SHAP Explanations for Multimodal Text-Tabular Models

James Burton

james.burton@durham.ac.uk

Durham University

Noura Al Moubayed

Durham University

Article

Keywords:

Posted Date: October 27th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3405528/v1

Additional Declarations: No competing interests reported.

SHAP Explanations for Multimodal Text-Tabular Models

James Burton[1,*] **and Noura Al Moubayed**[1]

[1]Durham University, Department of Computer Science, Durham, DH1 3LE, UK
[*]james.burton@durham.ac.uk

ABSTRACT

Research on model transparency has remained relatively limited within the growing field of multimodal machine learning, particularly with text-tabular datasets. To address this research gap, we present a novel multimodal masking framework that extends SHapley Additive exPlanations (SHAP) to text-tabular datasets. This framework, which we make publicly available, enables the generation of SHAP explanations for any text-tabular dataset using any combination method. By masking features according to their modality, our framework ensures that features are treated consistently across unimodal and multimodal settings. Furthermore, by deferring the model input formation until after the masking call, we make the framework agnostic to how the input is formatted, avoiding the issues that arise when pre-forming the data into text and applying the existing text masker. In an extensive study, we examine the impact that combination strategies and language models have on SHAP explanations. Notably, the choice of combination method considerably influences the features identified as most important by the model. Furthermore, our findings reveal that methods converting all input to text tend to assign greater relative importance to text features over tabular features.

Introduction

In machine learning, interpretable models have become increasingly crucial, especially in domains where decision-making processes must be transparent and understandable, such as healthcare, education, and finance[1,2]. The proposed EU Artificial Intelligence Act[3] would provide consumers with the right to an explanation, placing further demand on explainability. While multimodal tasks have been gaining increasingly prominent over the previous decade, the primary focus often lies in the text-image domain and less on text-tabular data[4]. Shi et al.[5] explore specifically for text and tabular data, using methods such as a simple *Weighted-Ensemble*, where two models are each trained on a single modality are their predictions combined linearly, or where the input is fit to a string template and a language model is trained in an *All-Text* approach. Prompting methods are assessed in Liu et al.[6], whereas Gu et al.[7] provides a unified framework for a selection of combination methods.

Despite the increasing popularity of multimodal tasks, there remains a gap in the research for producing explanations for these models. While there has been some work into text-image explainability[8,9], text-tabular remains unexplored. A popular tool for explaining unimodal models is SHapley Additive exPlanations (SHAP)[10], a game theory-based approach that relies on simulating coalitions of present and absent features. When multimodal data is fit to a single modality - such as in *All-Text* - before the masking phase, it is possible to elicit an explanation. However, masking tabular features as if they were text can lead to erroneous token groupings and importance values assigned to non-feature tokens. To address this, we propose a novel multimodal masker, a complementary addition to the SHAP library. By fusing the previously separated text and tabular maskers and by deferring input formation until after the masking stage, we make it possible to generate SHAP explanations for any text-tabular model while also avoiding the pitfalls of the unimodal masker and the *All-Text* method. With our approach, text and tabular features are treated consistently, no matter how they are combined. This framework, which we make publicly available, facilitates for the first time the generation of SHAP values for any text-tabular dataset and for any method of combining the two modalities. Moreover, we propose a series of experiments to compare the SHAP explanations of various combination methods on text-tabular datasets. By training four different text models, each on nine datasets and further with five combination methods, we intend not only to showcase the utility of our masker but also to gain insight as to which features drive performance, whether those features differ across experiments, and how reliance on each modality changes.

