Local Interpretation of Deep Learning Models for Aspect-Based Sentiment Analysis

Stefan Lam[a], Yin Liu[a], Max Broers[a], Jasper van der Vos[a], Flavius Frasincar[a,*], David Boekestijn[a], Finn van der Knaap[a]

[a]*Erasmus University Rotterdam, PO Box 1738, 3000 DR, Rotterdam, the Netherlands*

Abstract

Currently, deep learning models are commonly used for Aspect-Based Sentiment Analysis (ABSA). These deep learning models are often seen as black boxes, meaning that they are inherently difficult to interpret. To improve deep learning models, it is crucial to understand their inner workings. We aim to interpret black box models by implementing model-agnostic local interpretation methods. Inspired by Local Interpretable Model-agnostic Explanations (LIME) and Local Rule-based Explanations (LORE) and combined with a Similarity-based Sampling (SS) method, we propose SS-LIME and SS-LORE, and use Anchor to explain two state-of-the-art ABSA deep learning models. The deep learning models build upon the Left-Center-Right separated neural network with Rotatory attention (LCR-Rot) model, extended by iterating multiple times over the rotatory attention mechanism (LCR-Rot-hop) and hierarchical attention and context-dependent word embeddings (LCR-Rot-hop++). We evaluate the proposed models in terms of fidelity, hit rate, and user interpretability using the SemEval 2016 dataset consisting of restaurant reviews for ternary sentiment classification. Results show that the LCR-Rot-hop and LCR-Rot-hop++ models are best explained by SS-LIME and SS-LORE, respectively. Furthermore, we conclude that the LCR-Rot-hop++ model can be better interpreted than the LCR-Rot-hop model.

Keywords: Machine learning, Model-agnostic interpretation models, Textual data, Sentiment analysis

*Corresponding author; tel: +31 (0)10 408 1340; fax: +31 (0)10 408 9162

Email addresses: 481922sl@student.eur.nl (Stefan Lam), 473979yl@student.eur.nl (Yin Liu), 471390mb@student.eur.nl (Max Broers), 481019jv@student.eur.nl (Jasper van der Vos), frasincar@ese.eur.nl (Flavius Frasincar), boekestijn@ese.eur.nl (David Boekestijn), 573834fk@student.eur.nl (Finn van der Knaap)

1. Introduction

The increase in Social Web popularity causes tremendous growth in online reviews. In these reviews, customers express their opinions on a certain product or service they have used. These opinions can be categorized using the sentiment of the review, which we are able to determine based on the interpretation and classification of the emotions the text reflects (Cambria et al., 2011, Susanto et al., 2020). Certain words can influence whether a review is positive, negative, or neutral. Aspect-Based Sentiment Analysis (ABSA) (Schouten and Frasincar, 2016) techniques identify the sentiment of a review and link the sentiment to the corresponding aspects, like specific products or services (Liu, 2015, Cambria, 2016). In order to process large amounts of customer reviews, businesses often use automated approaches, frequently based on machine learning (Salehan and Kim, 2016, Saggi and Jain, 2018).

However, some machine learning models, such as deep neural networks, are highly complicated. Such models are often seen as black boxes, which means that there is little knowledge of how the model determines the output from a given input. This creates a lack of transparency within the model, making algorithms harder to understand. It leads to a risk of machine learning models using wrong reasoning to make a prediction. Since machine learning techniques are being integrated into modern life for Decision Support Systems (DSSs) more frequently, such as those in finance (Kraus and Feuerriegel, 2017), the medical sector (Panigutti et al., 2021), and sentiment analysis (Rana et al., 2021), it is important to correctly interpret these models to build user trust (Giboney et al., 2015). Thus, to improve complex models, methods are needed to interpret them. Moreover, transparency about data classification corresponds to the General Data Protection Regulation (EU, 2016), which states that customers have the right to know how companies make their predictions. Interpretability is thus relevant for both companies and consumers.

In this paper, we focus on ABSA in the restaurant domain. Wallaart and Frasincar (2019) introduce a state-of-the-art hybrid model for ABSA, which combines a domain ontology approach with the LCR-Rot-hop deep learning model. Truşcă et al. (2020) extend this deep learning model by adding a so-called hierarchical attention layer to the LCR-Rot-hop model, which is referred to as the LCR-Rot-hop++ model. These hybrid models are highly accurate for sentiment classification and could also be used in a DSS setting. For example, managers of restaurants might find it useful to know whether reviews about various aspects of their restaurant are negative, neutral, or positive,

such that they can make decisions accordingly.

However, the high accuracy of these models is not a good indication of how well we understand them, as deep learning models contain a huge number of parameters and are thus often not interpretable. Hence, the high accuracy could be based on wrong reasoning. For managers, it is important to know the reasoning behind the decisions made by a model, as this increases their trustworthiness in the model. For this reason, researchers have worked on explainable Artificial Intelligence (AI) solutions in order to explain such black box deep learning models (Arrieta et al., 2020).

Specifically, we focus on different model-agnostic methods to compare the interpretability of state-of-the-art hybrid models for ABSA in the restaurant domain. In particular, we aim to create local faithful interpretation models that explain the predictions made by these hybrid models. We focus on local interpretability as opposed to global interpretability, since it offers a more detailed explanation of why a certain instance is classified in a particular way. We formulate the following research question: *Which state-of-the-art hybrid model for ABSA can be best interpreted by a local interpretation model?* To answer this research question, we formulate the following subquestion: *Which local interpretation models give the best interpretation for each of the state-of-the-art hybrid models for ABSA?* To answer the subquestion, we propose three different methods to create interpretation models for the deep learning part of the hybrid models. These methods are adapted versions of existing methods. Adaptations are needed since the methods are originally defined for binary classification, while we work with ternary classification (positive, negative, and neutral).

The first method creates a linear interpretation model inspired by Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016). Complementary to this method we propose the Weighted Submodular Pick (WSP) algorithm, which is a weighted version of the Submodular Pick (SP) algorithm (Ribeiro et al., 2016). This algorithm selects the most representative instances to be locally interpreted. The other methods create rule-based interpretation models and are thus nonlinear. The first rule-based method is Anchor (Ribeiro et al., 2018). Anchors are sets of features able to explain predictions made by deep learning models using if-then rules. They explain an instance by analyzing a set of local instances and determining the most important features for the classification of the aspect. The last method is based on Local Rule-based Explanations (LORE)

(Guidotti et al., 2018a). The LORE method uses a decision tree classifier to extract decision rules and counterfactual rules. These rules provide us with similar explanations to Anchor but also consider counterfactuals.

In this paper we implement LIME and LORE with a Similarity-based Sampling (SS) method, which is based on part-of-speech (POS) tags and word embedding similarity between different features (Ribeiro et al., 2018). Our implementations are referred to as SS-LIME and SS-LORE. For the implementation of the proposed methods, we use the same datasets as in the works of Wallaart and Frasincar (2019) and Truşcă et al. (2020). More specifically, we use the SemEval 2016 Task 5 Subtask 1 Slot 3 (Pontiki et al., 2016) dataset, consisting of restaurant reviews, on model-agnostic local classifiers for interpretability.

We extend the current literature as follows:

- We utilize textual data instead of tabular data used by Guidotti et al. (2018a).
- We work with ternary classification (negative, neutral, positive) as opposed to binary classification (negative, positive) in previous literature (Ribeiro et al., 2016).
- We provide a homogeneous comparison between the implemented methods through the use of the same sampling method.
- We extend the SP algorithm to the WSP algorithm by adding a weighted component to better cater for local interpretability.
- We find that the LCR-Rot-hop and LCR-Rot-hop++ models can be best explained by SS-LIME and SS-LORE, respectively. Furthermore, The SS-LIME method is able to highlight the most important features. However, SS-LORE is more user-friendly than SS-LIME, since it provides us with explanations consisting of the reasons for a decision, and the changes of features leading to another decision.

The rest of the paper is structured as follows. In Section 2 we discuss relevant related work. Section 3 provides an overview of the data we use. Section 4 describes the used framework and our proposed methods and we evaluate the results of our research in Section 5. Last, Section 6 presents conclusions, discusses limitations, and suggests future research directions.

2. Related Work

With the recent advancements in sentiment analysis and more specifically ABSA, models are more and more referred to as black boxes, which implies that there is little knowledge of how the models generate the output from a given input. This leads to a lack of transparency in models, making it more difficult to comprehend them. First, in Section 2.1, we delve into state-of-the-art ABSA models, after which we shift our focus to existing techniques that focus on the interpretability of such black box models in Section 2.2.

2.1. State-of-the-art ABSA Models

Three categories of algorithms exist for ABSA (Schouten and Frasincar, 2016, Brauwers and Frasincar, 2023): knowledge-based approaches, machine learning approaches, and hybrid approaches. For knowledge-based algorithms, a sentiment dictionary is often used to find the sentiment score of a particular word. Then, the sentiment scores are used to determine the combined sentiment score of all words relevant to an aspect (Schouten and Frasincar, 2016). One such knowledge base is SenticNet, which is built using symbolic and subsymbolic AI methods (Cambria et al., 2022).

Machine learning approaches, in particular deep learning approaches, have recently shown great potential for ABSA. Because of their flexibility, deep learning models often provide better results than knowledge-based approaches for ABSA when a large annotated dataset is available. Many of these deep learning models make use of an attention mechanism, which allows them to focus on the relevant sentence features in the context of aspects (Zhang et al., 2019, Wang et al., 2020, Liang et al., 2022). The two approaches can be easily combined in hybrid methods.

The two proposed state-of-the-art hybrid models by Wallaart and Frasincar (2019) and Truşcă et al. (2020) use an ontology approach in combination with a deep learning model: the LCR-Rot-hop or the LCR-Rot-hop++ model. We explain the common structure of the deep learning models as well as their differences. When the ontology approach fails to detect the sentiment, the hybrid model uses a deep learning model as a backup to determine the sentiment. In this paper, we consider LCR-Rot-hop and LCR-Rot-hop++ as backup models. These models use a Left-Center-Right separated neural network with Rotatory attention (LCR-Rot) (Zheng and Xia, 2018). The LCR-Rot splits sentences into three parts: the aspect, the left context, and the right context. Then,

it assigns attention weights to capture the most important words in a context (Zheng and Xia, 2018). Wallaart and Frasincar (2019) extend the LCR-Rot model by iterating multiple times over the rotatory attention mechanism, creating the LCR-Rot-hop model. Truşcă et al. (2020) further extend the LCR-Rot-hop model by adding hierarchical attention to represent input sentences; this model is referred to as LCR-Rot-hop++. Another difference between the two deep learning models is the used word embeddings. The LCR-Rot-hop model uses the context-independent GloVe embeddings (Pennington et al., 2014), while the LCR-Rot-hop++ model uses the context-dependent Bidirectional Encoder Representations from Transformers (BERT) embeddings (Devlin et al., 2019).

