Explaining a Deep Neural Model with Hierarchical Attention for Aspect-Based Sentiment Classification Using Diagnostic Classifiers

Kunal Geed[1], Flavius Frasincar[1][0000−0002−8031−758X], and Maria Mihaela Truşcă[2](✉)

[1] Erasmus University Rotterdam, Burgemeester Oudlaan 50, 3062 PA Rotterdam, the Netherlands
[2] Bucharest University of Economic Studies, 010374 Bucharest, Romania
kunalgeed15@gmail.com, frasincar@ese.eur.nl, maria.trusca@csie.ase.ro

Abstract. LCR-Rot-hop++ is a state-of-art model for Aspect-Based Sentiment Classification. However, it is also a black-box model where the information encoded in each layer is not understood by the user. This study uses diagnostic classifiers, single layer neural networks, to evaluate the information encoded in each layer of the LCR-Rot-hop++ model. This is done by using various hypotheses designed to test for information deemed useful for sentiment analysis. We conclude that the model did not focus on identifying the aspect mentions associated with a word and the structure of the sentence. However, the model excelled in encoding information to identify which words are related to the target. Lastly, the model was able to encode to some extent information about the word sentiment and sentiments of the words related to the target.

Keywords: aspect-based sentiment classification · neural rotatory attention model · diagnostic classification

1 Introduction

The goal of Sentiment Analysis (SA) is to analyse a piece of text and identify the primary sentiment associated with a certain entity in the text [10]. According to [14] Aspect-Based Sentiment Analysis (ABSA) is a sub-task in SA and is generally divided into three different steps. The authors explain that the first step is to identify a sentiment-target pair, followed by classification of the sentiment-target pair, and, lastly, the aggregation of sentiment values to provide an overview. In this paper, we focus on neural networks designed for Aspect-Based Sentiment Classification (ABSC), which refers to the second step responsible for identifying the polarity associated with a specific target.

The application of ABSA is wide, and, although more complicated than SA, can lead to a much more comprehensive analysis. For this purpose, a state-of-the-art technique was developed in [17], which proposes a hybrid approach to

ABSA. Firstly, the authors make use of a domain ontology to identify aspects and sentiments towards these. Any inconclusive cases are then passed to a neural network that predicts the sentiments. Due to its high performance, we make use of this technique as the basis of our research.

Neural networks are considered to be black-box methods as the user is not able to explain the results based on the structure of the neural network, hence their inner-workings are not clear. Therefore, our research aims to improve the understanding of neural networks with a focus on the architecture presented in [17], which is part of the larger field of explainable AI (XAI). To solve this problem, we investigate if the model presented in [17] can capture specific information regarding the relationships between words and aspects. We further extend this by using the domain ontology to test if LCR-Rot-hop++ can encode the domain knowledge represented, in a sentiment analysis context, in the domain ontology. To investigate these questions, we use diagnostic classifiers as introduced in [7]. The major contributions of this work are as follows. While in [11] diagnostic classifiers are used to understand the inner-workings of the LCR-Rot-hop model, we focus on the more advanced LCR-Rot-hop++ model in this paper. Furthermore, in addition to diagnostic classifiers discussed in [11], we investigate if the aspects represented in the domain ontology are encoded in the neural network. To our knowledge, this is one of the first works that investigate the presence of a domain sentiment ontology signal in the representations produced by a neural attention model. All source data and code can be retrieved from `https://github.com/KunalGeed/DC-LCR-Rot-hop_plus_plus`.

The paper is structured as follows. In Sect. 2 we discuss the literature associated with ABSA and XAI. Sect. 3 explores the dataset used in this study and describes the pre-processing steps used to convert the dataset into the final dataset. In Sect. 4, we describe the methodology of the used aspect-based sentiment classifier and the methodology of diagnostic classifiers. Sect. 5 presents the results. Last, Sect. 6 draws conclusions from the results, states the limitations of our study, and suggests avenues for further research.

2 Related Works

This section discusses the relevant literature for this study. Subsection 2.1 provides a more in-depth analysis of Aspect-Based Sentiment Classification. Subsection 2.2 describes the related work of diagnostic classifiers.

2.1 Aspect-Based Sentiment Classification

ABSC usually relies on knowledge-based solutions, machine learning, or hybrid approaches. While classic machine learning models have modest performance rates, the more recent neural networks have managed to significantly increase the classification quality. Within neural networks, Long Short Term Memory (LSTM) [6] and its variants have shown great performance in ABSC. The Left-Center-Right (LCR) separated neural networks for ABSC is introduced in [16]

based on a bi-directional LSTM to address two problems that were about the target representation and the connection between the target and its context.