Related Work

SHAP values

At its core, SHAP leverages the concept of Shapley values[11] to quantify the individual contribution of each feature to a model's prediction. By simulating coalitions of present and absent features and observing the corresponding changes in output, SHAP

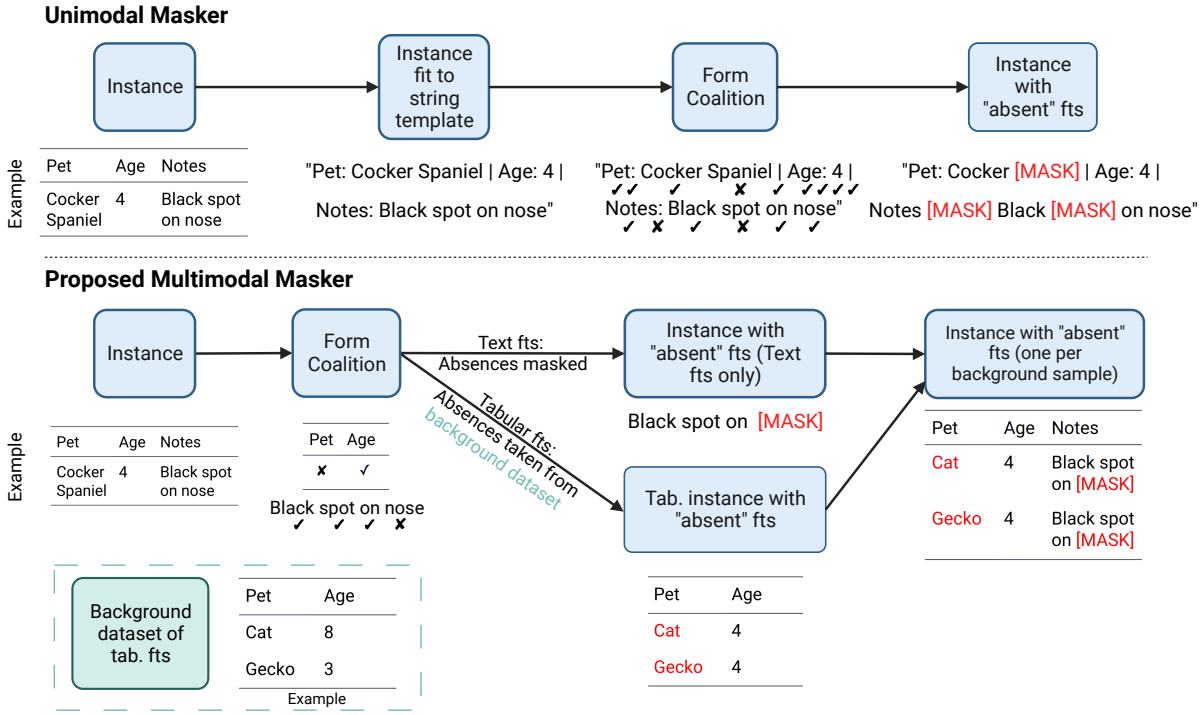

Figure 1. Comparison of masking processes with a multimodal text-tabular input, contrasting the current SHAP library's unimodal masker with the proposed multimodal masker. Using the unimodal masker in this scenario is only possible when we preform the input into a string. **Top:** Each token in the string input can be either absent (swapped for [MASK] token) or present. Note that tabular features are divided into tokens (e.g., "Cocker Spaniel"), and string template tokens, which should be omnipresent, can be absent(e.g., ":"). **Bottom:** Coalitions are formed, keeping tabular features as they are, while text features are split into tokens. Absent tabular features are sampled from a background dataset, while text tokens are replaced with [MASK] tokens. Once masked, features are reformed into an instance set, ready to be formatted for any multimodal model.

derives the marginal contributions of each feature to the final prediction[12]. In practice, SHAP requires three components to elicit an explanation: a model, an explainer, and a masker. The explainer's role is to dictate which coalitions are to be formed, using the masker to realize these binary coalitions as model-compatible inputs. These inputs are subsequently used by the model to generate a prediction. The choice of explainer and masker, both classes of the SHAP library, depends on the format of the data. For tabular data, the masker uses a background dataset to sample missing features, replacing missing features with a random sample and then integrating over the marginal distribution. For text data, an absent word piece is replaced with a mask token.

When dealing with text inputs or tabular data with many features, calculating the result of every possible coalition can be computationally intensive, especially for long text inputs that may consist of hundreds of tokens. The preferred solution is to use the partition explainer, which organizes features into a hierarchy and recursively calculates Shapley values. The resulting values are known as Owen values in game theory. For text, the partition hierarchy is created using a scoring system that clusters neighboring tokens together, favoring tokens belonging to the same word or not separated by punctuation. When used for tabular features, features are grouped based on their correlation.

