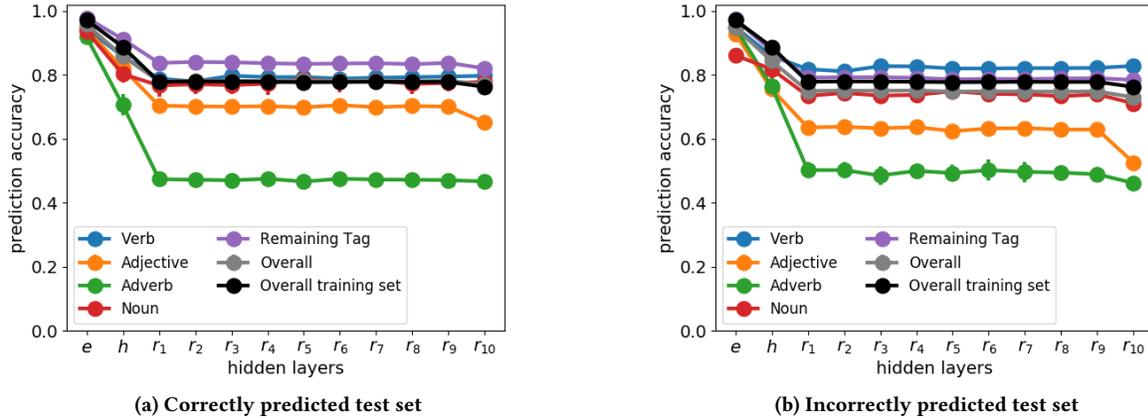
(a) Correctly predicted test set

(b) Incorrectly predicted test set

Figure 3: Accuracies of predicting the *part-of-speech*

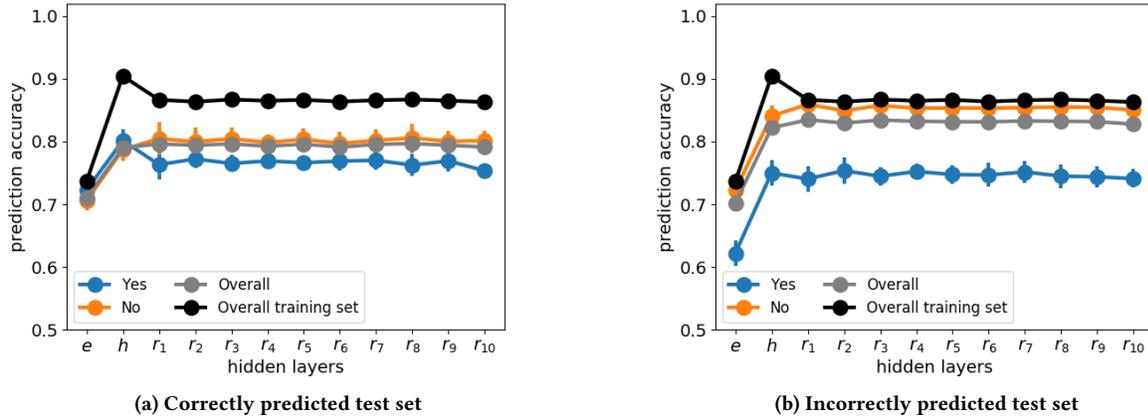(a) Correctly predicted test set

(b) Incorrectly predicted test set

Figure 4: Accuracies of predicting the *presence of a relation with an aspect*

the SemEval-2016 training set. The subsets are sampled with replacement. We later report the averages and standard deviations of the accuracies obtained with 10 diagnostic classifiers, each trained on one of the 10 considered subsets.

Inspired by the previous work [2], all our diagnostic classifiers are simple neural networks with one hidden layer of 300 neurons with a ReLU activation function. The output is determined by a dense layer with a softmax function. We initialise the weights by a uniform distribution $U(-0.1, 0.1)$ and the biases by zero for each layer. We use a learning rate of 0.0001 and a batch size of 20. The number of epochs is 90 except for the diagnostic classifiers predicting the *part-of-speech*. Since these classifiers tend to end up in a saddle point after too many epochs, we set their number of epochs to 20.