Although knowledge-based and machine learning approaches had shown individual success, the hybrid techniques developed by combining them proved to be even more effective. The hybrid technique for ABSC introduced in [15] utilizes an ontology-based model to first find as many sentiment classifications as possible and then solves the inconclusive cases using the Bag-of-Words (BOW) model. The model is improved in [18] by changing the backup classifier to the LCR-Rot models proposed in [16]. The authors further extended and improved upon the LCR-Rot model by repeating the rotary mechanism n times (LCR-Rot-hop model). The LCR-Rot-hop model is further improved in [17] by introducing deep contextual word embeddings and hierarchical attention leading to the LCR-Rot-hop++ model.

2.2 Diagnostic Classifiers

With the increase in the use of black-box methods, such as neural networks, there is an increased need for techniques to investigate what happens inside these black-box methods part of XAI [5]. An approach similar to diagnostic classifiers was proposed in [1]. In their work, the authors outline a framework that facilitates the understanding of encoded representation using auxiliary prediction tasks. They score representations by training classifiers which take the representations as input to tackle the auxiliary prediction tasks. If the trained classifier is unable to predict the property being tested in the prediction task, then it is concluded that the representations have not encoded that information [1].

Another technique used to facilitate understanding of the models' innerworking is introduced in [2]. Using a generator model like Variational Auto-Encoder or Generative Adversarial Network, the proposed approach aims to generate artificial inputs that mimic the output produced by the analysed model. As the models are considered black-box methods with no access to their inner gradients, the optimization of the generator relies on an evolutionary strategy. In the end, the artificial inputs are analysed to provide insights into the model capabilities.

Considering that the visualization techniques were not sufficient to gain insight into the information encoded by the recurrent neural network, diagnostic classifiers are introduced in [7] to gain better insight into the information encoded by recurrent neural networks. This led to the development of diagnostic classifiers where the authors tested multiple hypotheses about the information processed by the network. If the diagnostic classifiers can accurately predict the information, then it is concluded that the information is encoded in the network [7].

[8] made use of diagnostic classifiers to link what is going on inside the neural network to linguistic theory. Specifically, they examine the ability of LSTM to process Negative Polarity Items (NPI). The results show that the model can determine a relationship between the licensing context and NPI. As explained in [8], NPI are words that need to be licensed by a licensing context to form a

valid sentence, for example, "He did not buy any books" where "any" is an NPI and "not" is a licensing context. The authors determine that a good language model must be able to encode this relationship. This study is able successfully to link linguistic theory to deep learning [8].

The work in [3] attempts to understand the inner-workings of neural networks and specifically what the neural networks learn about the target language. They determine that lower levels of a neural network are better at capturing morphology. Hence they also hypothesize that lower levels of the neural network capture word structure and the higher levels capture word semantics [3].

[11] makes use of diagnostic classifiers for ABSC. Specifically, the authors evaluate, in detail, the LCR-Rot-hop method developed in [18]. In [11] the LCR-Rot-hop method is analyzed to investigate if the internal layers can encode word information, such as Part-of-Speech (POS) tag, sentiment value, presence of aspect relation, and aspect related sentiment value of words. They conclude that the word structure (POS) is captured by the lower levels of the neural network, and the higher levels are able to encode information about aspect relation and aspect related sentiment value, which is in line with a hypothesis proposed in [3],

3 Specification of the Data

This study makes use of the SemEval 2016 Dataset, Task 5, Sub-task 1, which contains an annotated dataset for ABSA [13]. A review is divided at a sentence level and for each opinion in a sentence, the target, category, and polarity are stated. The polarity of the opinion is the sentiment (positive, negative, or neutral) that the opinion has towards the target. The target is the word in the opinion towards which the sentiment is directed. Last, the category is related to the target and shows which aspect the target belongs to. Table 1 shows the class frequencies for the training and test set used to evaluate LCR-Rot-hop++. In both the test and training set, the *Positive* class is in the majority with more than 70%, and the *Neutral* class is in the minority with less than 5%. This imbalance could make it more difficult for the neural network to learn the *Neutral* class.