2.2. Interpretability of Black Box Models

To explain a black box model, such as the previously mentioned deep learning models, an interpretation model has to be faithful to the model, i.e., the interpretation model has to show similar behavior as the black box model (Ribeiro et al., 2016). A distinction is made between global and local interpretability (Du et al., 2019, Guidotti et al., 2018b). Global interpretability aims to understand the structure and parameters of a model in its entirety. However, this requires the interpretation model to be faithful to all instances of the black box model, which is impractical (Ribeiro et al., 2016). In contrast, local interpretability explains how individual predictions are made (Luo et al., 2024). The advantage of this approach is that the interpretation model only has to be locally faithful, i.e., it only has to behave the same as the black box model in the vicinity of the instance being explained (Ribeiro et al., 2016).

Du et al. (2019) introduce the post-hoc approach to create an interpretation model. This approach is based on a separate model constructed to provide explanations of the original model. The post-hoc approach is model-agnostic since it does not depend on the structure of a black box model and can thus be implemented on all models (Madsen et al., 2023). This approach does not affect the underlying model accuracy, but it is sometimes difficult to interpret the separately constructed model (Du et al., 2019). In this paper, we focus on the post-hoc approach by using different model-agnostic methods to create locally faithful interpretation models able to explain the LCR-Rot-hop and LCR-Rot-hop++ models. Previously, we proposed post-hoc approaches using diagnostic classifiers for global interpretation models of LCR-Rot-hop and LCR-Rot-hop++

(Meijer et al., 2021, Geed et al., 2022).

In current literature, several methods are able to provide us with local interpretation models explaining the inner workings of a black box model for a particular instance. The first method we study is LIME (Ribeiro et al., 2016). LIME is able to locally explain a black box by estimating a local interpretation model for each instance. The estimation is done by first uniformly sampling local instances in the vicinity of an instance and then estimating the interpretation model on this sample. However, this process does introduce instability in explanations, as repeated explanations with the same parameters might vary due to the sampling process. Several extensions have been proposed to address the above problem (Zhou et al., 2021, Tan et al., 2023). Yet, to make a fair comparison with the work of Ribeiro et al. (2016), we focus on LIME. Achieving a global explanation for the black box model is difficult with LIME since we have to consider all instances. Ribeiro et al. (2016) still aim to give a global explanation by picking a set of instances to explain with the highest global importance (best coverage of the features) according to the SP algorithm. We propose the WSP algorithm, which extends the SP algorithm by incorporating the local importance as weights when picking instances for global importance.

Another method mentioned in current literature is the Anchor method (Ribeiro et al., 2018). Anchor greedily selects a set of features that explain a black box model by comparing various sets of features in their coverage and precision. Another rule-based method is LORE (Guidotti et al., 2018a), which is used to create explanations that consist of a decision rule and a counterfactual rule. Both rules are extracted from a decision tree classifier.

Decision rules are of the form: "if X had occurred, then Y would have occurred", while counterfactuals are of the form: "if X had not occurred, then Y would not have occurred". LORE has a high level of applicability that obeys the domain constraints since humans want to know which changes lead to another prediction (Molnar, 2019). A benefit of this method is that it provides us with high-level and minimal-change contexts to modify the prediction (Guidotti et al., 2018a) compared to LIME or Anchor. We use restaurant reviews for our research and thus apply LORE to textual data, in which we extend the original paper of Guidotti et al. (2018a).

Pastor and Baralis (2019) introduce the Local Agnostic attribute Contribution Explanation (LACE) method, which is able to capture the joint effect of features on an instance. From the LACE method, we only implement the prediction differences defined by Pastor and Baralis (2019)

to evaluate the influence of the rules produced by Anchor and LORE on the prediction of an instance. Due to the high similarity of LACE and LORE, we only focus on the LORE method (they are both decision tree-based methods).

Mei et al. (2023) propose a disentangled linguistic graph model using signals to enhance transparency and explainability. The authors propose a knowledge-based approach, injecting extra external information into the model to help capture feature representations and boost explainability. Both Zhao and Yu (2021) and Dekker et al. (2023) inject external knowledge into the BERT model to better capture sentiment relations, thus focusing on intrinsic explainability (Du et al., 2019). In this paper, contrary to the above literature, we focus on a post-hoc approach, which does not change the underlying representation of a model. Post-hoc approaches do not make use of external information but change the surroundings of a model, keeping the underlying model accuracy the same.

Another stream of literature delves into attention mechanisms and if we can leverage the potential interpretability of those, especially as they are used by many state-of-the-art neural models for natural language processing. To be precise, Jain and Wallace (2019) explore the relationship between attention weights and model outputs, finding that the ability of attention weights to provide transparency or interpretability is limited. Yet, Wiegreffe and Pinter (2019) challenge many of the assumptions made by Jain and Wallace (2019), and propose four alternative tests to discern whether attention weights do not provide interpretability. Wiegreffe and Pinter (2019) show that previous literature does not disprove the functionality of attention weights when it comes to explainability. However, Bastings and Filippova (2020) question the use of attention for interpretability compared to, for example, saliency methods, showing that for user interpretability and explainability, there is no reason to use attention weights over saliency methods.

A branch of saliency methods focuses on gradient-based approaches, implying that these approaches use the gradient of a method's output to calculate the sensitivity of a model to changes in the input. For example, Bykov et al. (2022) introduce NoiseGrad, a method that introduces stochasticity into the weights of the model by drawing samples from tempered a Bayes posterior (Wenzel et al., 2020), which perturbs the decision boundary. Whilst the proposed method enhances local and global explanations compared to the considered baselines, it is computationally expensive as it suffers from increasing parameter spaces. However, gradient-based methods also have

possible shortcomings. Srinivas and Fleuret (2021) show that such methods are prone to capture information relating to the implicit density model, not the underlying black box model.

In this paper, we keep the comparison between Anchor, SS-LIME, and SS-LORE fair by using the same sampling method to generate the local instances. Specifically, we use a version of the SS method (Ribeiro et al., 2018), exploiting contextualized BERT embeddings to create local instances in a neighborhood. Compared to the uniform sampling method (Ribeiro et al., 2016), this method aims to generate a diverse proportion of positively, negatively, and neutrally labeled local instances.

Yet, the literature also proposes other methods of sampling in a neighborhood. Botari et al. (2020) introduce Meaningful LIME (MeLIME), which, in contrast to the work of Ribeiro et al. (2016), leverages the distribution of the data used to train the model in order to create more meaningful explanations. As a result, MeLIME is able to create improved explanations on, among others, tabular data and text. Instead, we use BERT to generate more contextualized word embeddings in order to sample around the neighborhood of the instance. In addition, MeLIME is more broad, as we specifically focus on ABSA. Focusing instead on SHapley Additive exPlanations (SHAP) compared to the approaches in this work, Ghalebikesabi et al. (2021) propose Neighborhood SHAP, sampling from a local reference population to create local instances, such that the neighborhood samples not only consider the metric space with regards to the original instance but also the data distribution.

3. Data

This section presents the data used in our research. First, in Section 3.1 we introduce the raw data and discuss how we preprocess the data. Next, Section 3.2 describes descriptive statistics of our data after preprocessing.

3.1. Raw Data and Preprocessing

The data used in this paper correspond to the datasets used by Wallaart and Frasincar (2019) and Truşcă et al. (2020), since we compare the interpretability of these state-of-the-art ABSA models. In particular, we use the SemEval 2016 Task 5 Subtask 1 Slot 3 (Pontiki et al., 2016) training and test datasets. These datasets consist of restaurant reviews with one or multiple sentences represented in the XML markup language. Each sentence contains one or multiple opinions, each

about a certain aspect (target) with a corresponding category, and with a polarity that expresses whether the reviewer is positive, negative, or neutral towards this aspect. Sentences containing more than one aspect are considered multiple times to classify the sentiment of each aspect. The XML data is preprocessed by first removing the opinions where the target is implicit (Wallaart and Frasincar, 2019) since this paper focuses on ABSA deep learning models that need a target to split the sentence into three parts. There are 657 (25%) sentences with implicit targets in the training dataset and 209 (24.3%) in the test dataset which we remove. The remaining sentences are processed using the NLTK platform (Bird et al., 2009). The data is then tokenized, and all the words are lemmatized using the WordNet lexical database (Miller, 1998). We could also use the SemEval 2015 dataset (Pontiki et al., 2015) in the experiments, but given that this one is contained in SemEval 2016 we only use the larger SemEval 2016 dataset.

3.2. Data Characteristics

The polarity frequencies of the SemEval 2016 train and test datasets after preprocessing are given in Table 1. We notice that the positive opinions dominate the negative and neutral opinions by a large percentage. Especially the neutral opinions are in the minority.

Table 1: Polarity counts and frequencies of the SemEval 2016 datasets.

	Polarity Counts and Frequencies			
	Positive	Neutral	Negative	Total
Train data	1319 (70.2%)	72 (3.8%)	488 (26.0%)	1879 (100.0%)
Test data	483 (74.3%)	32 (4.9%)	135 (20.8%)	650 (100.0%)

Furthermore, the LCR-Rot-hop model uses GloVe word embeddings (Pennington et al., 2014) with a dimensionality of 300, while the LCR-Rot-hop++ model uses BERT word embeddings (Devlin et al., 2019) with a dimensionality of 768. We initialize the embeddings of the words not appearing in the GloVe vocabulary with a normal distribution $N(0, 0.05^2)$ (Wallaart and Frasincar, 2019), while the out-of-vocabulary words of BERT are initialized by averaging the embeddings of the corresponding subwords (Devlin et al., 2019).

4. Methodology

To explain the LCR-Rot-hop and LCR-Rot-hop++ models, we first train them on the SemEval 2016 training dataset. Next, we implement different interpretation models for the trained models to explain the predictions made on the instances of the test dataset. An overview of the notations used in this section is provided in Table 2.

This section elaborates on the methods used in this paper. First, Section 4.1 explains the general structure of our local interpretation models. Section 4.2 describes the SS method. Then, Section 4.3 describes the local interpretation models. Finally, in Section 4.4 we present the performance measures used to compare the local interpretation models.

