SHAP-Based Explanation Methods: A Review for NLP Interpretability

Edoardo Mosca
TU Munich,
Department of Informatics,
Germany
edoardo.mosca@tum.de

Ferenc Szigeti
TU Munich,
Department of Informatics,
Germany
ferenc.szigeti@tum.de

Stella Tragianni
TU Munich,
Department of Informatics,
Germany
stella.tragianni@tum.de

Daniel Gallagher
University College Dublin,
Department of Informatics,
Ireland
daniel.gallagher1@ucdconnect.ie

Georg Groh
TU Munich,
Department of Informatics,
Germany
grohg@in.tum.de

Abstract

Model explanations are crucial for the transparent, safe, and trustworthy deployment of machine learning models. The *SHapley Additive exPlanations* (SHAP) framework is considered by many to be a gold standard for local explanations thanks to its solid theoretical background and general applicability. In the years following its publication, several variants appeared in the literature—presenting adaptations in the core assumptions and target applications. In this work, we review all relevant SHAP-based interpretability approaches available to date and provide instructive examples as well as recommendations regarding their applicability to NLP use cases.

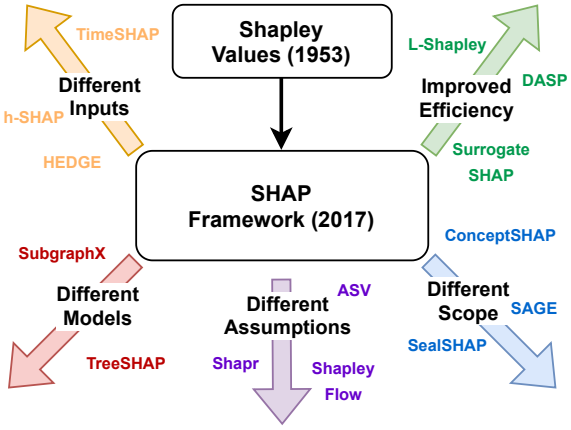

Figure 1: This work identifies five research directions pursued by Shapley- and SHAP-based approaches in XAI. Each direction, together with a few notable methods as examples, has been indicated by a different color.

1 Introduction

Several methods have been proposed to address the issue of opacity in modern machine learning models. Most notoriously, explanations are fundamental for *Deep Neural Networks* (DNNs) (Devlin et al., 2019; Madsen et al., 2021; Mosca et al., 2021) as these automatically learn millions of parameters and behave like black-boxes. Lundberg and Lee (2017) proposes *SHapley Additive exPlanations* (SHAP), a unified local-interpretability framework with a rigorous theoretical foundation on the game-theoretic concept of Shapley values (Shapley, 1953).

SHAP is nowadays considered a core contribution to the field of *eXplainable Artificial Intelligence* (XAI). Following its publication, a variety of explainability approaches based on SHAP's methodology has populated the literature and this trend continues to grow. Some present a new version of SHAP tailored to a certain type of input data—e.g. graphs (Yuan et al., 2021) and text (Chen et al., 2020)—or to specific models such as random forests (Lundberg et al., 2018). Others, instead, modify SHAP's underlying assumptions—e.g. features independence—to increase the original framework's flexibility for cases in which they are too strict or overly simplistic (Frye et al., 2019).

In this work, we **(1)** identify five broad research directions inspired by SHAP, **(2)** review available SHAP-based (or Shapley-value-based) approaches as members of such categories, and **(3)** investigate their applicability in the domain of *Natural Language Processing* (NLP).

Our work reviews 41 methods with a particular focus on their core assumptions, input require-

Proceedings of the 29th International Conference on Computational Linguistics, pages 4593–4603
October 12–17, 2022.

ments, explanation form, and available implementations. Furthermore, we provide NLP researchers with use-case-based recommendations and instructive examples.

2 Background

For the sake of clarity, we provide a gentle introduction to Shapley values and the methods for their estimation, most notably SHAP. All concepts will be explained informally, resorting to formalities when necessary.

2.1 Shapley Values

Shapley Values are a concept from game theory, originally developed as a measure to fairly distribute a reward among a set of players contributing to a certain outcome (Shapley, 1953). In the context of machine learning models, the players involved are the input features and the outcome is the model's decision, Shapley values attribute an importance score to each part of the input (Lundberg and Lee, 2017).

Given the set of input features $\mathbf{F} = \{1, 2, \ldots, p\}$, all features in a certain coalition $S \subseteq \mathbf{F}$ cooperate towards the outcome $val(S)$—with the default $val(\emptyset) = 0$. Shapley values redistribute the total outcome value $val(\mathbf{F})$ among all features based on their average marginal contribution across all possible coalitions S. More specifically, feature i's marginal contribution w.r.t. a coalition S:

$$\Delta_{val}(i, S) = val(S \cup \{i\}) - val(S)$$

is averaged across all $S \subseteq \mathbf{F} \setminus \{i\}$. Hence, the corresponding Shapley values $\phi_{val}(i)$ measures its contribution based on the formula:

$$\phi_{val}(i) = \sum_{S \subseteq \mathbf{F} \setminus \{i\}} \frac{|S|!(p - |S| - 1|)!}{p!} \Delta_{val}(i, S)$$

Here, the coefficient $\frac{|S|!(p-|S|-1|)!}{p!}$ is used as normalization term based on the number of choices for the subset S. This redistribution of the total outcome $val(\mathbf{F})$ respects the four properties of:

Efficiency: All features contributions add up to the total outcome, i.e. $\sum_{i \in \mathbf{F}} \phi_{val}(i) = val(\mathbf{F})$.