Table 1: Polarity frequencies in Training and Test Sets

Training Data			**Test Data**		
Polarity	**Frequency**	**%**	**Polarity**	**Frequency**	**%**
Negative	488	26.0	Negative	135	20.8
Neutral	72	3.8	Neutral	32	4.9
Positive	1319	70.2	Positive	483	74.3

Due to the fact that we use BERT word embeddings to represent words, we need to re-concatenate words that have been divided into word pieces in order to generate the dataset used to train and test the diagnostic classifiers. As any words that begin with "##" is a word piece belonging to the word preceding it, we can combine them into a single word. Due to each word also needing its

own BERT word piece embedding and hidden states, when we combine the word pieces we also need to generate a single word embedding or hidden states for the newly formed word. The word embedding and hidden states represent the layer information that is output by each layer of the LCR-Rot-hop++ model, prior to the final MLP layer for sentiment classification. A proposed solution [19] was to use a recurrent neural network to combine word piece embeddings into a single word embedding, however, without a large dataset to train the neural network this would result in inadequate word embeddings. One of the methods to get a single embedding that captures the meaning of a larger piece of text, such as a phrase or a sentence, from the individual embedding is to average the word embeddings to get a single word embedding representing the entire phrase [9]. We use this approach to combine word pieces and their embedding and layer information into a single vector due to its simplicity.

4 Method

This section is dedicated to the proposed methodology. Section 4.1 presents the backup model of the the two-step approach HAABSA++, and Sect. 4.2 provides an overview of the diagnostic classifiers used to understand the inner-working of the LCR-Rot-hop++ model.

4.1 LCR-Rot-hop++

We aim to investigate if a layer of the backup model of the hybrid approach presented in [17] (more precisely the LCR-Rot-hop++ neural network) encodes certain information. We will begin by training the neural network proposed in [17] on the training data. After the training is complete, we extract the hidden layers from all the correctly predicted instances to generate the features for our training dataset. The accuracy of our methods will be evaluated on the SemEval 2016 test set. We make several diagnostic classifiers to test our various hypotheses. Furthermore, the diagnostic classifiers are trained only on the correctly predicted instances from the training data, as training on the incorrect instances can possibly lead to the diagnostic classifiers learning incorrect information.

The context representations for LCR-Rot-hop++ are calculated at the sentence level. However, to create our dataset we require these representations to be at the word level. We get the word-level representations by omitting the sum when calculating the context representations at the sentence level, hence the formula to get the word level layer information is given in Equation 1.

$$r_i^l = \alpha_i^l \times h_i^l \tag{1}$$

Here, α_i^l is the attention score for the ith word in the left context. Similarly, h_i^l is the hidden state of the word. After this, we apply the hierarchical attention by multiplying the attention score calculated by the hierarchical attention process for the left context with r_i^l as shown in Equation 2.

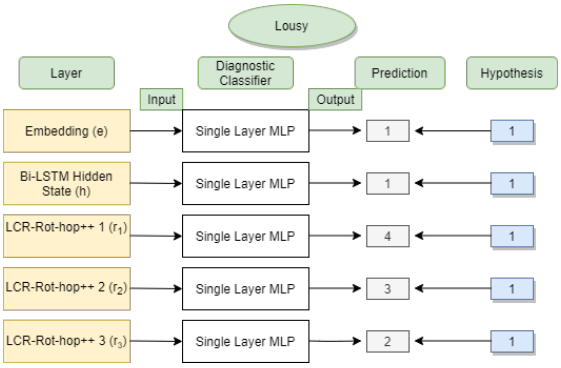
$$r_i^{l'} = \alpha^l \times r_i^l \tag{2}$$

Here, the α^l corresponds to the hierarchical attention score calculated for the left context. By making these changes we can extract hidden states of the various layers at the word level. In total five layers are extracted, $[e, h, r_1, r_2, r_3]$ which stand for the BERT embeddings, hidden states, hierarchical weighted representations 1, 2, and 3, respectively. The BERT embedding layer has a dimensionality of 768, while the rest of the layers have a dimensionality of 600. The dimensionality of 600 is due to the 300-dimensional hidden states of the bi-directional LSTM layer, which results in 600 neurons in total. The final layer is repeated three times (the hop part), hence resulting in five layers in total.

The newly extracted layer information is fed into a single layer MLP which is trained to predict the given hypothesis. A single layer MLP is used as we want a simple model, and the use of a simple model is also inspired by the works proposed in [3] and [11]. If a complicated model is required to extract the encoded information, then the information is not prominently present in the data. Due to the highly imbalanced nature of the dataset, we balance the dataset in the same manner as [11] by drawing $min(q_c, q_{mean})$ instances for each class, where q_c is the number of instances for class c, and q_{mean} is the average number of instances in a class, excluding the class with the highest number of instances.