4.1. General Approach of Local Interpretation Models

The methods we use (SS-LIME, Anchor, and SS-LORE) all try to learn the local behavior of a deep learning model around an instance $x \in X$. To learn this we draw a perturbed sample Z of M local instances $z \in Z$, where a local instance z is similar to x except for small changes in the features. In our research, x is (part of) a sentence from a restaurant review consisting of f features (words). The aspect has been excluded from these features since the sentiment towards the aspect has to be explained (Wallaart and Frasincar, 2019). In this paper, $b(x) = p$ is referred to as the predicted sentiment p by the black box model b and $m \in \{\text{SS-LIME, Anchor, SS-LORE}\}$ as the used interpretation model. In this research, the black box model is either the LCR-Rot-hop or the LCR-Rot-hop++ model.

Let $|X| = N$ denote the number of available instances and ξ_x^m be the local explanation of instance x by the interpretation model m. For the LCR-Rot-hop and LCR-Rot-hop++ models, x is represented as a sequence of f word embeddings, which is not easily interpretable and thus difficult to change. For this reason, we define $x' \in \{0,1\}^{|F|}$ as the interpretable representation of x, where F is the set consisting of all different words in the SemEval 2016 datasets. Consequently, we define the set Z' consisting of M interpretable local instances $z' \in \{0,1\}^{|F|}$. The interpretable instance x' can be interpreted as a binary vector indicating the presence or absence of features. The interpretable representation x' is necessary because we are only able to explain the prediction made by a black box model if we are able to interpret the predicted instance x first. The general

Table 2: Overview of used notations.

X	Set of original instances
N	Number of available instances ($\|X\|$)
Z	Set of local instances around $x \in X$
Z'	Set of binary representations of local instances $z \in Z$
M	Local neighborhood size of instances generated around $x \in X$
ξ_x^m	Local explanation of instance x by the interpretation model m
W	Word in a sentence
β_j^k	Marginal effect of the jth feature to indicate which class an original instance is classified as
e_{ij}	Local influence of the jth feature on the ith instance
e_j	Total absolute marginal effect of the jth feature on all classes
I_j	Global importance of the jth feature across all the instances
Λ	Set of R instances from X selected by a picking algorithm
Ξ	Influence matrix
F	Set of all different words in the SemEval 2016 datasets (vocabulary)
S	Maximum number of features to be considered for local explanation
$\Psi(\Lambda)$	Set of features contained in the instances of Λ
$A(x)$	Set of features influencing the sentiment of x (anchor)
τ	Minimum precision of $A(x)$
$b(x)$	Black box prediction for x
δ	Width of confidence
ϵ	Tolerance of confidence
$\pi_x(z)$	Proximity kernel between x and z
σ	Width of proximity kernel
$\mathcal{A}_B$	Set of B best anchors
J	Maximum number of iterations for the Anchor algorithm
C	Decision tree classifier
d	Maximum depth of decision tree
Φ	Set of counterfactual rules

approach for explaining the prediction of an instance x is given in Algorithm 1.

Algorithm 1 General approach: Explaining the prediction of instance x

Input: The black box b, interpretation model m, and the instance $x \in X$

$Z' \leftarrow \emptyset$ ▷ Initialization of the set of local interpretable instances

for $i \in \{1, 2, ..., M\}$ **do**

  $z'_i \leftarrow sample\_around(x)$ ▷ Sampling a local instance z'_i around x

  $Z' \leftarrow Z' \cup z'_i$

end

$\xi_x^m \leftarrow m(Z', b)$

return ξ_x^m

To further illustrate the general approach, Figure 1 provides a flowchart indicating the relations between our implemented methods. The SS method, which is used to sample around the instance x, is further discussed in Section 4.2. Lastly, the specification of each local explanation ξ_x^m is provided in Sections 4.3.1, 4.3.3 and 4.3.4 for SS-LIME, Anchor and SS-LORE, respectively.

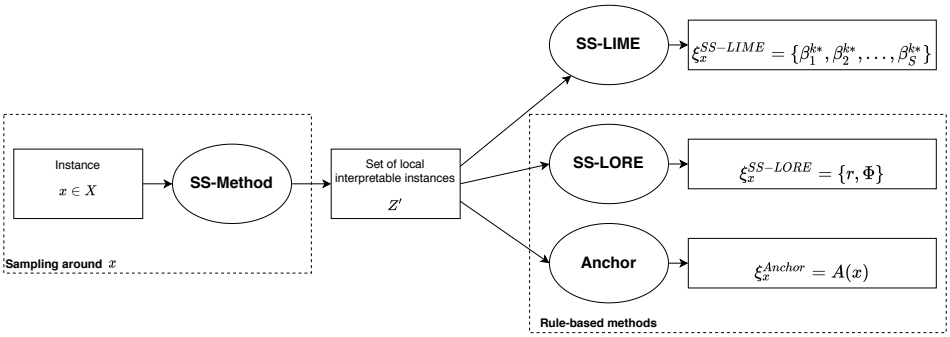


Figure 1: Flowchart of the general approach to locally explain an instance $x \in X$ with the considered methods: SS-LIME, Anchor, and SS-LORE.

4.2. The Similarity-based Sampling Method

We generate the set of local instances Z according to the SS method, which is similar to the method used by Ribeiro et al. (2018). This perturbation approach is used for all our methods to keep the performance comparison fair. In particular, we generate the local instances $z \in Z$ by replacing the features in x with different features that have the same POS tag according to a perturbation distribution $\mathcal{D}$. To be precise, new features are chosen with a probability that is proportional to

the similarity of the original and new features in an embedding space. We consider the following POS tags: *NOUN*, *VERB*, *ADJ*, *ADV*, *ADP*, and *DET*. These POS tags represent the noun, verb, adjective, adverb, adposition, and determiner, respectively. The similarity between two words W_1 and W_2 is calculated with the cosine distance function D and the corresponding word embeddings. The distance function is given as:

$$D(W_1, W_2) = \frac{W_1 \cdot W_2}{\|W_1\|\|W_2\|}, \tag{1}$$

which measures the similarity between the words W_1 and W_2 based on the angle of the corresponding word embeddings (Molnar, 2019). The intuition behind the sampling method is that we ensure that the sampled instances are in the vicinity or neighborhood of the local instance (i.e., the sampled instances have a similar meaning) by replacing words based on the similarity in the word embedding space, given that the POS tags are the same such that the semantic structure of the sentence does not change. Algorithm 2 gives the procedure for generating a local interpretable instance z' according to the SS method.

To generate a local interpretable instance z' for instance x, we first generate a local instance z by calculating the distances between a word $W_1 \in x$ and the words $W_2 \in F$ with the distance function given in Equation (1). Then, for a word W_1, we pick the top $n = 50$ similar words in F with the same POS tag. We consider these n words as possible replacements for the word W_1. To pick a replacement we randomly choose between these n words, where the distances between W_1 and the n words are used to increase the probability of choosing a word. The generated local instance z can then be easily transformed to a binary interpretable representation z'. However, the LCR-Rot-hop and LCR-Rot-hop++ both use different word embeddings. As a result, the generated local instances for these models are different.

Algorithm 2 Similarity-based Sampling method to generate a local interpretable instance z'

Input: Instance $x \in X$

$z \leftarrow \emptyset$ ▷ Initialization of the local instance z

for $W_1 \in x$ **do**

  $distances \leftarrow \emptyset$ ▷ Set of distances for the word $W_1 \in x$

  for $W_2 \in F$ **do**

    $distances \leftarrow distances \cup D(W_1, W_2)$ ▷ Distance according to Equation (1)

  end

  $top\_n,\ distances\_n \leftarrow pick\_top\_n(distances)$ ▷ Top n similar words with the distances

  $W \leftarrow random\_choice(top\_n,\ distances\_n)$ ▷ Picking a random word W

  $z \leftarrow z \cup W$

end

$z' \leftarrow transform(z)$

return z'

4.3. Local Interpretation Model

In this section, we describe the used local interpretation models and the (W)SP algorithm, which provide an interpretation of the sentiment prediction by the deep learning models.

4.3.1. SS-LIME

SS-LIME is an adaptation of the LIME method (Ribeiro et al., 2016), which uses a local log-linear interpretation model for a one-vs-all logistic regression to explain the sentiment prediction of an instance x. In our case, the interpretation model corresponds to a classification problem with three classes $k \in K = \{\text{-1, 0, 1}\}$ denoting the negative, neutral, and positive sentiment, respectively. Multinomial logistic regression (Bishop, 2006) is a technique that essentially trains a single model $g^k(x')$ for each class k by solving $|K|$ binary classification problems consisting of two classes: k and the complement k^c. The log-linear interpretation model $g^k(x')$ with x' as the interpretable representation is defined as:

$$g^k(x') = \ln(Pr[b(x) = k \mid x]) = \beta_0^k + \sum_{j \in F} \beta_j^k x'_j - \ln(L), \tag{2}$$

which is the natural logarithm of the probability that the predicted sentiment of the black box

model b towards the aspect of x is classified as k. The coefficients β_j^k denote the marginal effects of the jth feature to indicate whether x is classified as class k or k^c. Let $\boldsymbol{\beta}^k$ be the vector consisting of the coefficients $\{\beta_0^k, \beta_1^k, ..., \beta_{|F|}^k\}$. Then, L is a normalization term defined as $\sum_{k \in K} e^{\boldsymbol{\beta}^k} x'$ which ensures that the set of probabilities $Pr[b(x) = k \mid x]$ forms a probability distribution (Bishop, 2006). Using the normalization term, the probabilities can then be written as:

$$Pr[b(x) = k \mid x] = \frac{e^{\boldsymbol{\beta}^k x'}}{\sum_{c \in K} e^{\boldsymbol{\beta}^c x'}}, \tag{3}$$

which shows that the sign and magnitude of β_j^k determine the effect of the jth feature on the probability of classifying x' as class k. The marginal effects provide us with the local explanation for an instance x. However, local explanations should remain concise to be interpretable by users (Ribeiro et al., 2016). Thus, we select a maximum of S features to be considered for the local explanation. LIME originally uses Lasso regression to select S features, therefore it is possible to select features not contained in the instance x. For the explanations to stay interpretable, we need to select features contained in the instance x. Thus, in contrast to LIME, we first estimate Equation (2) on all features, and we then select a maximum of S features that have the highest influence given that the features are contained in x. These influences can be measured with $e_j = \sum_{k \in K} |\beta_j^k|$, which is the total absolute marginal effect of the jth feature on all classes.