Symmetry: If $val(S \cup \{i\}) = val(S \cup \{j\})$ for all $S \subseteq \mathbf{F} \setminus \{i, j\}$, then $\phi_{val}(i) = \phi_{val}(j)$

Dummy: If $val(S \cup \{i\}) = val(S)$ for all $S \subseteq \mathbf{F}$, then $\phi_{val}(i) = 0$

Additivity: In the presence of a single game with two outcomes val_1 and val_2, then Shapley values are additive w.r.t. the combined outcome, i.e. $\phi_{val_1+val_2}(i) = \phi_{val_1}(i) + \phi_{val_2}(i)$

2.2 Shapley Values Approximation and SHAP

The idea of utilizing Shapley values to compute feature attribution scores precedes the SHAP framework (Lipovetsky and Conklin, 2001; Song et al., 2016). In this case, the outcome val of the game is the prediction of a machine learning model f and Shapley values $\phi_f(i)$ measure the influence that each feature i has based on its current value. The early literature also worked on approximation strategies, as the exponential number of coalitions renders the exact estimation of Shapley values unfeasible (Štrumbelj and Kononenko, 2014; Datta et al., 2016). The main idea from these works is to compute $\phi_f(i)$ only for a smaller selection of subsets $S \subseteq \mathbf{F}$ and to estimate the effect of removing a feature by integrating over training samples. This eliminates the need to retrain the model for each choice of S.

The work from Lundberg and Lee (2017) introduces a new perspective that unifies Shapley value estimation with popular explainability methods such as LIME (Ribeiro et al., 2016), LRP (Binder et al., 2016), and DeepLIFT (Shrikumar et al., 2017). Furthermore, they propose SHAP values as a unified measure of feature importance and prove them to be the unique solution respecting the criteria of *local accuracy*, *missingness*, and *consistency*. The authors contribute a library of methods to efficiently approximate SHAP values in a variety of settings:

KernelSHAP: Adaptation of LIME—hence model-agnostic—to approximate SHAP values. As it works for any model f, it cannot make any assumption on its structure and is thus the slowest within the framework.

LinearSHAP: Specific to linear models, uses the model's weight coefficients and optionally accounts for inter-feature correlations.

DeepSHAP: Adaptation of DeepLIFT—hence specific to neural networks–to approximate SHAP values. Considerably faster than its model-agnostic counterpart as it makes assumptions about the model's compositional nature.

While not initially presented in Lundberg and Lee (2017), the following algorithms were later

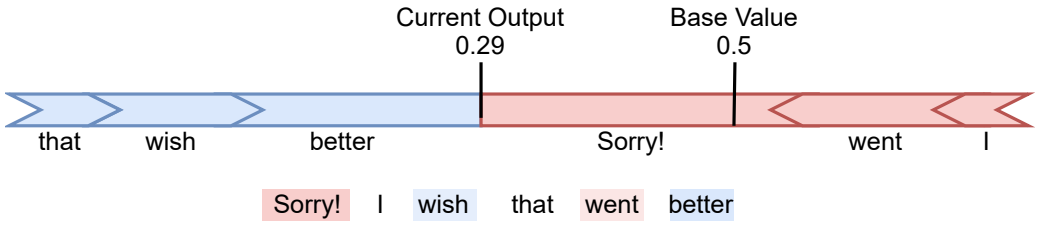

Figure 2: Example of explanation for sentiment analysis that can be generated with the SHAP library, e.g. with KernelSHAP. The base value indicates the model's average prediction. Each feature—i.e. word—contributes to the outcome, thus justifying the difference between the average and the current outcome.

added as part of the framework:

PartitionSHAP: Faster version of KernelSHAP that hierarchically clusters features. This hierarchy defines feature coalitions based on their interactions.

GradientSHAP: An extension of the *Integrated Gradients* (IG) method (Sundararajan et al., 2017)—again specific to neural networks—that aggregates gradients over the difference between the expected model output and the current output.

TreeSHAP: A fast method for computing exact SHAP values for both trees and ensembles (Lundberg et al., 2020a). In comparison to KernelSHAP, it also accounts for interactions among features.

Other minor approaches—PermutationSHAP, SamplingSHAP, ExactSHAP, and MimicSHAP—are also available in the official library[1]. To avoid confusion, we point out that the implementations have slightly different names: they use *"Explainer"* instead of *"SHAP"*. For instance, KernelSHAP and DeepSHAP are implemented with the names of *KernelExplainer* and *DeepExplainer* respectively. Figure 2 sketches an explanation generated with SHAP.

3 Search and Selection Criteria

As the popularity of SHAP increases, also the number of approaches based on it or directly on Shapley values has been on the rise. In fact, $\sim 3,200$ of the $\sim 6,900$ papers citing Lundberg and Lee (2017) are from 2021, an exponential increase when compared to previous years (1563, 567, and 118)[2].