After we train all diagnostic classifiers on the internal layers and class labels of the randomly selected words in the correctly predicted opinions from the SemEval-2016 training dataset, we predict the class labels from the internal layers ($e_i, h_i, r_{i1}, ..., r_{in}$) of all words in the SemEval-2016 test dataset, and verify the accuracy of the diagnostic classifiers. If a diagnostic classifier predicts the class

labels with a high accuracy, we conclude that the corresponding hidden layer is keeping track of the corresponding information.

5 EVALUATION

The following Section 5.1 and Section 5.2 present the performances of the LCR-Rot-Hop neural network and diagnostic classifiers, respectively.

5.1 Performance of the LCR-Rot-hop

Instead of using 3 hops like [17], we train the LCR-Rot-hop model using 10 hops that allows us to better understand the information detection process of the rotatory attention. The model runs using 150 epochs, and a learning rate of 0.05, an L_2-norm regularisation term of 0.00001, a momentum of 0.9, and a batch size of 20. Furthermore, during training 50% of neurons are randomly dropped out of the network. Table 5 gives its accuracy results on the SemEval-2016 training and test data for a single run. Unlike the Wallaart and Frasincar's work, we do not run the model in two steps, which means that the LCR-Rot-hop model runs over the entire test set, without
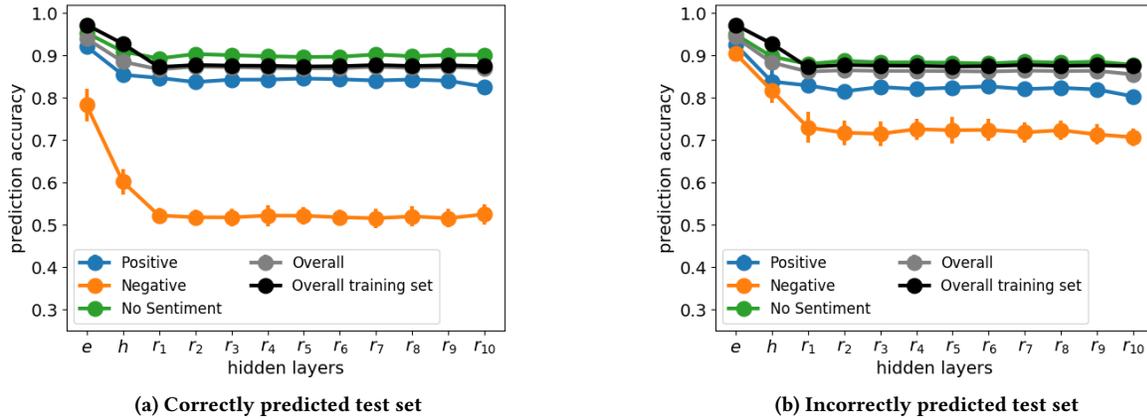
(a) Correctly predicted test set

(b) Incorrectly predicted test set

Figure 5: Accuracies of predicting the *sentiment value*

(a) Correctly predicted test set

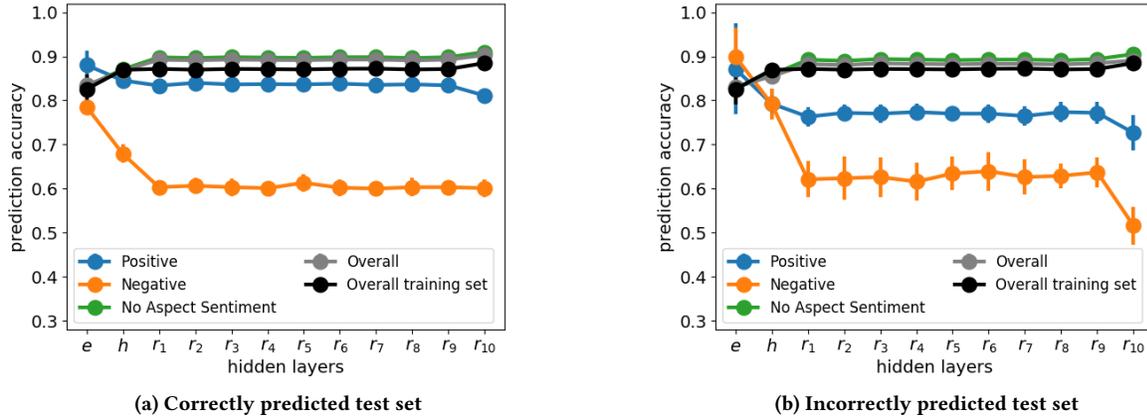(b) Incorrectly predicted test set

Figure 6: Accuracy of predicting the *aspect-related sentiment value*

Table 5: Accuracy results of the LCR-Rot-hop (with 10 hops)