The local explanation of x for the prediction on class k can be given by the following specification $\xi_{x,k}^{SS-LIME} = \{\beta_1^{k*}, \beta_2^{k*}, ..., \beta_S^{k*}\}$, which is the set of marginal effects towards class k with the S highest influences. To determine this explanation we estimate Equation (2) by applying a weighted multinominal logistic regression trained on the set Z' of interpretable local instances z' with label $b(z)$, obtained by the SS method, using the proximity $\pi_x(z)$ between x and z as our sample weights:

$$\pi_x(z) = \exp(-D(x, z)^2 / \sigma^2). \tag{4}$$

Equation (4) represents an exponential kernel function (Molnar, 2019) with the hyperparameter σ as the width of the kernel and $D(x, z)$ defined as the distance function which measures the cosine similarity between the word embeddings of instances x and z. The kernel function measures how close two instances are from each other, consequently giving the highest weights to the local instances z which are closest to x when estimating Equation (2).

4.3.2. (Weighted) Submodular Pick Algorithm

Estimating Equation (2) provides us with a local interpretable model for an instance x. Nevertheless, we still try to obtain an approximate global interpretation by interpreting a set of representative instances for the whole dataset, as this would be desired. We aim to give a global understanding by explaining a set Λ of R instances. Since users cannot check all the instances individually, we introduce the WSP algorithm to solve this problem. This algorithm is similar to the SP algorithm (Ribeiro et al., 2016) but with a difference in the use of local weights when picking the instances.

The SP algorithm tries to pick R instances with the highest coverage across the features, i.e., it picks the R instances containing the highest number of different features, while keeping the global importance of the features as high as possible. Let $\boldsymbol{\Xi}$ denote the $N \times |F|$ influence matrix containing the local influences e_{ij} of the ith instance and jth feature. The global importance of the jth feature across all the instances can be measured as $I_j = \sqrt{\sum_{i=1}^{N} e_{ij}}$ (Ribeiro et al., 2016). Let $\Psi(\Lambda)$ be the set of features contained in the instances of Λ. Then, the SP coverage of these instances is calculated as:

$$\text{cov}^{\text{SP}}(\Lambda, I) = \sum_{j \in \Psi(\Lambda)} I_j, \tag{5}$$

which can also be seen as the total global importance across all the features in $\Psi(\Lambda)$. The SP algorithm iteratively picks instances with the largest gain in the SP coverage. The SP gain in coverage for the ith instance is defined as:

$$G_i^{SP}(\Lambda, \Xi, I) = \sum_{j \in F \setminus \Psi(\Lambda)} \mathbb{1}_{[e_{ij}>0]} I_j, \tag{6}$$

where $\mathbb{1}_{[e_{ij}>0]}$ is an indicator function indicating whether the influence of the jth feature on the ith instance is positive. We apply a weighted version of SP, which considers the local influences e_{ij} when picking instances. To incorporate the local influences, we define the WSP coverage as follows:

$$\text{cov}^{\text{WSP}}(\Lambda, I) = \sum_{j \in \Psi(\Lambda)} I_j \sum_{i \in \Lambda} e_{ij} \leq \sum_{j \in \Psi(\Lambda)} I_j^3, \tag{7}$$

where $\sum_{j \in \Psi(\Lambda)} I_j^3$ is an upper bound on the WSP coverage. Then, the WSP gain in coverage for the ith instance is given by:

$$G_i^{WSP}(\Lambda, \Xi, I) = \sum_{j \in F \setminus \Psi(\Lambda)} e_{ij} I_j, \tag{8}$$

where the local influences e_{ij} ensure that the coverage is also dependent on the local importance of features. Note that we do not have to use the indicator function as in Equation (6) because the local influences are nonnegative and we care about the sizes of the local influences. Both the SP and WSP algorithms aim to maximize the gain in coverage by implementing a greedy algorithm (Ribeiro et al., 2016). The picking algorithms aim to pick R instances that are able to globally explain the black box models. We motivate the use of local weights by stating the following: "A set of picked instances can only globally explain a black box model if we are able to locally interpret these instances first". We make this statement since users generally need a local understanding first before achieving a global understanding. The local weights ensure that only instances are picked where the globally important features are sufficiently highlighted.

4.3.3. Anchor

In this section, we discuss the Anchor method, which is based on if-then rules. In contrast to LIME, rule-based models are nonlinear. The Anchor method takes an instance x and returns an anchor $A(x)$ containing a set of features influencing the sentiment of x regarding an aspect.

We generate the set of local instances Z with the perturbation distribution $\mathcal{D}$ as described in Section 4.2 and we denote $\mathcal{D}(\cdot \mid A)$ as the conditional distribution when a (combination of) word(s) A applies to a certain local instance. $A(x)$ is an anchor for instance x if the set of words in the anchor is contained in x and the anchor has a precision for $b(x)$ with at least a probability τ (Ribeiro et al., 2018). The latter condition is defined as follows:

$$\text{prec}(A(x)) = \mathbb{E}_{\mathcal{D}(z|A(x))}[\mathbb{1}_{b(x)=b(z)}] \geq \tau, \tag{9}$$

where the precision of $A(x)$ is defined as the average number of local instances z which contain $A(x)$ and have the same black box prediction as x.

We construct potential anchors according to the Beam-Search method (Ribeiro et al., 2018). This method uses the KL-LUCB algorithm with multiple arms (Kaufmann and Kalyanakrishnan, 2013). The algorithm works by constructing confidence regions (Barron and Cover, 1991) and selects the B best anchors in an iteration, and the anchor with the highest upper bound (A'), provided it is not one of the B best. In the algorithm, anchors are selected while keeping the precision as high as possible. However, it is intractable to compute the precision given in Equation (9)

directly (Ribeiro et al., 2018). Instead, the algorithm searches for anchors satisfying the precision constraint with a high probability $1 - \delta$:

$$Pr(\text{prec}(A(x)) \geq \tau) \geq 1 - \delta, \tag{10}$$

where δ is the width.

We draw local instances from the perturbation distribution until there is statistical confidence about the precision of our anchors. Statistical confidence is gained when the following holds:

$$\text{prec}_{\text{lb}}(A(x)) \geq \text{prec}_{\text{ub}}(A'(x)) - \epsilon, \ \forall A \in \mathcal{A}_B, \tag{11}$$

where ϵ is the tolerance and $\mathcal{A}_B$ is the set of the B best anchors.

Multiple anchors could meet the criterion given in Equation (10). If this is the case, we choose the anchor with the largest coverage. If there are no anchors that fulfill this constraint, we construct a new set of anchors and find the new B best anchors. Coverage for rule-based models is calculated differently than for linear models, such as SS-LIME. Specifically, the coverage for anchor $A(x)$ is equal to the average number of local instances that contain this anchor:

$$\text{cov}^A(A(x)) = \mathbb{E}_{\mathcal{D}(z|A(x))}[A(z)]. \tag{12}$$

We make two adjustments compared to the method described by Ribeiro et al. (2018). The original algorithm has trouble gaining statistical confidence and can take up to 400 iterations to select the B best anchors. This results in a very large computational load. To make the computational load smaller, we adjust the method for selecting the B best anchors by including a maximum number of iterations J that are allowed to be run. If the algorithm reaches J, our version of the algorithm returns the B best anchors up to that point. This may influence the validity of selecting the B best anchors with statistical confidence, but it greatly lowers the computational load for instances that contain a lot of features. The second adjustment is the initialization of a minimal coverage for every anchor to fulfill. This way, fewer candidate anchors need to be evaluated, reducing the computational load.

4.3.4. SS-LORE

In contrast to the LORE method (Guidotti et al., 2018a), SS-LORE uses the SS method to create a set of local instances to train the decision tree classifier C. We construct the decision tree by using

the C4.5 algorithm (Quinlan, 1992) for ternary classification (negative, neutral, positive), whereas tabular data is used by Guidotti et al. (2018a). From the constructed decision tree, we are able to obtain decision rules $r = (v \rightarrow y)$, where v is the root-leaf path consisting of the split decisions sd at each decision node to reach the leaf node with label y. We specify a maximum depth d for the decision tree to keep the resulting root-leaf paths concise.

SS-LORE uses the decision tree to extract counterfactual explanations. These consist of a decision rule $r = (v \rightarrow y)$ explaining the reasons for a decision, and a set Φ of counterfactual rules $\varphi = (q \rightarrow \hat{y})$ with q the root-leaf path consisting of the split decisions sd to reach the leaf node with a label $\hat{y} \neq y$. These counterfactual rules suggest the changes in the features of an instance leading to a different prediction (Guidotti et al., 2018a). A condition for a counterfactual rule is that it should describe the smallest change in the features of an instance leading to a different prediction (Molnar, 2019). We find the counterfactual rules satisfying this condition with the same procedure as described by Guidotti et al. (2018a). The local explanation of SS-LORE for an instance x can then be given by $\xi_x^{SS-LORE} = (r, \Phi)$.

As extra output, we compute, just as LORE, a set X_q of counterfactual instances x_q from the counterfactual rules. These counterfactual instances correspond to the instance x with the smallest changes in the features of x leading to another prediction according to the counterfactual rules. We create counterfactual instances for each counterfactual rule by minimally changing the features of x, such that it satisfies the split decisions in the path q. The goal of the counterfactual instances is to determine whether the original prediction made by the black box actually changes when we modify an instance according to the counterfactual rules.

4.4. Performance Evaluation

4.4.1. Performance Measures

An interpretation model performs well globally if it is able to mimic the black box model with high certainty for all instances. To measure global interpretation, we use the following performance measures (Hedström et al., 2023):

- **Hit rate**: Compares the prediction made by the black box model b and the interpretation model m on the instances $x \in X$. If these are equal, return 1, otherwise, return 0. To calculate the hit rate, we divide the number of equal predictions by the number of instances

in X. This determines how well an interpretation model is suited to be used for a black box model. The higher the hit rate, the more suitable the interpretation model (Guidotti et al., 2018a).

- **Fidelity**: Compares the prediction of the black box model b and the interpretation models m_x on the local instances in the set Z_x used to train m_x for all instances $x \in X$. If these are equal, return 1, otherwise, return 0. The fidelity is then calculated by dividing the number of equal predictions by the number of local instances in the set Z_x. This measures how faithful an interpretation model m is to the black box model b. A higher fidelity implies that the interpretation model is more faithful (Guidotti et al., 2018a).