4.2 Diagnostic Classifier

An overview of diagnostic classifiers is provided in Figure 1. In this figure, we are evaluating the word "lousy" for the POS hypothesis. Knowing that each word is assigned a label that ranges between 0 and 4 for POS tags: Nouns, Adjectives, Adverbs, Verbs, or "Remaining" words, we notice that the adjective "lousy" is properly classified only by the first layers of the model.



Classifier.png

Fig. 1: Overview of the Diagnostic Classifier

In this paper, we test various hypotheses to analyze if the neural network encodes certain information. Below we list the various hypotheses being tested in this paper and how the corresponding tests are generated. Some of these have

already been considered in [11], however, for the simpler LCR-Rot-hop model and not the advanced LCR-Rot-hop++ model.

POS tagging is the process of assigning tags to the words based on the POS and the grammatical categories such as tense, singular/plural, etc. Due to limited amounts of data available we omit predicting grammatical categories and limit ourselves to four Part-Of-Speech tags, already mentioned above. The words classified as anything other than these four are categorized under "Remaining". This process is done using the Stanford CoreNLP package. This hypothesis is designed to check if the neural network can understand the structure of a sentence and its various components. Figure 2a shows an example for POS classification.

Mention Tagging involves predicting the Aspect Mention related to the word. We use the ontology to identify the Aspect Mention a word is connected to. We match the word to a concept in the ontology and ensure maximum matches by checking all lexicalizations of a concept. If there is a match, we check what Aspect Mention this concept is a subclass of in order to identify the aspect the word is referring to. Due to the limited coverage of the ontology, the size of this dataset is much more limited than the others. This hypothesis helps to understand what part of the ontology the neural network can understand and is encoded in the neural network. We test this hypothesis by checking if the neural network can identify various aspects of the ontology. An example of mention tagging is given in Figure 2b.

Aspect Relation Classification is the task of predicting the presence of a relation between the words in the context and the target/aspect. Hence, this is a binary classification problem. To generate the dataset, we make use of the Stanford Dependency Parser, which identifies the various grammatical relationships between words in a sentence. If any relationships exist between a context word and its target, we label that word as 1, and 0, otherwise. This hypothesis helps to check if the neural network is encoding information about the relationship between a context word and the target. Figure 2c shows an example of relation tagging. The dependencies are indicated by an arrow from the context word to the target word.

Word Sentiment Classification is the task of predicting the sentiment of a word as either *Positive*, *Neutral/No Sentiment*, or *Negative*. To identify word sentiment, we make use of a two-step procedure. First, we match the word to a concept in the ontology if it is possible. For this, we use the various lexical representations a concept has. After matching words to a concept, we check if the concept belongs to the *Positive* or *Negative* subclasses of the *Sentiment Value* class defined in the ontology and use that to identify the sentiment. If the word does not match any concept in the ontology or is related to a concept that does not belong to the Positive or Negative subclasses, we use as back-up the NLTK SentiWordNet library to identify the word sentiment. NLTK SentiWordNet identifies the sentiment based on its most frequently used context. It can also classify the word as *Neutral/No Sentiment*. Due to the limited coverage of the ontology, we have to use the NLTK SentiWordNet so that we have a larger dataset to be used to train and test. This hypothesis is designed to identify if

the neural network can correctly detect the sentiment of the word. Figure 2d shows an example for Word Sentiment Classification.

Target-Related Sentiment Classification is a combination of the previous two tasks discussed, namely Word Sentiment Classification and Aspect Relation Classification. We generate another dataset which combines the information from the previous two datasets. If a word has a relation with the target (Aspect Relation Classification) we gather the sentiment of the word (Word Sentiment Classification) and assign that sentiment. If there is no relation or if the sentiment is Neutral, we identify it as "No sentiment". This hypothesis checks if the neural network can identify the words that have a relation to the target and what sentiment they hold. An example of this can be seen in Figure 2e.