Besides the papers already known to us, we manually screened all works citing SHAP with at least 15 citations[2]. This systematical search, based on the assumption that SHAP-based approaches should at least reference Lundberg and Lee (2017), helped us uncover several relevant contributions and mitigate the selection bias induced by our previous knowledge. The threshold of 15 citations was introduced to speed up our manual search and to filter out works that have not received the research community's attention. To account for temporal bias—i.e. that publications accumulate citations over time—we lowered the threshold to 10 for papers published in the most recent years (2021 and 2022)[2]. We only consider and review papers that contributed new SHAP-based approaches and exclude those—like (Wang, 2019) and (Antwarg et al., 2019)—utilizing SHAP (almost) off-the-shelf. Similarly, we exclude works such as Wang et al. (2020) and Huber et al. (2022) utilizing Shapley values for purposes not directly connected with explainability.

4 Existing Reviews

Previous reviews like Linardatos et al. (2021), Vilone and Longo (2020), and Madsen et al. (2021) present extensive overviews of explainability methods, but only briefly mention SHAP and a few of its derivates. Others—such as Covert et al. (2021), Sundararajan and Najmi (2020), and Kumar et al. (2020)—review some Shapley-based methods in detail (between 5 and 9) but do not construct a comprehensive review. Our work, in contrast, significantly extends this range and covers more than 40 approaches.

5 Review: SHAP-Based Approaches

Several works proposed methods based on SHAP, or more generally on Shapley values, following the contribution from Lundberg and Lee (2017). While the changes and variations introduced have been at times criticized for not being as rigorous as SHAP in following its core assumptions (Sundararajan

[1] https://github.com/slundberg/shap

[2] All queries are performed with Google Scholar. Accessed on 10.05.2022.

and Najmi, 2020), SHAP-based methods continue to increase in both quantity and popularity.

Our review categorizes SHAP-based approaches available to date based on *how they differ from* and *how they improve on* the original SHAP framework. We identify five broad categories in the existing literature, each one of them describing a different research direction pursued by its members:

(C1) Tailored to Different Input Data: This category contains approaches specialized on specific input data structures such as graphs (Wang et al., 2021), structured text (Chen et al., 2020), and images (Teneggi et al., 2021). In some cases, approaches are used complementary for applications dealing with multimodal inputs (Wich et al., 2021; Mosca et al., 2022b).

(C2) Explaining Different Models: Methods in this class are specifically designed to explain predictions from particular types of machine learning models such as random forests (Lundberg et al., 2018; Labreuche and Fossier, 2018) and neural networks (Ghorbani and Zou, 2021). Hence, these are model-specific.

(C3) Modifying Core Assumptions: SHAP treats features as independent. Newer methods offer the possibility to account for dependencies between features (Frye et al., 2019) and for causal structures behind their interactions (Heskes et al., 2020).

(C4) Producing Different Explanations Types: SHAP is a framework for local feature-attribution explanations, i.e. it attributes scores to input components based on their instance-level contributions. Methods in this category have a different scope and generate explanations that convey a different type of information. This can vary from global explanations (Covert et al., 2020) to counterfactual explanations (Singal et al., 2019) and concept explanations (Yeh et al., 2020).

(C5) Estimating Shapley Values More Efficiently: These approaches comprise alternative strategies for the approximation of Shapley values. Their focus is on leveraging prior knowledge about the data and model to improve the approximation *efficiency* and *accuracy* (Messalas et al., 2019; Chen et al., 2018).

Clearly, these categories are not designed to be exclusive. Therefore, an approach can fall in more than one if it differs from SHAP in multiple aspects. Table 1 provides an overview of all approaches with their main characteristics. As one can observe, the majority of approaches are identified as part of more categories, i.e. research directions.

5.1 Approaches Tailored to Different Inputs

SHAP does not make strong assumptions on the target model's input. While this suggests that it is suitable for all input types, its lack of specificity results in limitations when applied directly to different inputs than tabular data.

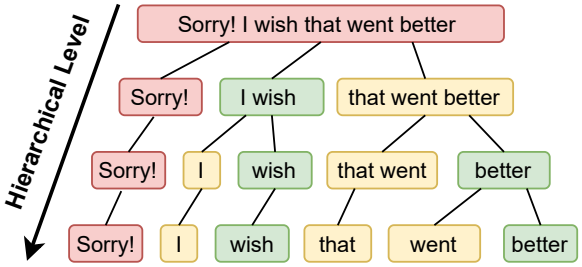

Figure 3: Example of hierarchical explanation that can be generated with HEDGE (Chen et al., 2020) for a sentiment analysis model. Each token is colored by contribution: negative (red), neutral (yellow), and positive (green). Going one level lower represents a token-breakdown step and thus more fine-grained Shapley values.

For text data, only measuring each individual feature's effect is an oversimplification, as words present strong interactions and their meaning and contribution heavily rely on the context. Thus, when it comes to text data, only considering single words as features is quite restrictive and relevance scores should be applied to multi-level tokens or even to entire sentences. *Hierarchical Explanation via Divisive GEneration* (HEDGE) (Chen et al., 2020) is an example of a SHAP-based method addressing this issue for (long) texts. Based on the weakest token interactions, it iteratively divides the text into shorter phrases and words in a top-down fashion. At each level, a relevance score is attributed to each token, resulting in a hierarchical explanation (Chen et al., 2020). PartitionSHAP, recently added to the official SHAP repository[3], follows a similar strategy by creating hierarchical features coalitions and measuring their interactions.