The hit rate and fidelity are used to measure the global performance of the interpretation models. A high value for both measures is needed before an interpretation model is able to mimic the black box model with high certainty.

The local explanation for SS-LORE consists of two rules: the decision rule and the counterfactual rule. We therefore calculate the hit rate and fidelity for both rules. However, per definition, the predictions of counterfactual rules do not correspond to the predictions of the black box model. Hence, we compare the predictions on the counterfactual instances $x_q \in X_q$ instead of the instances $x \in X$, such that we are able to compute the hit rate for counterfactual rules. Furthermore, the Anchor method (Ribeiro et al., 2018) constructs rules that always have the same prediction as the black box model for instances $x \in X$. Hence, the hit rate of these rules will be perfect.

4.4.2. User Interpretability

Interpretation models have good user interpretability if users are able to interpret them well for the prediction of an instance. According to Molnar (2019), there is no real consensus about what interpretability is in machine learning. However, Molnar (2019) still attempts to define some criteria that should be satisfied for a good local explanation: (1) the explanation should be concise, (2) the explanation should be understandable for users, and (3) the features highlighting the explanation should be consistent with prior beliefs of the user. Below, some measures are described that help satisfy these criteria.

- **Marginal effects:** For the linear interpretation model, SS-LIME, we measure the relative importance of the features in an instance with the marginal effects obtained by the log-

linear model, as described in Section 4.3.1. These marginal effects are able to highlight the features in an instance with the highest local importance (Ribeiro et al., 2016). The highlighted features contribute the most towards the predicted sentiment of the black box model according to the linear interpretation model.

- **Prediction difference:** For the rule-based interpretation models, Anchor and SS-LORE, we measure the relative importance of the features in an instance with the prediction difference (Pastor and Baralis, 2019). The prediction difference compares features in an instance by estimating the influence of a set of features on the prediction of an instance.

  In particular, the prediction difference describes how much the probability of the sentiment prediction made by $b(x)$ changes when one feature is omitted. However, features are able to jointly influence a prediction. Thus, we estimate the omission of the set ζ consisting of the features in a rule and extend our definition of the prediction difference similar to Pastor and Baralis (2019):

$$\Delta_{\zeta}^{k}(x) = Pr[b(x) = k|x] - Pr[b(x \setminus \zeta) = k \mid x], \tag{13}$$

  where the first term denotes the probability that the sentiment of instance x is classified as k and the second term denotes the probability that x is classified as k when ζ is omitted. These probabilities are calculated with the softmax layer (Wallaart and Frasincar, 2019) of the used deep learning model.

  In summary, Equation (13) calculates the influence of a rule on class k. The prediction differences range from -1 to 1. The larger the prediction difference, the more the feature(s) influence(s) the prediction. If the prediction difference is positive, it means that the omission of the set of features decreases the probability that $b(x) = k$. Consequently, including the set of features increases the probability of $b(x) = k$.

We satisfy the first mentioned criterion for SS-LIME and SS-LORE by setting a maximum of S features to be considered and a depth d to keep the extracted rules concise for the local explanation. However, the Anchor method does not satisfy this criterion. The second and third criteria are harder to satisfy since the understanding and prior beliefs of local explanations differ for different users. It is easier to satisfy the second criterion for the rule-based interpretation models, since these models, Anchor and SS-LORE, consist of the construction of sets of features

indicating a prediction and are thus intuitive to understand (Ribeiro et al., 2018). SS-LIME does not have this convenient structure. Therefore, we use the marginal effect of the features to make the explanation understandable for users. The third criterion cannot be preliminarily satisfied for our interpretation models (Molnar, 2019). However, we aim to check this criterion by using the marginal effects and the prediction differences. Both measures give a relative comparison of feature importance. The third criterion can then be checked by analyzing whether the highlighted features by the marginal effects for SS-LIME, or the prediction differences for Anchor and SS-LORE, are consistent with our prior beliefs.

We determine user interpretability by explaining several instances with our interpretation models, with the objective of giving a global explanation (Ribeiro et al., 2016). It is impractical to pick all instances, thus we pick R instances according to a picking algorithm. We first choose the best picking algorithm by comparing the following picking algorithms: Random Pick (RP), SP, and WSP. The RP algorithm randomly picks an instance, and we use it as a benchmark to compare the picking algorithms. We compare RP, SP, and WSP for the LCR-Rot-hop and LCR-Rot-hop++ models separately, and choose the algorithm that picks the instances with the most highlighted features consistent with the prior beliefs of users. To try and generalize prior beliefs for textual data, we consider them as the most opinionated features towards the target in an instance. Opinionated features are the most important features for sentiment classification since they provide the best description of the target. We compute Fleiss' kappa κ (Fleiss and Cohen, 1973) for our prior beliefs in order to reflect their degree of reliability.

5. Evaluation

In this section, we evaluate the performance of the proposed interpretation models. First, in Section 5.1 we discuss the performance of the hybrid models. Next, Section 5.2 provides the characteristics of the sampled local instances from the SS method. Then, Section 5.3 compares and provides the results of our local interpretation models. The code, used for the implementation of the proposed interpretation models, is written in Python 3.7 and is publicly available at `https://github.com/StefanLam99/Explaining_ABSA`.

5.1. State-of-the-Art Hybrid Models

In order to evaluate the performance of the proposed interpretation models, we first train the deep learning models on the SemEval 2016 training data. Next, ontology reasoning takes place on the test data, resulting in a test accuracy of 85.7%. The remaining test data, which the ontology is unable to classify, corresponds to 45.4% of the original test data. The deep learning models obtain a test accuracy of 83.4% and 84.2% on the remaining test data for LCR-Rot-hop and LCR-Rot-hop++, respectively. Thus, based on the test accuracy, LCR-Rot-hop++ performs better than LCR-Rot-hop. The counts and frequencies of the predicted sentiment are given in Table 3.

Table 3: Polarity counts and frequencies of the test data after the ontology reasoning and the corresponding predicted sentiment by the deep learning models.

	Polarity counts and frequencies			
	Positive	Neutral	Negative	Total
Test data after ontology	218 (73.9%)	19 (6.4%)	58 (19.7%)	295 (100.0%)
Predictions: LCR-Rot-hop	220 (74.6%)	1 (0.3%)	74 (25.1%)	295 (100.0%)
Predictions: LCR-Rot-hop++	240 (81.4%)	1 (0.3%)	54 (18.3%)	295 (100.0%)

In Table 3, we observe that both deep learning models have the same number of predictions towards the neutral sentiment. However, the LCR-Rot-hop model predicts more negative sentiments (74 *vs.* 54) and less positive sentiments (220 *vs.* 240) than the LCR-Rot-hop++ model.

5.2. Characteristics Local Instances

The trained deep-learning models are used to generate the local instances that we use for our interpretation models. Table 4 presents the average proportions of the positively, neutrally, and negatively labeled generated local instances with the SS method for the LCR-Rot-hop and LCR-Rot-hop++ models, where we set $M = 5000$ for each instance.

Table 4: The average proportion of the labels for the generated local instances with the Similarity-based Sampling method.

	Positive	Neutral	Negative
LCR-Rot-hop	71.8%	0.8%	27.4%
LCR-Rot-hop++	88.1%	0.8%	11.1%

Table 4 shows that the SS method is not able to make a sample of local instances with an equal proportion of positive, neutral, and negative labels. Rather, we notice that the neutral-labeled local instances are in a big minority. A reason for the lack of diversity could be the used perturbation distribution since this distribution replaces features only according to the similarity. This might generate different local instances in the vicinity of the original instance. However, the local instances are then still fairly similar to the original instance and thus the black box model is not able to change its prediction often enough for the local instance to generate a diverse sample.

5.3. Local Interpretation Models

This section evaluates the performance of the local interpretation models SS-LIME, Anchor, and the decision and counterfactual rules from SS-LORE. We set $M = 5000$ and use the following hyperparameters: for SS-LIME: $\sigma = 1.0$ (Ribeiro et al., 2016), for Anchor: $B = 3$, $\epsilon = 0.25$, $\delta = 0.05$ and $\tau = 0.75$ (Ribeiro et al., 2018), and for SS-LORE: $d = 5$ (Guidotti et al., 2018a).

5.3.1. Performance Measures

Table 5 reports the hit rate and fidelity measures for SS-LIME, Anchor, and the decision and counterfactual rules from SS-LORE.

We observe that Anchor, as previously mentioned, achieves a perfect hit rate for both deep learning models. However, both the Anchor method and the counterfactual rules from SS-LORE have a lower fidelity compared to the other interpretation models for both deep learning models, indicating that both Anchor and the counterfactuals from SS-LORE are not suited to mimic LCR-Rot-hop and LCR-Rot-hop++.

Furthermore, the counterfactual rules obtain relatively low measures for the hit rate compared to the other interpretation models. This is due to a lack of significant changes in the features of

Table 5: Performance measures for the interpretation models. For all models except SS-LIME, the 5% significance level, denoted by *, is given for the z-test assessing the null hypothesis of a performance measure of a method being equal to the respective performance measure of SS-LIME.

	LCR-Rot-hop		**LCR-Rot-hop++**	
	Hit rate	Fidelity	Hit rate	Fidelity
SS-LIME	96.6%	**95.6%**	94.0%	97.6%
Anchor	**100%***	88.0%*	**100%***	88.4%*
Decision rules	85.4%*	93.4%*	97.8%*	**98.5%***
Counterfactual rules	33.7%*	72.6%*	49.0%*	76.3%*

the original instance when constructing the counterfactual instance according to these rules. This results in the black box models not changing their predictions. Thus, the low hit rate indicates that the black box models have trouble changing their predictions according to counterfactual rules. However, the LCR-Rot-hop++ model still performs better for the counterfactual rules than the LCR-Rot-hop model, indicating that SS-LORE is more suited to be used for LCR-Rot-hop++.

We observe that SS-LIME obtains the highest fidelity out of the interpretation models for the LCR-Rot-hop model, while the decision rules from SS-LORE have the highest fidelity for the LCR-Rot-hop++ model. In addition, the difference with other models is statically significant at a 5% significance level. Combining this with the relatively high hit rates with values of 96.6% and 97.8% respectively, which are also significantly better than all models except Anchor, we conclude that in terms of global performance measures SS-LIME and SS-LORE are best suited to mimic LCR-Rot-hop and LCR-Rot-hop++, respectively.