	Training set		Test set	
	Freq.	acc. (%)	Freq.	acc. (%)
Positive	1319	99.4	483	94.6
Neutral	72	95.8	32	25.0
Negative	488	95.9	135	69.6
Overall	1879	98.4	650	86.0

applying the ontology first. This is because we are interested to measure the performance of the neural network only.

5.2 Performance of the diagnostic classifiers

Figures 3, 4, 5, and 6 illustrate the mean accuracies, and standard deviations of predicting the *part-of-speech*, *sentiment value*, *presence of a relation with an aspect*, and *aspect-related sentiment value* of the words in the correctly and incorrectly predicted opinions in the test data set. Since the standard deviations are rather small, the vertical lines, which represent the standard deviations in the graphs, are fairly short, and often not observable. We plot a colored line, for

the prediction performance for each class label over the test set. We plot a grey line for the overall mean predicting performance over the test set, and we plot a black line for the overall mean prediction performance over the correctly predicted opinions from the training set. The information is split over over the internal representations of LCR-Rot-hop model: embeddings (*e*), hidden states (*h*), and context representation (*r*).

The performance of predicting the *part-of-speech* by its diagnostic classifiers is illustrated by Figure 3. Given that the information about the performance is split between the internal layers, it allows us to notice the much higher accuracy of predicting the *part-of-speech* from the embeddings than of predicting from the context representations. Furthermore, the internal layers of the correctly predicted test set determine the class label "Adjective" much better than the same layers of the incorrectly predicted test set. The reason behind this lies in the fact that the adjectives have usually *sentiment value*, and affect the overall sentiment orientation.

The performance of predicting the *presence of a relation* with the aspect by its diagnostic classifiers is illustrated by Figure 4. Unlike the case of *part-of-speech*, we can predict the absence and

the presence of a relation better from the context representations than from the embeddings. This presumption is valid, especially in predicting the absence of a relation. Remarkably, the internal layers of the incorrectly predicted test set indicate much better the absence of a relation than the internal layers of the correctly predicted test set.

Figure 5 illustrates the performance of predicting the *sentiment value* by its diagnostic classifiers. Analyzing the figure, we notice that the classifiers applied at the level of internal layers are particularly good at predicting "No Sentiment", and poor at predicting "Negative". Furthermore, from Figure 5 we see that it is easier to extract the sentiment value from the embeddings than from the context representations. Also, we notice that the correctly predicted test set indicates better the words with the label "No Sentiment" than the scenario with incorrect predictions. Knowing which words in the opinion do not have a defined sentiment value, may be more important for correctly predicting the sentiment value, than knowing the sentiment words.

Given Figure 6, we see that the context representation predicts slightly better the *aspect-related sentiment value* than the embeddings. Also, the diagnostic classifiers identify more easily the words with no sentiment value or unrelated to aspects. In addition, Figure 6 shows the label "Positive" from the internal layers of the correctly predicted test set is predicted with a higher accuracy than from the case with incorrect predictions. Probably, keeping track of positive words, which are related to the aspect, has a greater effect on predicting the sentiment value correctly than keeping track of negative words, since Table 5 shows us that the model predicts positive opinions much more accurately than negative opinions.

Looking at all four figures, we see that we do not predict more accurately from the context representation of the first round than from the context representation of the tenth round, or vice versa. Their prediction accuracies are in general about the same value. Apparently, iterating the rotatory attention mechanism does not contribute to an improvement or deterioration in the prediction accuracies of the diagnostic classifiers. The context representations are keeping track of roughly the same information. While the context representation are keeping track of roughly the same information, the embeddings representations and hidden states bring informational diversity.