[3]https://github.com/slundberg/shap

Method	Categories	Description	NLP Applicability / Implementation
SHAP (Lundberg and Lee, 2017)		The original SHAP framework including the methods: KernelSHAP, LinearSHAP, DeepSHAP, etc.	*Ready Off-the-Shelf* Python
AVA (Bhatt et al., 2020)	**(C5)**	Combines the explanations of nearest neighbors to explain a given instance	*Adaptable* n.a.
ASV (Frye et al., 2019)	**(C1) (C3)**	Relaxes the symmetry axiom of Shapley values to incorporate causal structure into explanations	*Potentially Applicable* R
BShap (Sundararajan and Najmi, 2020)	**(C4) (C5)**	Baseline approach to facilitate comparison between different Shapley value based methods	*Adaptable* n.a.
C- and L-Shapley (Chen et al., 2018)	**(C3) (C5)**	Efficient feature attribution method that models data as a graph by considering only neighboring features	*Ready Off-the-Shelf* TensorFlow
CASV (Singal et al., 2019)	**(C1) (C2) (C3) (C4)**	Shapley value adaptation to account for counterfactuals by adhering to the Rubin Causal Model	*Not Relevant* n.a.
Causal Shapley (Heskes et al., 2020)	**(C1) (C3)**	Computing feature importance on data with (partial) causal ordering using Pearl's do-calculus	*Potentially Applicable* R
ConceptSHAP (Yeh et al., 2020)	**(C4)**	Unsupervised discover of concepts inherent to the data and model based on Shapley values	*Ready Off-the-Shelf* PyTorch
DASP (Ancona et al., 2019)	**(C3) (C5)**	Polynomial-time approximation of Shapley values in DNNs	*Adaptable* TensorFlow
Data Shapley (Ghorbani and Zou, 2019)	**(C4)**	Shapley-based importance attribution method for individual data instances in the training set	*Potentially Applicable* TensorFlow
DeepSHAP v2 (Chen et al., 2021)	**(C2) (C5)**	Computes efficiently SHAP values for DNNs with an extension to explain stacks of mixed model types	*Adaptable* n.a.
GrammarSHAP (Mosca et al., 2022a)	**(C1) (C3)**	Hierarchical explanations for text inputs based on the sentence grammatical structure	*Adaptable* n.a.
gSHAP (Tan et al., 2018)	**(C4)**	Generates intuitive Shapley-based global by aggregating local explanations	*Potentially Applicable* n.a.
h-SHAP (Teneggi et al., 2021)	**(C1) (C5)**	Hierarchical implementation of Shapley values for their efficient computation in image data	*Potentially Applicable* PyTorch
HEDGE (Chen et al., 2020)	**(C1) (C3)**	Hierarchical explanations based on feature interaction detection specifically for text data	*Ready Off-the-Shelf* PyTorch
Integrated Hessians (Janizek et al., 2021)	**(C5)**	Extension of Integrated Gradients to explain pairwise feature interactions in NNs	*Ready Off-the-Shelf* PyTorch
lossSHAP (Lundberg et al., 2020b)	**(C2) (C4)**	Obtain global explanations by aggregating local explanations with TreeSHAP	*Potentially Applicable* Python
MCDA Explainer (Labreuche and Fossier, 2018)	**(C1) (C2) (C3)**	Proposes the *influence index*, which is an extension of Shapley values for MCDA tree models	*Not Relevant* n.a.
Neuron Shapley (Ghorbani and Zou, 2021)	**(C2) (C4)**	Quantifies the contributions of single neurons to single predictions and overall model performance	*Adaptable* TensorFlow
R2 decomposition (Redell, 2019)	**(C5)**	Feature importance attribution based on Shapley value variance decomposition	*Potentially Applicable* R
Shapley Flow (Wang et al., 2021)	**(C1) (C3)**	Enables the addition of a causal graph encoding relationships among input features	*Potentially Applicable* Python
SAGE (Covert et al., 2020)	**(C4) (C5)**	Efficiently quantifies each feature's contribution to the model's performance for global explainability	*Potentially Applicable* Python
SealSHAP (Parvez and Chang, 2021)	**(C4)**	Shapley-based usefulness measure of individual data sources for transfer learning	*Ready Off-the-Shelf* TensorFlow
Shap-C (Ramon et al., 2019)	**(C4) (C5)**	Combination of computing counterfactuals and Shapley Values	*Potentially Applicable* Python
Shapley Residuals (Kumar et al., 2021)	**(C4)**	Captures information lost by KernelSHAP in Shapley Residuals, which characterize feature dependence	*Potentially Applicable* n.a.
Shapley Taylor index (Dhamdhere et al., 2020)	**(C3) (C5)**	Generalization of the Shapley value that attributes the model's prediction to interactions of subsets of features	*Potentially Applicable* n.a.
Shapr (Aas et al., 2021)	**(C3)**	Extends KernelSHAP to handle data with dependent features and produce more realistic explanations	*Potentially Applicable* R
SPVIM (Williamson and Feng, 2020)	**(C4) (C5)**	Global variable importance measure using an efficient regression-based Shapley value estimator	*Not Relevant* Python and R
SubgraphX (Yuan et al., 2021)	**(C1) (C2) (C5)**	Explain GNNs by identifying important subgraphs using Shapley values as importance measures	*Not Relevant* PyTorch
SurrogateSHAP (Messalas et al., 2019)	**(C5)**	An XGBoost tree model is trained as a surrogate model on the target model and TreeSHAP is applied to explain it	*Potentially Applicable* n.a.
TreeSHAP (Lundberg et al., 2018)	**(C2) (C5)**	Fast and exact method to estimate SHAP values for tree models and ensembles of trees	*Potentially Applicable* Python
TimeSHAP (Bento et al., 2021)	**(C1) (C2) (C4)**	Adapts KernelSHAP to sequential data and produces feature, event and cell-wise explanations	*Potentially Applicable* n.a.