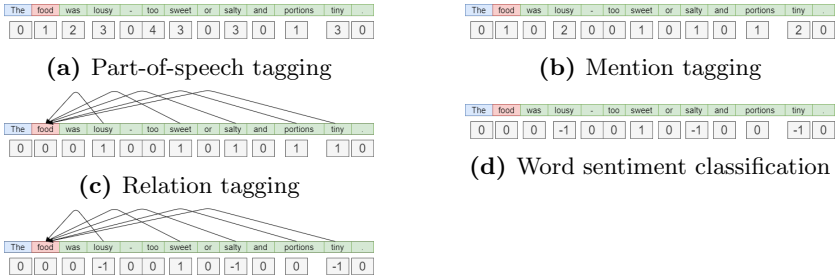


Fig. 2: Examples with part-of-speech tagging, mention tagging, relation tagging, word sentiment classification, and aspect sentiment classification

The diagnostic classifiers are implemented using the `scikit-learn` library in Python. We make use of the `MLPClassifier` function in the library for the diagnostic classifiers. `MLPClassifer` has the ReLU activation function and a constant learning rate of 0.001. Hyper-parameter optimization was performed using the `GridSearchCV` function provided in the `scikit-learn` library on the training data with three folds.

5 Evaluation

To analyze if the neural network can encode hypotheses, such as the structure of a sentence (POS tagging) or the sentiment of a word, we employ diagnostic classifiers to investigate if the layer information can encode the various information correctly. We make use of the accuracy and the weighted F1 score to measure the performance of the diagnostic classifier. We discuss individual hypotheses and compare them to the results reported in [11]. Last, we provide an overview of how the LCR-Rot-hop++ model encoded the various hypotheses and compare their performance.

POS tagging. Table 2 shows the results for the diagnostic classifier trained to predict the POS tag of a word. Table 2 shows that the accuracy is highest for the embedding layer but falls as we move deeper into the neural network, although there is a slight increase at the end. A similar trend is shown by the F1 score, although there is an increase in the weighted F1 score in the second weighted hierarchical layer. This suggests that the deeper layers of the neural network encode less information about the POS tags. Overall, the embedding layer tends to best encode information about the structure of the sentence, while the information is lost or becomes less pronounced in the data as it moves deeper into the network. According to the results reported in [11], a steep fall in the accuracy is visible after the embedding layer, which continues in the hidden state layer. Last, the accuracy is stabilized for the weighted layers, although there is a slight increase in the third weighted layer, which is also observed in our results. However, our reported accuracies for POS tags are significantly lower compared to [11]. A possible reason for the relatively low accuracy and F1 scores could be the BERT embeddings used to represent words. This could confuse the diagnostic classifier as the same words have different representations, in different contexts, but could still have the same POS tag. As we move deeper into the neural network, we are losing information regarding the POS tags which suggest that the model is deeming it unnecessary for sentiment classification. The optimal number of neurons for each classifier is given in Table 2.

Table 2: Diagnostic Classifier results for POS Tagging

Layer	Accuracy (%)	F1 (%)	Number of Neurons
Embedding	65.51%	69.96%	500
Hidden State	58.18%	63.58%	700
Hierarchical Weighted State 1	55.57%	61.53%	500
Hierarchical Weighted State 2	55.54%	61.62%	500
Hierarchical Weighted State 3	56.50%	62.19%	700

Aspect Mention Tagging. The Aspect Mention tagging is a new task introduced in the current work to check if the various aspects in the domain are being encoded in the neural network. According to Table 3, the accuracy falls as we move deeper into the model. While the BERT embedding layer has the highest accuracy, the hierarchical weighted layers are the least effective. However, within the hierarchical weighted layers, the accuracy only decreases minutely and is relatively stable. It is to be noted that the Mention Tagging hypothesis has a highly imbalanced dataset, and after balancing the dataset we are left with a much smaller dataset which might adversely affect the classifier. Furthermore, due to the imbalance in the data, the weighted F1 is a better evaluation metric and also provides a slightly different result. According to F1, the performances of the embedding layer and the hidden state are extremely close to each other. The

embedding layer is below the hidden layer by an extremely small margin. The trend for the weighted F1 scores is downwards, similar to the accuracy. From this information, we can see that the embedding layer is able to best encode information about the Aspect Mentions. Overall, our results suggest that as we move deeper into the neural network, information about the aspects is to some extent lost. It is to be noted that a word could be related to multiple aspects, and hence a multi-class diagnostic classifier could be replaced with a multi-label diagnostic classifier. The optimal number of neurons for each classifier is given in Table 3.