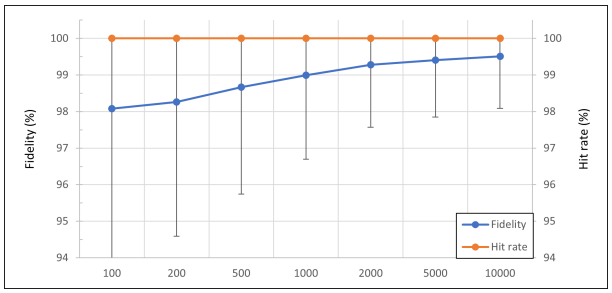
For SS-LIME and the decision and counterfactual rules from SS-LORE, we further investigate how the hit rate, fidelity, and training times per instance depend on the local neighborhood size. Specifically, we plot these performance measures against $M = 100$, 200, 500, 1K, 2K, 5K, and 10K in Figure 2 (note the different scales of the y-axes). Due to resource constraints, these measures were calculated over a subset of 50 out of the 295 instances. Therefore, they do not necessarily match the results in Table 5.

For SS-LIME, we see a positive relation between local neighborhood size and the performance of the model's fidelity. Not only does the fidelity steadily improve, but it becomes more consistent

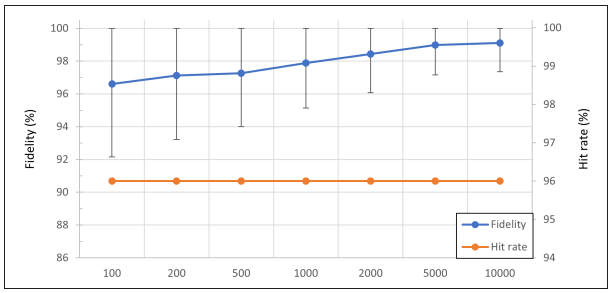
across the instances as well (reflected by the shorter error bars, which represent the standard deviation of the model's fidelity). The reverse is true for the decision rules from SS-LORE, suggesting that it benefits from a smaller neighborhood size. For the hit rate, we observe a steady performance across all neighborhood sizes for both SS-LIME and the decision rules from SS-LORE.

Moving onto the counterfactual rules, the trend is less obvious, although the relation between fidelity and M is more negative than positive, especially for $M > 2K$. The hit rate seems to develop more or less randomly. A straightforward explanation is that the counterfactual rules describe changes in the features of an instance that should lead to a different prediction. As a consequence, there are multiple outcomes from the interpretation model and the black box model that, although technically "correct", do not match (and hence do not result in a hit). For example, consider an instance where the correct label is 1 and the counterfactual rule leads to the interpretation model predicting label 0, while the black box model predicts label -1. Evidently, the hit rate of the counterfactual rules from SS-LORE depends less on M, but more on the rules themselves.

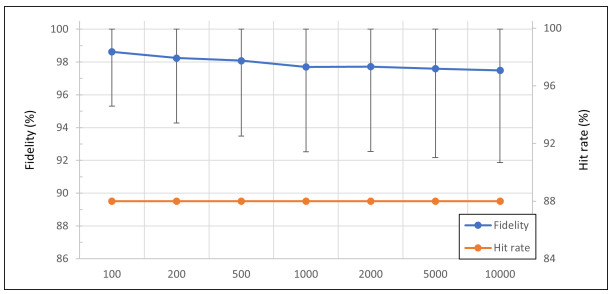
Last, it is worth noting that the training times per instance increase approximately linearly in M, which, for SS-LIME, leads to a trade-off between training time and performance. For SS-LORE, this trade-off is less relevant, as we observe that performance generally decreases with increasing M. Additionally, training times per instance increase at a noticeably greater rate for SS-LORE compared to SS-LIME.



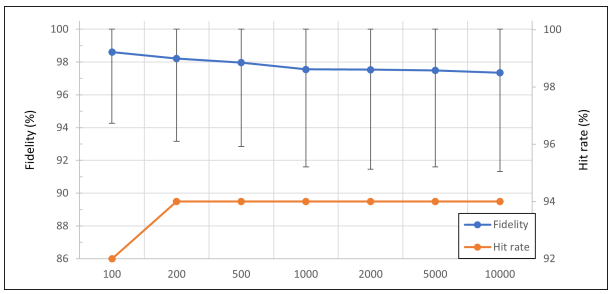
(a) Performance measures of SS-LIME for LCR-Rot-hop.



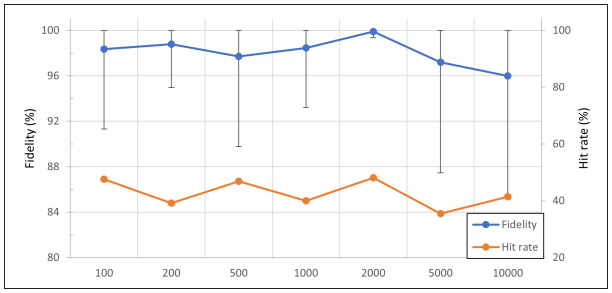
(b) Performance measures of SS-LIME for LCR-Rot-hop++.



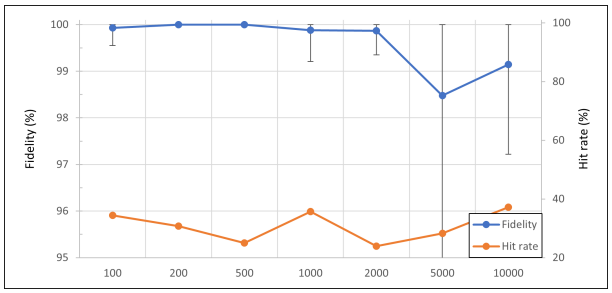
(c) Performance measures of decision rules from SS-LORE for LCR-Rot-hop.



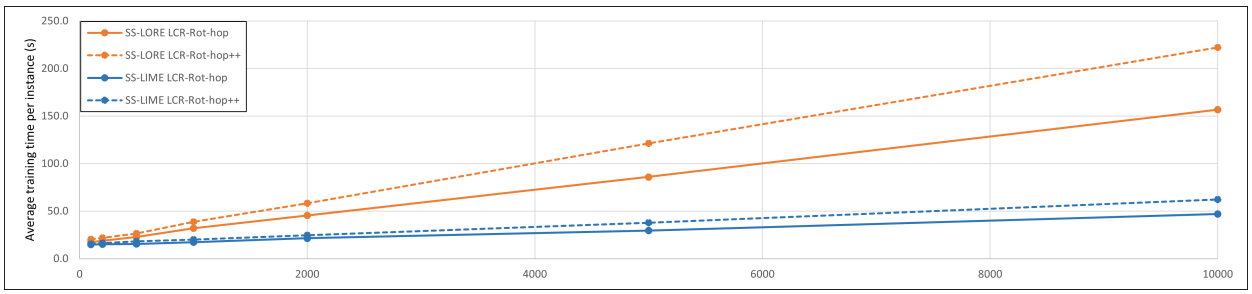
(d) Performance measures of decision rules from SS-LORE for LCR-Rot-hop++.



(e) Performance measures of counterfactual rules from SS-LORE for LCR-Rot-hop.



(f) Performance measures of counterfactual rules from SS-LORE for LCR-Rot-hop++.



(g) Average training time per instance (50 instances) of SS-LIME and SS-LORE for LCR-Rot-hop and LCR-Rot-hop++.

Figure 2: Performance measures fidelity (left y-axis with error bars denoting standard deviations) and hit rate (right y-axis) **(a-f)**, and average training time per instance **(g)**, for varying M, which controls the local neighborhood size (x-axis).

5.3.2. Comparison of Picking Algorithms

In this section, we compare the three picking algorithms to determine which algorithm is best suited to pick instances for our interpretation models. These picking algorithms are implemented using SS-LIME. We set $S = 4$ features to be considered for the explanation and each deep learning model picks $R = 5$ instances for each picking algorithm. Furthermore, the degree of reliability κ of our prior beliefs is 84.1% based on four test participants with a background in econometrics.

Table 6: Average number of features consistent with our prior beliefs for the picked instances.

Picking algorithm	**LCR-Rot-hop**	**LCR-Rot-hop++**
RP	1.15	0.75
SP	1.10	1.05
WSP	**1.35**	**1.55**

We compare five instances selected by each picking algorithm for each deep learning model. Results show that WSP picks instances that, on average, highlight the most opinionated features that are consistent with our prior beliefs. The results are displayed in Table 6. In Tables 7 and 8 we present the results of a single RP-picked instance and the first picked instances by SP and WSP for the LCR-Rot-hop and LCR-Rot-hop++, respectively. Based on our prior beliefs we decide which features in the instances are the most opinionated towards the target. The target is given in bold for the instances: RP-1, RP-2, SP-1, SP-2, WSP-1, and WSP-2.

Table 7: Highlighted features, prior beliefs, true label, predicted label, and the number of consistent features of the picked instances for the LCR-Rot-hop model according to SS-LIME.

		LCR-Rot-hop			
Instance	Highlighted features	Prior beliefs	True	Predicted	# consistent
RP-1	("thing", "the", **"nice"**, "location")	("positive", **"nice"**)	-1	1	1
SP-1	("every", "the", "perfect", **"fantastic"**)	(**"fantastic"**)	1	1	1
WSP-1	("creme", "interesting", **"delicious"**, **"very"**)	(**"very"**, "savory", **"delicious"**)	1	1	2

- RP-1: "The only positive thing about **Misoposto** is the nice location."

- SP-1: "The **food** is fantastic, and the waiting staff has been perfect every single time we've been there."

- WSP-1: "The appetizer was interesting, but the **creme brulee** was very savory and delicious."

Table 8: Highlighted features, prior beliefs, true label, predicted label, and the number of consistent features of the picked instances for the LCR-Rot-hop++ model according to SS-LIME.

		LCR-Rot-hop++			
Instance	Highlighted features	Prior beliefs	True	Predicted	# consistent
RP-2	("only", "is", "thing", "location")	("positive", "nice")	-1	1	0
SP-2	("system", "seat", **"hip"**, "sound")	(**"hip"**)	1	-1	1
WSP-2	(",", **"pricey"**, "place", **"n't"**)	(**"pricey"**, **"n't"**, "fancy")	-1	-1	2

- RP-2: "The only positive thing about **Misoposto** is the nice location."

- SP-2: "The **music** playing was very hip, 20-30 something pop, but the subwoofer to the sound system was located under my seat, which became annoying midway through dinner."