6 CONCLUSION

In this paper, we propose various diagnostic classifiers in order to examine whether the ABSC state-of-the-art neural attention model LCR-Rot-hop encodes the *part of speech*, the *presence of aspect relation*, the *sentiment value*, and the *aspect-related sentiment value*, deemed as useful information at the word level for the considered task. We conclude that the context representation determines mostly the *presence of a relation with an aspect*, and the *aspect-related sentiment value*, without having a relevant performance difference among the 10 considered context representations. On the other hand, the remaining two features *part-of-speech*, and *sentiment value* are better detected by the embedding representations.

As for future work, we advice to optimise the hyper-parameters of the LCR-Rot-hop model, and the diagnostic classifiers. Moreover, we suggest to better specify the *sentiment value*, and the *presence of*

a relation towards the aspect, since the ontology does not contain all relevant words, and the sentiment score from NLTK SentiWordNet, which is based on the most frequently used form, might not be the correct sentiment score. Furthermore, we propose to compute the prediction accuracy only over the words that indicate the sentiment value of the opinion, instead of computing the prediction accuracy over all words. In doing so, we can better understand how these words affect the final sentiment classification of the LCR-Rot-hop model. As well, we consider important to check the applicability of our inferences also on other sentiment datasets that comprise not only restaurant opinions.

REFERENCES

[1] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. In *5th International Conference on Learning Representations (ICLR 2017)*.

[2] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James R. Glass. 2017. What do Neural Machine Translation Models Learn about Morphology?. In *55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*. ACL, 861–872.

[3] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media.

[4] Sepp Hochreiter and Jürgen Schmidhuber. 1996. LSTM can Solve Hard Long Time Lag Problems. In *9th International Conference on Neural Information Processing Systems 9 (NIPS 1996)*. MIT Press, 473–479.

[5] Dieuwke Hupkes, Sara Veldhoen, and Willem H. Zuidema. 2018. Visualisation and 'Diagnostic Classifiers' Reveal How Recurrent and Recursive Neural Networks Process Hierarchical Structure. *Journal of Artificial Intelligence Research* 61 (2018), 907–926.

[6] Jaap Jumelet and Dieuwke Hupkes. 2018. Do Language Models Understand Anything? On the Ability of LSTMs to Understand Negative Polarity Items. *arXiv preprint arXiv:1808.10627* (2018).

[7] Bing Liu. 2015. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.

[8] Qiao Liu, Haibin Zhang, Yifu Zeng, Ziqi Huang, and Zufeng Wu. 2018. Content Attention Model for Aspect Based Sentiment Analysis. In *2018 World Wide Web Conference on World Wide Web (WWW 2018)*. ACM, 1023–1032.

[9] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Prismatic Inc, Steven J. Bethard, and David Mcclosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014): System Demonstrations*. ACL, 55–60.

[10] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. ACL, 1532–1543.

[11] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2016. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *10th International Workshop on Semantic Evaluation (SemEval 2016)*. ACL, 19–30.

[12] Kim Schouten and Flavius Frasincar. 2015. Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* 28, 3 (2015), 813–830.

[13] Kim Schouten and Flavius Frasincar. 2018. Ontology-Driven Sentiment Analysis of Product and Service Aspects. In *15th Extended Semantic Web Conference (ESWC 2018) (LNCS)*, Vol. 10843. Springer, 608–623.

[14] Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfiste, and Alexander Rush. 2018. Lstmvis: A Tool for Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks. *IEEE Transactions on Visualization and Computer Graphic* 24(1) (2018), 667–676.

[15] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective LSTMs for Target-Dependent Sentiment Classification. In *26th International Conference on Computational Linguistics (COLING 2016)*. ACL, 3298–3307.

[16] Paul Voigt and Axel Von dem Bussche. 2017. The EU General Data Protection Regulation (GDPR). *A Practical Guide, 1st Ed., Springer* (2017).

[17] Olaf Wallaart and Flavius Frasincar. 2019. A Hybrid Approach for Aspect-Based Sentiment Analysis Using a Lexicalized Domain Ontology and Attentional Neural Models. In *16th Extended Semantic Web Conference (ESWC 2019) (LNCS)*, Vol. 11503. Springer, 363–378.

[18] Shiliang Zheng and Rui Xia. 2018. Left-Center-Right Separated Neural Network for Aspect-based Sentiment Analysis with Rotatory Attention. *arXiv preprint arXiv:1802.00892* (2018).