Table 1: Overview of available Shapley- and SHAP-based methods. For each method we also indicate the categories it belongs to, its main idea and intuition, and its applicability to NLP together with the available implementations. See 6.1 for more details about our NLP-applicability assessment.

Figure 3 sketches an example of a hierarchical explanation for text data.

For models trained on graph data, especially graph DNNs, Yuan et al. (2021) proposed to explain predictions by using Shapley values as a measure of subgraph importance. The resulting method—named SubgraphX—also captures the interactions between different subgraphs.

On images, SHAP can face computational limitations as the number of features, i.e. pixels, can become extremely large. h-SHAP (Teneggi et al., 2021) efficiently retrieves exact Shapley values by hierarchically excluding irrelevant image areas from the computation. This is done following the observation that, if a certain area in the image is uninformative, so are its constituent sub-areas, which are therefore not worth exploring.

5.2 Approaches Explaining Different Models

Explanation methods making fewer assumptions on the target classifier benefit from better applicability as they can explain a wider range of models. However, this can hinder explanations in terms of accuracy, information granularity, and computational efficiency. As we have already seen in 2.2: KernelSHAP has the key advantage of being model-agnostic, but it is drastically more inefficient than its DNN-specific counterpart DeepSHAP (Lundberg and Lee, 2017).

An example of a highly-specialized explainability method is TreeSHAP, presented by Lundberg et al. (2018) as an extension of the SHAP framework. This approach, only applicable to decision trees or ensembles thereof, is a highly efficient algorithm for exact SHAP values retrieval. Not only the approach needs considerably less computational effort than the more general variants such as KernelSHAP, but it leverages the decision tree structure to compute SHAP interaction values and thus captures pairwise interactions between features.

Ghorbani and Zou (2021) proposes *Neuron Shapley*, a framework targeting DNN models which is able to quantify each individual neuron's contribution to single predictions and overall model performance. An example of the kind of explanation enabled by Neuron Shapley is visualized in figure 4. By analyzing interactions between neurons and picking those which exhibit the largest Shapley value, this method is particularly suitable for identifying neurons responsible for biases and vulnerabilities (Ghorbani and Zou, 2021).

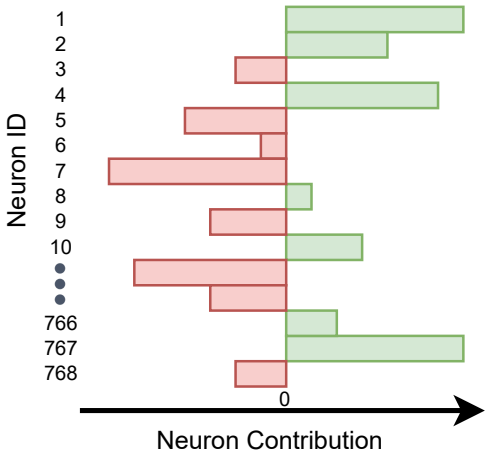

Figure 4: Sketch of a Neuron Shapley explanation for the 768 neurons of BERT output layer (Devlin et al., 2019). A Shapley value is assigned to each neuron depending depending on how they contribute towards the prediction (green) or against it (red).

5.3 Approaches Modifying Core Assumptions

Assumptions made by SHAP can be at times too restrictive or simplistic, which can prevent explanations from accessing and leveraging crucial information such as dependency relationships between input features. For instance, already the symmetry property of Shapley values treats features as independent. While this can be true in some cases, for instance when dealing with tabular data with uncorrelated variables, it is an oversimplification when it comes to texts, images, and more structured data.

Frye et al. (2019) introduces *Asymmetric Shapley Values* (ASV), which drops the symmetry assumption and enables the generation of model-agnostic explanations incorporating any causal dependency known to be present in the data. Similar approaches are:

- *Causal Shapley* (Heskes et al., 2020), additionally requiring a partial causal ordering of the features as input.
- *Shapley Flow* (Wang et al., 2021), which leverages a causal graph, encoding relationships among input features.
- *Shapr* (Aas et al., 2021), an extension of KernelSHAP relaxing the feature independence assumption.