- WSP-2: "Food wise, its okay but a bit pricey for what you get considering the **restaurant** is n't a fancy place."

The tables show that the instances WSP-1 and WSP-2 contain the highest number of highlighted features consistent with our prior beliefs. We conclude that adding weights to the SP algorithm improves picking instances where opinionated features are present. For this reason, we continue our research with WSP as the picking algorithm for the comparison between the interpretation models, since we consider opinionated features the most important features for sentiment classification of textual data.

5.3.3. Comparison of Interpretation Models

In this section, we aim to determine the best interpretation model for the black box models by explaining the instances WSP-1 and WSP-2 for the LCR-Rot-hop and LCR-Rot-hop++ models, respectively. In Table 9, we present the generated rules from Anchor and SS-LORE for the instance WSP-1, where label 1 stands for positive, and label -1 for negative. Figure 3 shows the marginal effects generated by SS-LIME and the prediction differences for the rule-based approaches Anchor and SS-LORE.

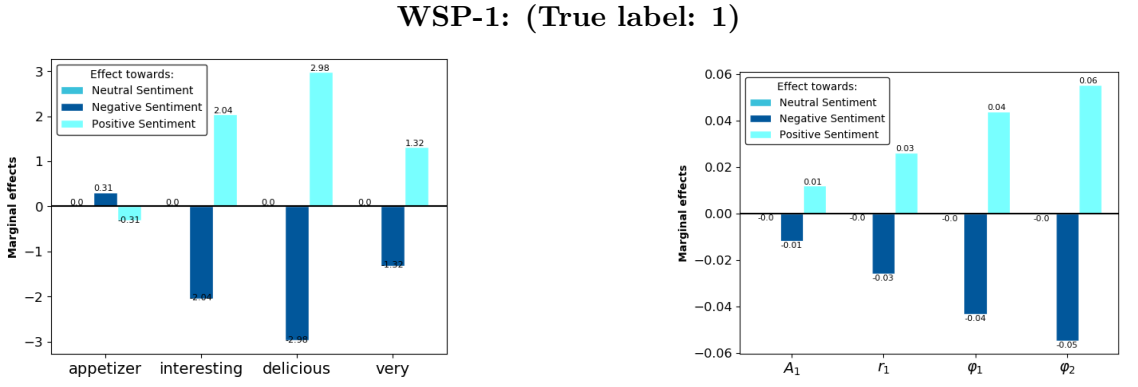
First, from Figure 3a, we observe that "*delicious*" provides the highest marginal effects towards the positive sentiment for the SS-LIME model. Furthermore, "*very*" also contributes positively to the positive sentiment. As mentioned in Table 7, the selection of these features is in line with our prior beliefs and thus provides an explanation that can be interpreted by users. However, we also notice that the remaining two features, "*appetizer*" and "*interesting*", are not opinionated (towards the target). This shows that SS-LIME does not perfectly capture all relevant features for the target.

When we consider the rules generated for WSP-1, which are displayed in Table 9, we notice that the generated rules from SS-LORE provide a better explanation than Anchor since they only contain opinionated features, such as *"delicious"*, *"savory"* and *"very"*, as opposed to some non-opinionated features, like *"was"* and *"but"*, which are included in the anchor. However, the associated labels of the counterfactual rules φ_1 and φ_2 are not consistent with the prediction differences as shown in Figure 3b, indicating that SS-LORE is not able to extract counterfactual rules in line with the predictions of the LCR-Rot-hop model. This decreases the interpretability of the SS-LORE rules. Since SS-LIME provides features that both reflect the sentiment towards the target of WSP-1 and are easily interpreted, we prefer SS-LIME over SS-LORE and thus conclude that the prediction of instance WSP-1 is best explained by the SS-LIME model.

Table 9: Generated rules for the instance WSP-1.

	Instance: WSP-1
Anchor:	A_1 = ("delicious", "very", "was", "but" $\rightarrow$ 1)
Decision rule:	r_1 = ("delicious" $\rightarrow$ 1)
Counterfactual rule(s):	$\Phi_1 = \{\varphi_1$ = ("delicious", "savory", $\neg$ "very" $\rightarrow$ -1),
	φ_2 = ("delicious", "very" $\rightarrow$ -1)$\}$

Note: $\neg$ indicates that the feature is not in the instance.



(a) Marginal effects for SS-LIME.

(b) Prediction differences for rule-based models.

Figure 3: "The appetizer was interesting, but the **creme brulee** was very savory and delicious."

For instance WSP-2, we observe from Figure 4a that *"n't"* has the highest marginal effect towards a negative sentiment. This is consistent with our prior beliefs and provides a valid expla-

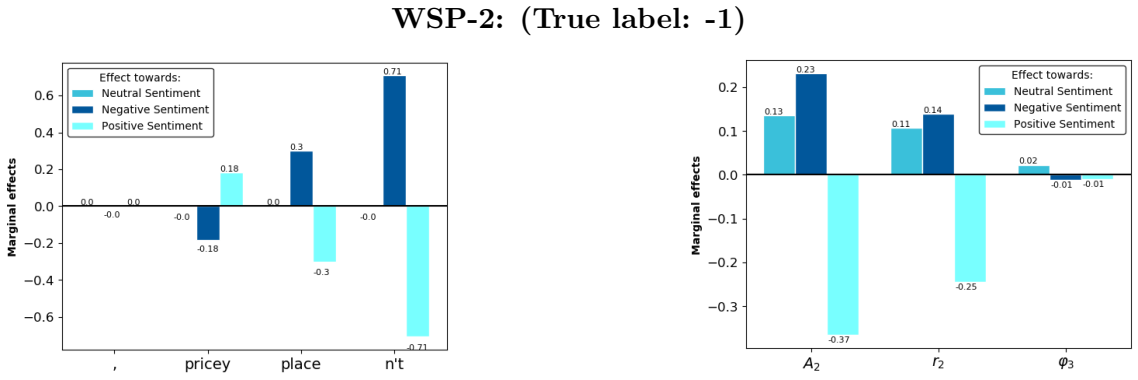
nation. However, this explanation is not sufficient for users as it does not explicitly capture what words are negatively opinionated towards the target *"restaurant"* of the instance.

When we evaluate the rule-based models in Table 10, we notice that Anchor provides a very lengthy rule A_2, which makes the rule not interpretable for users. The decision rule r_2 is able to capture the correct relevant features with respect to the target and does not contain any irrelevant features. Furthermore, the corresponding counterfactual rule φ_3 is consistent with the decision rule, since φ_3 indicates a positive sentiment when *"n't"* is left out of the instance. This results in a set of rules that clearly provides users with the features leading to the decision of a black box model and the features leading to a different decision. For these reasons, we conclude that WSP-2 is best explained by the SS-LORE model.

Table 10: Generated rules for instance WSP-2.

	Instance: WSP-2
Anchor:	A_2 = ("considering", "you", "wise", "what", "place", "ok", "its", "is", "but", "fancy" $\rightarrow$ -1)
Decision rule:	r_2 = ("fancy", $\neg$ "place", "n't" $\rightarrow$ -1)
Counterfactual rule(s):	$\Phi_2 = \{\varphi_3$ = ("fancy", "place", $\neg$ "n't" $\rightarrow$ 1)$\}$

Note: $\neg$ indicates that the feature is not in the instance.



(a) Marginal effects for SS-LIME.

(b) Prediction differences for rule-based models.

Figure 4: "Food wise, it's okay but a bit pricey for what you get considering the **restaurant** is n't a fancy place."

6. Conclusion

In this paper, we focused on comparing the interpretability of two state-of-the-art hybrid models for ABSA of restaurant reviews. To compare the interpretability, we used three post-hoc local interpretation models: SS-LIME, Anchor, and SS-LORE to interpret the deep learning models LCR-Rot-hop and LCR-Rot-hop++. These interpretation models are compared in terms of mimicking performance and user interpretability. Furthermore, we extended the interpretation models such that they could be used for ternary classification, in contrast to binary classification used in LIME (Ribeiro et al., 2016) or tabular data used in LORE (Guidotti et al., 2018a). Last, we also extended the SP algorithm by adding a weighted component. This resulted in the WSP algorithm, which picks more instances where opinionated features are present compared to the SP algorithm.

To answer our research question, we first provide an answer to our subquestion: *Which local interpretation models give the best interpretation for each of the state-of-the-art hybrid models for ABSA?* This can be answered by considering the mimicking performance and the user interpretability of the interpretation models for the picked instances by the WSP algorithm. From the results, we conclude that the LCR-Rot-hop and LCR-Rot-hop++ models can be best explained by SS-LIME and SS-LORE, respectively.

When we compare the interpretation ability of SS-LIME and SS-LORE for the deep learning models, we conclude that SS-LIME is less interpretable. The SS-LIME method is able to highlight the most important features, but it is not able to give the joint effect or changes of features leading to another prediction. In contrast, SS-LORE is more user-friendly than SS-LIME, since it provides us with explanations consisting of the reasons for a decision, and the changes of features leading to another decision. We answer our research question: *Which state-of-the-art hybrid model for ABSA can be best interpreted by a local interpretation model?* by concluding that LCR-Rot-hop++ can be better interpreted since it is best represented by the local interpretation model SS-LORE (using fidelity and hit rate). Regarding decision support, these results show that the LCR-Rot-hop++ model has both better user trustworthiness and accuracy than the LCR-Rot-hop model, and should thus be the preferred deep learning model for state-of-the-art hybrid models for ABSA in the restaurant domain.

However, one limitation of this research is that in the SS method, we sample local instances that are not necessarily generated with an equal proportion of negative, neutral, and positive labels.

Currently, the probability of picking a feature as a replacement is proportional to the similarity. This results in the local instances being fairly similar to the original instance. Instead, it would be interesting to consider a perturbation distribution that is able to generate local instances with an equal proportion of negative, neutral, and positive labels and letting the selection probability also be proportional to the distance between the features.

Future research could investigate the effect of more refined methods to interpret the results produced by a certain deep learning model. For example, one could use argumentation theory (Dragoni et al., 2018) to find the relations in an argumentation graph and to find correlations between the predictions of a deep learning model and the output of said argumentation graph, or use the Quantus package (Hedström et al., 2023) as a toolbox to further extend the quantitative evaluation of the proposed interpretability models. Second, it could be interesting to improve the SS method by generating balanced neighborhoods. This should further refine the local classifiers. Furthermore, our neighborhood sampling method can be extended to other methods, such as SHAP, to explore the performance of the proposed sampling method using different approaches. Last, SS-LIME and SS-LORE have been applied to two state-of-the-art ABSA deep learning models. Future work could delve into the application of SS-LIME and SS-LORE to other models, such as the method of Su et al. (2021) using a progressive self-supervised attention approach.