Limitations of Using Unimodal Masker for Multimodal Explanations

Generating SHAP explanations for multimodal inputs poses a challenge within the existing framework as it is inherently designed for unimodal data representations. Consequently, the generation of SHAP explanations is only possible in the *All-Text* scenario where we are training a text model and have already formed our data into a single modality (text). With any other approach, it is not possible to elicit a SHAP explanation. In addition to this limitation, we illustrate the problems that arise when using the unimodal masker for the *All-Text* method. *All-Text* requires molding an input into a string template, following a format such as *Column name*: *Column value*, delineated by 'l'. However, when we form this input prior to the masking call, the masker is unable to distinguish between tabular feature, text feature, or string template. Figure 1 (top) illustrates the process of using the unimodal masker and *All-Text*. As shown, each word piece of the tokenized input is now open to being selected as

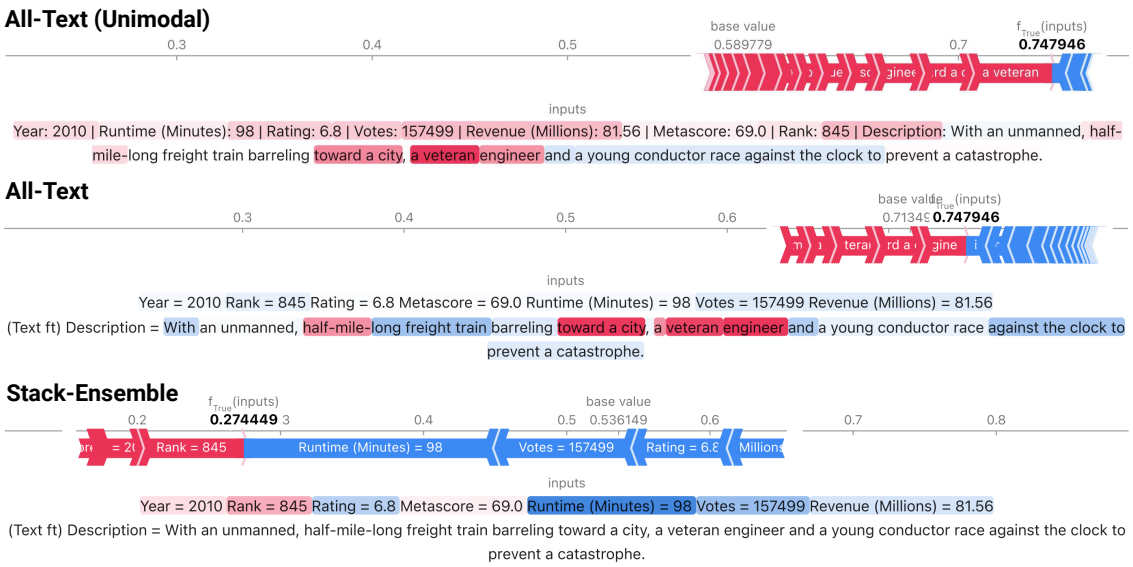

Figure 2. Three SHAP explanations of the same instance, all using text model= DistilBERT, dataset = *imdb*. Red indicates features that support the prediction, blue indicates those contributing in the opposite direction. Neighboring work tokens with the same color indicate membership of the same group for the given grouping threshold. **Top:** Using the original SHAP library's unimodal masker, combination method=*All-Text*. **Middle:** Using the proposed multimodal masker and combination method=*All-Text*. **Bottom:** Using the proposed multimodal masker and combination method=*Stack-Ensemble*.

present or absent in a coalition. However, as the masker is treating the input as if it were a passage of text, tabular features are not sampled from a background dataset as is precedent. Instead, as demonstrated in the top right of the figure, tabular features can be split into multiple tokens and masked separately. Furthermore, the string template, which is present and immutable for all instances and thereby necessarily has no impact between one instance and another, has no way of being recognized as such in the unimodal framework. Instead, as shown at the top of Figure 1 (top), template tokens will be treated as input features.