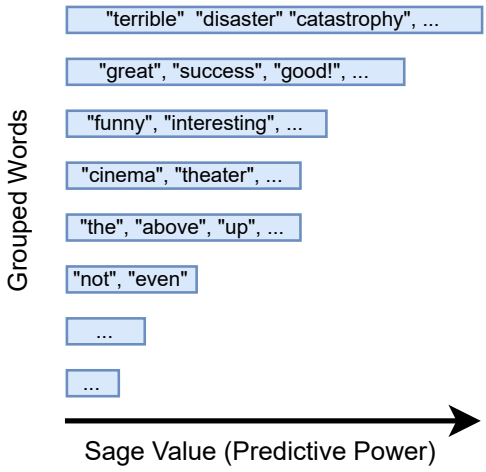

Figure 5: Example of SAGE explanation for a sentiment analysis model. Since the number of global features is as large as the vocabulary, words need to be grouped together (e.g. by similarity) to reduce the number of features to be explained.

5.4 Approaches Producing Different Explanation Types

The SHAP framework and many of its derivatives mainly focus on generating local explanations based on feature importance. However, the general applicability of Shapley values combined with its strong foundations also offers potential for different explainability settings. More recent works have explored the usage of Shapley values to build other types of explanations conveying different kinds of information about the model and the available data.

For instance, *Data Shapley* (Ghorbani and Zou, 2019) estimates the importance of each training sample for a given machine learning model. Similarly, SealSHAP (Parvez and Chang, 2021) attributes usefulness scores to data sources for transfer learning.

Covert et al. (2020) introduces *Shapley Additive Global importancE* (SAGE), an explainability method analogous to SHAP but with a core focus on global explainability. More in detail, SAGE is a model-agnostic method that quantifies the predictive power of each input feature for a given model while also accounting for their interactions. An instructive example for NLP is shown in figure 5.

Alongside local and global explainability, works like Yeh et al. (2020) adapt the notion of Shapley values for concept analysis (Sajjad et al., 2021). Given a set of concepts extracted from a model, the authors define the notion of *completeness* as a measure to indicate how sufficient such concepts are in explaining the model's predictive behavior. Furthermore, they propose ConceptSHAP, an unsupervised approach able to automatically retrieve a set of interpretable concepts without needing to know them in advance.

5.5 Approaches Proposed for Estimation Efficiency

While Shapley values convey useful information about the importance or contribution of a certain input component, their computation quickly becomes infeasible as coalitions grow exponentially w.r.t. input size. The SHAP framework already addresses this issue by providing more efficient estimation techniques. Nevertheless, later works continued to explore improvements to further decrease the computational effort necessary to produce meaningful explanations.

Chen et al. (2018) leverage features dependencies in image and text data to build two efficient algorithms, *L-Shapley* and *C-Shapley*, for Shapley values estimation. Their methods only consider a subset of the possible coalitions based on the data's underlying graph structure, which connects for instance adjacent words and pixels in texts and images respectively.

SurrogateSHAP (Messalas et al., 2019), instead, trains an XGBoost tree as a surrogate for the original model. The surrogate is then used to generate SHAP explanations, which considerably reduces the computational cost compared to directly applying SHAP to the original (more complex) model.

6 Relevance for NLP Research

Large and complex neural NLP models—such as BERT (Devlin et al., 2019) and GPT-3 (Brown et al., 2020)—are used extensively in research and industry. The trend is justified by the strong correlation between models' size and their performance (Madsen et al., 2021; Brown et al., 2020). Naturally, increasing model complexity causes a higher demand for NLP explainability. In this section, we match this demand to the reviewed SHAP-based methods and provide researchers with use-case-based recommendations.

6.1 Applicability of the Approaches

In table 1 (rightmost column), we also evaluate each SHAP-based explainability approach based on its applicability to neural NLP models. In this regard, our assessment considers *availability of*

implementations, *suitability for text data*, and *conceptual complexity* as relevant factors. We organize all reviewed approaches into four tiers:

- *Ready Off-the-Shelf*: The code is available and is ready to be used as-is.
- *Adaptable*: The code is available and there are straightforward steps for its adaptation to NLP use cases. Alternatively, no code is available but there are clear instructions for an ad-hoc implementation for the NLP domain.
- *Potentially Applicable*: Strong assumptions and substantial implementation work are required to apply the method to NLP.
- *Not Relevant*: The method is only applicable to other domains and it does not provide any apparent value for explaining NLP models.

6.2 Recommendations for NLP Use Cases

To build feature attribution explanations, HEDGE (Chen et al., 2020) is arguably the most suitable choice, as hierarchical explanations can contain more information than their non-hierarchical counterpart, e.g. generated with SHAP. The strength of HEDGE becomes even more apparent when dealing with long texts, where sentence structure is of major relevance for the model to be explained. *L-Shapley*, *C-Shapley* (Chen et al., 2018) and PartitionSHAP can also be considered where hierarchical explanations are not necessary and very computationally efficient methods are required instead.