References

A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Information Fusion 58 (2020) 82–115.

A. R. Barron, T. M. Cover, Minimum complexity density estimation, IEEE Transactions on Information Theory 37 (4) (1991) 1034–1054.

J. Bastings, K. Filippova, The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?, in: Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP 2020), ACL, 149–155, 2020.

S. Bird, E. Klein, E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, O'Reilly Media, Inc., 2009.

C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

T. Botari, F. Hvilshøj, R. Izbicki, A. C. P. L. F. de Carvalho, MeLIME: Meaningful local explanation for machine learning models, arXiv preprint arXiv:2009.05818 (2020).

G. Brauwers, F. Frasincar, A survey on aspect-based sentiment classification, ACM Computing Surveys 55 (4) (2023) 65:1–65:37.

K. Bykov, A. Hedström, S. Nakajima, M. M. Höhne, NoiseGrad - Enhancing explanations by introducing stochasticity to model weights, in: Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2022), AAAI, 6132–6140, 2022.

E. Cambria, Affective Computing and Sentiment Analysis, IEEE Intelligent Systems 31 (2) (2016) 102–107.

E. Cambria, Q. Liu, S. Decherchi, F. Xing, K. Kwok, SenticNet 7: A Commonsense-based Neurosymbolic AI Framework for Explainable Sentiment Analysis, in: 13th Language Resources and Evaluation Conference (LREC 2022), ELRA, 3829–3839, 2022.

E. Cambria, A. Livingstone, A. Hussain, The Hourglass of Emotions, in: A. Esposito, A. M. Esposito, A. Vinciarelli, R. Hoffmann, V. C. Müller (Eds.), Cognitive Behavioural Systems, Springer, 144–157, 2011.

R. Dekker, D. Gielisse, C. Jaggan, S. Meijers, F. Frasincar, Knowledge Injection for Aspect-Based Sentiment Classification, in: 34th International Conference on Database and Expert Systems Applications, vol. 14147 of *LNCS*, Springer, 173–187, 2023.

J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technology (NAACL-HLT 2019), ACL, 4171–4186. ACL, 2019.

M. Dragoni, C. da Costa Pereira, A. G. B. Tettamanzi, S. Villata, Combining Argumentation and Aspect-Based Opinion Mining: The SMACk System, AI Communications 31 (1) (2018) 75–95.

M. Du, N. Liu, X. Hu, Techniques for interpretable machine learning, Communications of the ACM 63 (1) (2019) 68–77.

EU, Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), `http://data.europa.eu/eli/reg/2016/679/oj`, 2016.

J. L. Fleiss, J. Cohen, The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability, Educational and Psychological Measurement 33 (3) (1973) 613–619.

K. Geed, F. Frasincar, M. M. Trusca, Explaining a Deep Neural Model with Hierarchical Attention for Aspect-Based Sentiment Classification Using Diagnostic Classifiers, in: 22nd International Conference on Web Engineering (ICWE 2022), vol. 13362 of *LNCS*, Springer, 268–282, 2022.

S. Ghalebikesabi, L. Ter-Minassian, K. DiazOrdaz, C. C. Holmes, On locality of local explanation models, in: 34th Annual Conference on Neural Information Processing Systems (NIPS 2021), Curran Associates, Inc., 18395–18407, 2021.

J. S. Giboney, S. A. Brown, P. B. Lowry, J. F. Nunamaker Jr, User acceptance of knowledge-based system recommendations: Explanations, arguments, and fit, Decision Support Systems 72 (2015) 1–10.

R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems, arXiv preprint arXiv:1805.10820 (2018).

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A Survey of Methods for Explaining Black Box Models, ACM Computing Surveys 51 (5) (2018b) 93:1–93:42.

A. Hedström, L. Weber, D. Krakowczyk, D. Bareeva, F. Motzkus, W. Samek, S. Lapuschkin, M. M. M. Höhne, Quantus: An explainable AI toolkit for responsible evaluation of neural network explanations and beyond, Journal of Machine Learning Research 24 (34) (2023) 1–11.

S. Jain, B. C. Wallace, Attention is not explanation, in: 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), ACL, 3543–3556, 2019.

E. Kaufmann, S. Kalyanakrishnan, Information complexity in bandit subset selection, in: 26th Annual Conference on Learning Theory (COLT 2013), vol. 30 of *JMLR Workshop and Conference Proceedings*, JMLR.org, 228–251, 2013.

M. Kraus, S. Feuerriegel, Decision support from financial disclosures with deep neural networks and transfer learning, Decision Support Systems 104 (2017) 38–48.

B. Liang, H. Su, L. Gui, E. Cambria, R. Xu, Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks, Knowledge-Based Systems 235 (2022) 107643.

B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Cambridge University Press, 2015.

S. Luo, H. Ivison, S. C. Han, J. Poon, Local interpretations for explainable natural language processing: A survey, ACM Computing Surveys 56 (9) (2024) 232:1–232:36.

A. Madsen, S. Reddy, S. Chandar, Post-hoc interpretability for neural NLP: A survey, ACM Computing Surveys 55 (8) (2023) 155:1–155:42.

X. Mei, Y. Zhou, C. Zhu, M. Wu, M. Li, S. Pan, A disentangled linguistic graph model for explainable aspect-based sentiment analysis, Knowledge-Based Systems 260 (2023) 110150.

L. Meijer, F. Frasincar, M. M. Trusca, Explaining a neural attention model for aspect-based sentiment classification using diagnostic classification, in: 36th ACM/SIGAPP Symposium on Applied Computing (SAC 2021), ACM, 821–827, 2021.

G. A. Miller, WordNet: An Electronic Lexical Database, MIT Press, 1998.

C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, `https://christophm.github.io/interpretable-ml-book/`, 2019.

C. Panigutti, A. Perotti, A. Panisson, P. Bajardi, D. Pedreschi, FairLens: Auditing black-box clinical decision support systems, Information Processing & Management 58 (5) (2021) 102657.

E. Pastor, E. Baralis, Explaining black box models by means of local rules, in: 34th ACM/SIGAPP Symposium on Applied Computing (SAC 2019), ACM, 510–517, 2019.

J. Pennington, R. Socher, C. D. Manning, GloVe: Global vectors for word representation, in: 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), ACL, 1532–1543, 2014.

M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. María Jiménez-Zafra, G. Eryiğit, SemEval-2016 task 5: Aspect based sentiment analysis, in: 10th International Workshop on Semantic Evaluation (SemEval-2016), ACL, 19–30, 2016.

M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, SemEval-2015 Task 12: Aspect Based Sentiment Analysis, in: 9th International Workshop on Semantic Evaluation (SemEval 2015), ACL, 486–495, 2015.

J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1992.

M. R. R. Rana, S. U. Rehman, A. Nawaz, T. Ali, M. Ahmed, A Conceptual Model for Decision Support Systems Using Aspect Based Sentiment Analysis, Romanian Academy Series A-Mathematics Physics Technical Sciences Information Science 22 (4) (2021) 381–390.

M. T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier, in: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016), ACM, 1135–1144, 2016.

M. T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, in: Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018), AAAI Press, 1527–1535, 2018.

M. K. Saggi, S. Jain, A survey towards an integration of big data analytics to big insights for value-creation, Information Processing & Management 54 (5) (2018) 758–790.

M. Salehan, D. J. Kim, Predicting the performance of online consumer reviews: A sentiment mining approach to big data analytics, Decision Support Systems 81 (2016) 30–40.

K. Schouten, F. Frasincar, Survey on aspect-level sentiment analysis, IEEE Transactions on Knowledge and Data Engineering 28 (3) (2016) 813–830.

S. Srinivas, F. Fleuret, Rethinking the role of gradient-based attribution methods for model interpretability, in: 9th International Conference on Learning Representations (ICLR 2021), OpenReview.net, 2021.

J. Su, J. Tang, H. Jiang, Z. Lu, Y. Ge, L. Song, D. Xiong, L. Sun, J. Luo, Enhanced aspect-based sentiment analysis models with progressive self-supervised attention learning, Artificial Intelligence 296 (2021) 103477.

Y. Susanto, A. G. Livingstone, B. C. Ng, E. Cambria, The Hourglass Model Revisited, IEEE Intelligent Systems 35 (5) (2020) 96–102.

Z. Tan, Y. Tian, J. Li, GLIME: General, stable and local LIME explanation, in: 36th Annual Conference on Neural Information Processing Systems (NIPS 2023), Curran Associates, Inc., 36250–36277, 2023.

M. M. Truşcă, D. Wassenberg, F. Frasincar, R. Dekker, A Hybrid Approach for Aspect-Based Sentiment Analysis Using Deep Contextual Word Embeddings and Hierarchical Attention, in: 20th International Conference of Web Engineering (ICWE 2020), vol. 12128 of *LNCS*, Springer, 365–380, 2020.

O. Wallaart, F. Frasincar, 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), vol. 11503 of *LNCS*, Springer, 363–378, 2019.

K. Wang, W. Shen, Y. Yang, X. Quan, R. Wang, Relational Graph Attention Network for Aspect-based Sentiment Analysis, in: 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), ACL, 3229–3238, 2020.

F. Wenzel, K. Roth, B. S. Veeling, J. Swiatkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, S. Nowozin, How good is the Bayes posterior in deep neural networks really?, in: 37th International Conference on Machine Learning (ICML 2020), vol. 119 of *PMLR*, PMLR, 10248–10259, 2020.

S. Wiegreffe, Y. Pinter, Attention is not not explanation, in: 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), ACL, 11–20, 2019.

C. Zhang, Q. Li, D. Song, Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks, in: 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), ACL, 4567–4577, 2019.

A. Zhao, Y. Yu, Knowledge-enabled BERT for aspect-based sentiment analysis, Knowledge-Based Systems 227 (2021) 107220.

S. Zheng, R. Xia, Left-Center-Right Separated Neural Network for Aspect-based Sentiment Analysis with Rotatory Attention, arXiv preprint arXiv:1802.00892 (2018).

Z. Zhou, G. Hooker, F. Wang, S-LIME: Stabilized-LIME for model explanation, in: 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2021), ACM, 2429–2438, 2021.