Table 3: Diagnostic Classifier results for Mention Tagging

Layer	Accuracy (%)	F1 (%)	Number of Neurons
Embedding	79.50%	61.91%	500
Hidden State	77.08%	61.99%	900
Hierarchical Weighted State 1	73.49%	60.40%	700
Hierarchical Weighted State 2	73.37%	59.68%	500
Hierarchical Weighted State 3	73.15%	58.22%	500

Aspect Relation Classification. Table 4 shows the results of the diagnostic classifier for identifying Aspect Relations. This task checks if the neural network can identify words that are related to the target. Table 4 shows that the highest accuracy is present in the hidden state layer, while the lowest accuracy is in the embedding layer. As we go deeper into the neural network we see a huge spike in its ability to encode aspect relations at the hidden states layers, but after that, there is a small decline in accuracy for the next layer followed by small fluctuations in the remaining layers. A similar pattern is seen in the weighted F1 score, where the hidden state layer can encode the aspect relations best. This suggests that the model can identify words related to the target better as we move deeper into the neural network and although there is a small drop moving into the hierarchical layers, the model is able to identify words related to the target relatively well. This is logical as the neural network aims to identify words that are related to the target, towards which it is trying to classify the sentiment, and hence its ability to identify words related to the target should improve as we go deeper into the model. Out of all the layers, the hidden states appear to encode aspect relations the best. A possible reason for the hidden state performing better than the hierarchical layers could be that some words are related to the aspect but have no sentiment value, hence the model does not pay attention to those kinds of words deeper into the model, resulting in slightly lower accuracy. [11] showcases a similar pattern for aspect relations. There is a spike for the hidden state layer followed approximately the same values (or lower) for the weighted layers. The optimal number of neurons for each classifier is given in Table 4.

Word Sentiment Classification. Table 5 shows the performance of the diagnostic classifiers for identifying the sentiment of a word. The results prove that as we go deeper into the neural network, the accuracy and the weighted F1 score fall, although there is a spike for the third hierarchical weighted layer. A possible reason for the higher performance of the BERT embedding layer is probably due to the nature of word embeddings that can hold information about their context, alleviating the problem of sentiment detection. Overall, we see that information about the word sentiments is lost as we move deeper into the network. This could be justified due to Type-2 Sentiment Mentions [17] causing some words to not be important for determining the sentiment towards the target as they are not related to that aspect. [11] does find a similar downward trend initially, although at a much higher accuracy. [11] observes that following the downward trend, the accuracy stabilizes for the weighted layers, however, this is not the case for this study as we observe another increase in the final layer. The optimal number of neurons for each classifier is given in Table 5.

Table 4: Diagnostic Classifier results for Aspect Relation

Layer	Accuracy (%)	F1 (%)	Number of Neurons
Embedding	73.06%	78.03%	700
Hidden State	82.38%	84.04%	900
Hierarchical Weighted State 1	80.85%	82.79%	500
Hierarchical Weighted State 2	81.89%	83.53%	1100
Hierarchical Weighted State 3	80.66%	82.58%	900

Table 5: Diagnostic Classifier results for Word Sentiment

Layer	Accuracy (%)	F1 (%)	Number of Neurons
Embedding	77.03%	80.81%	900
Hidden State	67.84%	73.69%	900
Hierarchical Weighted State 1	66.82%	72.95%	700
Hierarchical Weighted State 2	63.13%	70.27%	1100
Hierarchical Weighted State 3	66.00%	72.01%	900

Target-Related Sentiment Classification. Table 6 shows the results for the diagnostic classification of the Target-Related Sentiment Classification task, which has to check if the neural network can predict the sentiment of the words specifically related to the target. Table 6 shows that the accuracy is highest in the hidden state layer and falls as we move deeper into the neural network, before rising again in the final layer. However, the accuracy never increases past the hidden state layer. The weighted F1 score follows a similar pattern, although it is much less pronounced for the spike in the final layer. As this hypothesis is a combination of two other hypotheses, its trend can be explained