For model debugging, Neuron Shapley is suitable to identify neurons that are responsible for unintended biases or that are particularly vulnerable to adversarial attacks (Ghorbani and Zou, 2021). Pruning these neurons can be an effective method of alleviating such model defects (Ghorbani and Zou, 2021). To gain a global understanding of what the model has learned in practice, SAGE (Covert et al., 2020) combined with word grouping provides a summary of the features—e.g. words—that are most relevant for the model's performance. In this case, pruning irrelevant features can be also tested to improve model accuracy. A similar summary can be provided by ConceptSHAP (Yeh et al., 2020), which can compile a comprehensive list of the concepts identified by the model in an unsupervised fashion. Furthermore, ConceptSHAP can be used to determine the amount of model variance covered by the whole set of identified concepts (Yeh et al., 2020).

If causal structures or dependencies present in the text are known and can be explicitly modeled, then methods such as ASV (Frye et al., 2019), Shapley Flow (Wang et al., 2021), and Causal Shapley (Heskes et al., 2020) can leverage such information. For use cases involving graphs as part of multi-modal inputs—e.g. modeling a social network (Wich et al., 2021)—any of the previous methods can be combined with SubGraphX (Yuan et al., 2021) to also produce explanations for the graph component of the input.

When it comes to *sequence-to-sequence* tasks such as question answering and machine translation, the usage of SHAP-based methods has not been explored in depth. With a few exceptions[4], available approaches seem particularly tailored only to classification settings. We believe this is a strong limitation and we encourage the reader to look for alternatives.

7 Criticisms

The usage of Shapley values for generating model explanations has also been criticized. For instance, Kumar et al. (2020) shows that using Shapley values for feature importance leads to mathematical inconsistencies which can only be mitigated by introducing further complexity like causality assumptions. Moreover, the authors argue that Shapley values do not represent an intuitive solution to the human-centric goals of model explanations and thus are only suitable in a limited range of settings.

Sundararajan and Najmi (2020), on the other hand, criticize some Shapley-value-based methods. In fact, while a strong case for utilizing Shapley values can be made thanks to their uniqueness result in satisfying certain properties (see 2.1), often methods employing them operate under different assumptions and hence the uniqueness results loses validity in their context.

Merrick and Taly (2020) argues that existing SHAP-based literature focuses on the axiomatic foundation of Shapley values and their efficient estimation but neglects the uncertainty of the explanations produced. The authors illustrate how small differences in the underlying game formulation can lead to sudden leaps in Shapley values and can attribute a positive contribution to features that do not play any role in the machine learning model.

[4] https://shap.readthedocs.io/en/latest/text_examples.html

8 Conclusion

SHAP is a core contribution to explainable artificial intelligence and one of the most popular frameworks for local interpretability. A considerable amount of recent works has proposed SHAP-based approaches, which we identify as part of five different yet overlapping research directions. In particular, the recent literature has worked towards **(C1)** *tailoring explanations to different input data*, **(C2)** *explaining specific models*, **(C3)** *improving the framework's flexibility via modifying core assumptions*, **(C4)** *producing different explanation types*, and **(C5)** *estimating Shapley values more efficiently*.

This work has reviewed a total of 41 approaches and has organized them based on the introduced categories. As expected, given the overlapping nature of the classification, the majority of existing methods fall into multiple categories and have therefore each made distinct contributions to the field. While most of them are not directly applicable to NLP settings, we identified a few that can be beneficial for current practitioners. Furthermore, we have compiled a list of recommendations for each NLP use case. We also observe a severe limitation of SHAP-based methods in terms of applicability to sequence-to-sequence NLP tasks.

We hope our work provides NLP/XAI practitioners and newcomers with a comprehensive overview of SHAP-based approaches, with references to stimulate further investigation and future advances in academic and industrial research.

Acknowledgments

This paper has been supported by the German *Federal Ministry of Education and Research* (BMBF, grant 01IS17049).

References

Kjersti Aas, Martin Jullum, and Anders Løland. 2021. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502.

Marco Ancona, Cengiz Oztireli, and Markus Gross. 2019. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pages 272–281. PMLR.

Liat Antwarg, Ronnie Mindlin Miller, Bracha Shapira, and Lior Rokach. 2019. Explaining anomalies detected by autoencoders using shap. *arXiv preprint arXiv:1903.02407*.

Joao Bento, Pedro Saleiro, Andre Cruz, Mario Figueiredo, and Pedro Bizarro. 2021. Timeshap: Explaining recurrent models through sequence perturbations. *KDD*.

Umang Bhatt, Adrian Weller, and José MF Moura. 2020. Evaluating and aggregating feature-based model explanations. *arXiv preprint arXiv:2005.00631*.

Alexander Binder, Sebastian Bach, Gregoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Layer-wise relevance propagation for deep neural network architectures. In *Information science and applications (ICISA) 2016*, pages 913–922. Springer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. 2020. Generating hierarchical explanations on text classification via feature interaction detection. *arXiv preprint arXiv:2004.02015*.

Hugh Chen, Scott Lundberg, and Su-In Lee. 2021. Explaining models by propagating shapley values of local components. In *Explainable AI in Healthcare and Medicine*, pages 261–270. Springer.

Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. 2018. L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*.

Ian Covert, Scott Lundberg, and Su-In Lee. 2021. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90.

Ian Covert, Scott M Lundberg, and Su-In Lee. 2020. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33:17212–17223.

Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Kedar Dhamdhere, Ashish Agarwal, and Mukund Sundararajan. 2020. The shapley taylor interaction index. *PMLR*.

Christopher Frye, Colin Rowat, and Ilya Feige. 2019. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *NeurIPS 2020*.

Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR.

Amirata Ghorbani and James Zou. 2021. Neuron shapley: Discovering the responsible neurons. *NeurIPS 2021*.

Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. 2020. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *NeurIPS 2020*.

Lukas Huber, Marc Alexander Kühn, Edoardo Mosca, and Georg Groh. 2022. Detecting word-level adversarial text attacks via SHapley additive exPlanations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 156–166, Dublin, Ireland. Association for Computational Linguistics.

Joseph Janizek, Pascal Sturmfels, and Su-In Lee. 2021. Explaining explanations: Axiomatic feature interactions for deep networks. *JMLR*.

I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. 2020. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pages 5491–5500. PMLR.

Indra Kumar, Carlos Scheidegger, Suresh Venkatasubramanian, and Sorelle Friedler. 2021. Shapley residuals: Quantifying the limits of the shapley value for explanations. *NeurIPS*.

Christophe Labreuche and Simon Fossier. 2018. Explaining multi-criteria decision aiding models with an extended shapley value. In *IJCAI*, pages 331–339.

Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2021. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18.

Stan Lipovetsky and Michael Conklin. 2001. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330.

Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020a. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839.

Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020b. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67.

Scott M Lundberg, Gabriel G Erion, and Su-In Lee. 2018. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *NeurIPS 2017*.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2021. Post-hoc interpretability for neural nlp: A survey. *arXiv preprint arXiv:2108.04840*.

Luke Merrick and Ankur Taly. 2020. The explanation game: Explaining machine learning models using shapley values. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 17–38. Springer.

Andreas Messalas, Yiannis Kanellopoulos, and Christos Makris. 2019. Model-agnostic interpretability with shapley values. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–7. IEEE.

Edoardo Mosca, Defne Demirtürk, Luca Mülln, Fabio Raffagnato, and Georg Groh. 2022a. GrammarSHAP: An efficient model-agnostic and structure-aware NLP explainer. In *Proceedings of the First Workshop on Learning with Natural Language Supervision*, pages 10–16, Dublin, Ireland. Association for Computational Linguistics.

Edoardo Mosca, Katharina Harmann, Tobias Eder, and Georg Groh. 2022b. Explaining neural NLP models for the joint analysis of open-and-closed-ended survey answers. In *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*, pages 49–63, Seattle, U.S.A. Association for Computational Linguistics.

Edoardo Mosca, Maximilian Wich, and Georg Groh. 2021. Understanding and interpreting the impact of user context in hate speech detection. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 91–102.

Md Rizwan Parvez and Kai-Wei Chang. 2021. Evaluating the values of sources in transfer learning. *NAACL 2021*.

Yanou Ramon, David Martens, Foster Provost, and Theodoros Evgeniou. 2019. Counterfactual explanation algorithms for behavioral and textual data. *arXiv preprint arXiv:1912.01819*.

Nickalus Redell. 2019. Shapley decomposition of r-squared in machine learning models. *arXiv preprint arXiv:1908.09718*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Hassan Sajjad, Narine Kokhlikyan, Fahim Dalvi, and Nadir Durrani. 2021. Fine-grained interpretation and causation analysis in deep nlp models. *arXiv preprint arXiv:2105.08039*.

Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games 2.28*, page 307–317.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Raghav Singal, Omar Besbes, Antoine Desir, Vineet Goyal, and Garud Iyengar. 2019. Shapley meets uniform: An axiomatic framework for attribution in online advertising. In *The World Wide Web Conference*, pages 1713–1723.

Eunhye Song, Barry L Nelson, and Jeremy Staum. 2016. Shapley effects for global sensitivity analysis: Theory and computation. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1060–1083.

Erik Štrumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665.

Mukund Sundararajan and Amir Najmi. 2020. The many shapley values for model explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Sarah Tan, Giles Hooker, Paul Koch, Albert Gordo, and Rich Caruana. 2018. Considerations when learning additive explanations for black-box models. *arXiv preprint arXiv:1801.08640 3*.

Jacopo Teneggi, Alexandre Luster, and Jeremias Sulam. 2021. Fast hierarchical games for image explanations. *arXiv preprint arXiv:2104.06164*.

Giulia Vilone and Luca Longo. 2020. Explainable artificial intelligence: a systematic review. *arXiv preprint arXiv:2006.00093*.

Guan Wang. 2019. Interpret federated learning with shapley values. *arXiv preprint arXiv:1905.04519*.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2020. Shapley q-value: A local reward approach to solve global reward games. *AAAI*, 34:7285–7292.

Jiaxuan Wang, Jenna Wiens, and Scott Lundberg. 2021. Shapley flow: A graph-based approach to interpreting model predictions. *AISTATS 2021*.

Maximilian Wich, Edoardo Mosca, Adrian Gorniak, Johannes Hingerl, and Georg Groh. 2021. Explainable abusive language classification leveraging user and network data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 481–496. Springer.

Brian Williamson and Jean Feng. 2020. Efficient nonparametric statistical inference on population feature importance using shapley values. In *International Conference on Machine Learning*, pages 10282–10291. PMLR.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. 2020. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On explainability of graph neural networks via subgraph explorations. *arXiv preprint arXiv:2102.05152*.