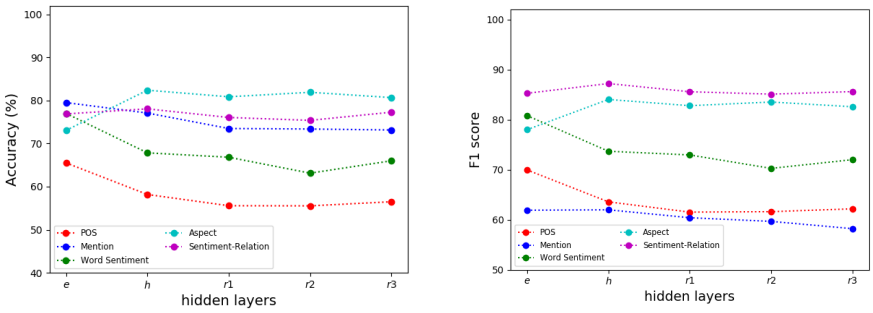
through them. We observe, that the Aspect Relation accuracy increases and then stabilizes but for the Word Sentiment hypothesis it decreases before a spike in accuracy at the end. The increase in accuracy for the hidden state layer is possibly due to the increase in the layers' ability to identify words related to the target being greater than the fall in its ability to identify the sentiment. Furthermore, as the accuracy for Aspect Relations stabilizes, but the accuracy for the word sentiment hypothesis continues to fall, we observe a downward trend for the layers following the hidden state. However, the final spike can be explained by the spike in accuracy for the Word Sentiment hypothesis, while the accuracy of the Aspect Relation hypothesis remains approximately the same. We observe that the neural network places more importance on identifying the sentiment of the words related to the aspect, as we observe a relatively good accuracy for Target-Related Sentiment Classification in the final layer, which is within expectations as that is an important task for ABSC. The optimal number of neurons for each classifier is given in Table 6.

Table 6: Diagnostic Classifier results for Target-Related Word Sentiment

Layer	Accuracy (%)	F1 (%)	Number of Neurons
Embedding	76.88%	85.27%	500
Hidden State	78.05%	87.22%	700
Hierarchical Weighted State 1	76.05%	85.58%	700
Hierarchical Weighted State 2	75.38%	85.10%	1100
Hierarchical Weighted State 3	77.28%	85.61%	500

5.1 Overview

Figure 3a and Figure 3b show the accuracy and F1 scores, respectively, for the different hypotheses in a single graph. We can see in Figure 3b that the model is successful at learning about Aspect Relations, Word Sentiments, and the sentiment of the word if it is related to the target (Target-Related Word Sentiment). This is a good sign as these tasks are extremely important for ABSC. A major difference between Figure 3b and Figure 3a is that the Mention Tagging hypothesis is performing the worse when compared using the weighted F1 score but good when comparing based on the accuracy. A reason for this disparity in results could be due to the data imbalance and the fact that the Mention Tagging dataset is much smaller compared to the other hypotheses datasets due to the limited coverage of the ontology. The performance for POS tagging and Mention Tagging is low, based on the weighted F1 score, which suggests that the model is not able to encode information about the structure of the sentence and which Aspect Mention a word is related to. These results are to be expected as these tasks are not important for ABSC, as identifying the sentiment supersedes POS tagging and the Aspect Mentions are usually already identified.



(a) Accuracy. **(b)** F1 score.

Fig. 3: Overview of the Accuracy and F1 score for the different hypotheses.

From these results, we can conclude that while the LCR-Rot-hop++ model learns about the word sentiment and structure of the sentence in the starting layers, the more complex details such as which words are related to the target and the sentiment of those words are learnt deeper into the model.

6 Conclusion

In this study, we proposed the use of diagnostic classifiers to investigate if the hidden layers in the LCR-Rot-hop++ model can encode information regarding various hypotheses that are important for ABSC. These hypotheses are:

- POS tagging: We found that the BERT embeddings were the best in classifying POS tags, while the other layers had significantly lower accuracies and F1 scores. This implied that deeper into the model, information about the POS tags is not encoded. According to the weighted F1 score, the LCR-Rot-hop++ model does not capture information about the structure of the sentence.
- Mention Tagging: We found that the accuracy and weighted F1 score significantly fell deeper into the model. This implied that the neural network does not encode information about the Aspect Mention related to the word. The best accuracy for mention tagging was found in the embedding layer. This also suggested that the model did not find this information important as we lose this information as we proceed deeper into the network.
- Aspect Relation Classification: The neural network was able to encode information regarding which words are related to the target. We found relatively high accuracy and weighted F1 score. The weighted F1 score and the accuracy rose deeper into the network and stabilized at the hierarchical weighted layers. This means that the network was able to learn information about which words are related to the targets.

– Word Sentiment: The ability to identify the sentiment of a word fell as we went deeper into the neural network. The best accuracy and weighted F1 score were for the embedding layer. The relatively high accuracy and weighted F1 score for the embedding layer could be due to the contextualization. Overall, the LCR-Rot-hop++ showed moderate success in encoding information regarding the word sentiments.
– Target Related Word Sentiment: We found that the hidden state layer had the highest accuracy for the ability to identify words that are related to the target and then their sentiment. As we moved deeper to the network it fell for a bit before once again rising. Overall, we found that the neural network is able to encode information regarding the sentiments of the words related to the target the best, which is within expectations as this information is highly relevant for ABSC.

In the future, this research should be repeated for different neural networks designed for ABSC, as that might give insight into what kind of neural network works best for certain hypotheses. Furthermore, for the Mention Tagging hypothesis, a multi-class, multi-label diagnostic classifier could be trained to account for one word being related to multiple Aspect Mentions. In addition, as imbalanced datasets are present in the real world, we should look to combining the model with more advanced re-sampling techniques, such as Condensed Nearest Neighbor [12]. It is to be noted that this procedure must be done carefully, as certain oversampling techniques, such as SMOTE [4] and its variants, generate synthetic data and adding synthetic data is counter-intuitive as we want to investigate if the hypothesis is encoded in the layers originally. The final suggestion would be to explore how and where the neural network learns other concepts represented in the ontology besides the aspect mention (e.g., sentiment expressions).

References

1. Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., Goldberg, Y.: Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In: 2017 International Conference on Learning Representations (ICLR 2017) (2016)
2. Barbalau, A., Cosma, A., Ionescu, R.T., Popescu, M.: A generic and model-agnostic exemplar synthetization framework for explainable AI. In: 31st European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2020). LNCS, vol. 12458, pp. 190–205. Springer (2020)
3. Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., Glass, J.R.: What do neural machine translation models learn about morphology? In: 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017). pp. 861–872. ACL (2017)
4. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16**, 321–357 (2002)
5. Chrupała, G., Alishahi, A.: Correlating neural and symbolic representations of language. In: 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019). pp. 2952–2962. ACL (2019)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)

7. Hupkes, D., Zuidema, W.: Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure (extended abstract). In: 27th International Joint Conference on Artificial Intelligence (IJCAI 2018). pp. 5617–5621. International Joint Conferences on Artificial Intelligence Organization (2018)
8. Jumelet, J., Hupkes, D.: Do language models understand anything? On the ability of LSTMs to understand negative polarity items. In: 2018 EMNLP Workshop: Analyzing and Interpreting Neural Networks for NLP (BlackBox NLP 2019). pp. 222–231. ACL (2018)
9. Kenter, T., de Rijke, M.: Short text similarity with word embeddings. In: 24th ACM International on Conference on Information and Knowledge Management (CIKM 2015). pp. 1411–1420. ACM (2015)
10. Liu, B.: Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press, 2 edn. (2020)
11. Meijer, L., Frasincar, F., Truşcă, M.M.: Explaining a neural attention model for aspect-based sentiment classification using diagnostic classification. In: 36th Annual ACM Symposium on Applied Computing. pp. 821–827. SAC 2021, ACM (2021)
12. More, A.: Survey of resampling techniques for improving classification performance in unbalanced datasets. arXiv preprint arXiv:1608.06048
13. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S.M., Eryiğit, G.: SemEval-2016 task 5: Aspect based sentiment analysis. In: the 10th International Workshop on Semantic Evaluation (SemEval 2016). pp. 19–30. ACL (2016)
14. Schouten, K., Frasincar, F.: Survey on aspect-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering **28**(3), 813–830 (2016)
15. Schouten, K., Frasincar, F.: Ontology-driven sentiment analysis of product and service aspects. In: 15th Extended Semantic Web Conference (ESWC 2018). LNCS, vol. 10843, pp. 608–623. Springer (2018)
16. Shiliang, Z., Xia, R.: Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention. arXiv preprint arXiv:1802.00892 (2018)
17. Truşcă, M.M., Wassenberg, D., Frasincar, F., Dekker, R.: A hybrid approach for aspect-based sentiment analysis using deep contextual word embeddings and hierarchical attention. In: 20th International Conference on Web Engineering (ICWE 2020). LNCS, vol. 12128, pp. 365–380. Springer (2020)
18. Wallaart, O., Frasincar, F.: 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, pp. 363–378. Springer (2019)
19. Zhang, Z., Wu, Y., Zhao, H., Li, Z., Zhang, S., Zhou, X., Zhou, X.: Semantics-aware BERT for language understanding. In: 34th AAAI Conference on Artificial Intelligence (AAAI 2021). pp. 687–719. AAAI Press (2020)