These issues are exacerbated further when we use the partition explainer, specifically during the formation of the partition hierarchy. Neighboring tokens are grouped but do not take into account whether neighbors are members of the same or differing features, leading to incorrect grouping, separation of tokens, and unrealistic and misleading explanations. In Figure 2, we see each one of the numeric, tabular features being split up to form erroneous groups. We show a portion of the example below, with one group highlighted in red.

> Year: 2010 | Runtime (Minutes): **98 | Rating: 6.**8 | Votes 157499 ...

When this group is absent, we are left with a misleading and unrealistic 8 minutes as a movie runtime.

> Year: 2010 | Runtime (Minutes): 8 | Votes 157499 ...

Methodology

In this study, we introduce a novel multimodal masker that incorporates both text and tabular features without needing to first convert to a single modality in order to extend SHAP's capabilities for multimodal data. Our objective is to ensure that text and tabular features are treated consistently, no matter whether it is in a unimodal or multimodal scenario or which combination method is chosen. We condense our changes into a single masker class and a simple model wrapper such that it can be easily integrated into the existing SHAP framework. Finally, we also make adjustments to the SHAP plotter, as illustrated in Figure 2 (middle, bottom). For clarity, we plot tabular column names and values together and add a label to indicate which features are text. During experimentation, we generate SHAP values for each combination of three independent variables: combination method (CM), text model (TM), and dataset (DS). We analyze the resulting explanations; notably, we compare the explanations gathered by our novel multimodal masker versus a unimodal one using the *All-Text* combination method.

Figure 1 (bottom) demonstrates the process of an explainer forming a coalition and then using the multimodal masker to form the model-compatible input. The key to the process is deferring the model input formation to after the masking call. In doing so, our masker takes the unformatted features as input, separating out text and tabular features. In this process, text and tabular features are separated such that absent tokens text tokens are substituted for a mask token, and absent tabular features

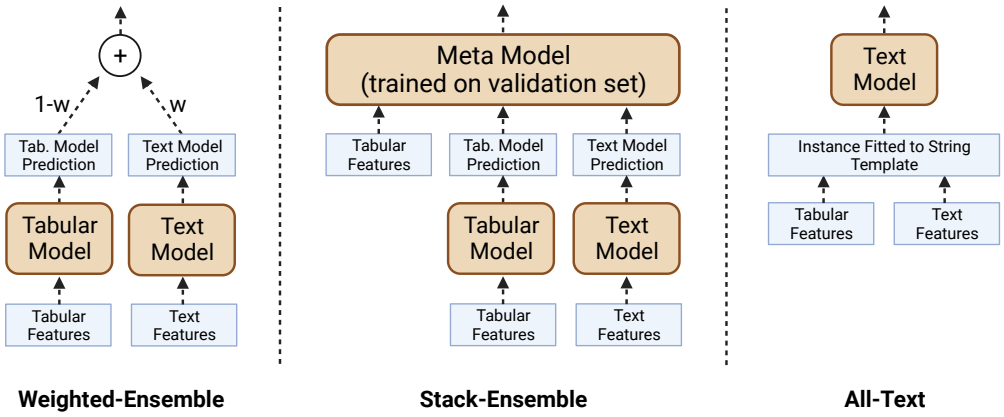

Figure 3. Combination methods used in this study. **Left:** A *Weighted-Ensemble*. **Middle:** A *Stack-Ensemble*. **Right:** *All-Text*.

are sampled from a background dataset, as would be the case with unimodal data. This results in a masked instance set with present and absent features realized (one for each background dataset sample), as seen on the far right of the diagram. This set is then passed to the model prediction function, and the average prediction is used by the explainer to calculate the SHAP values. We wrap the model prediction with a simple function to convert the masked instance set to a model-compatible input. For example, we can divert tabular and text features to a tabular and text model, respectively, in a *Weighted-Ensemble*, or format each instance to a string template for the *All-Text* approach. In maintaining the separation, we make this process completely agnostic of the combination method. To ensure compatibility with the partition explainer, we add functionality to create a joint partition hierarchy for a multimodal input. Following precedent, tabular features use feature correlations, while neighboring are grouped using the existing SHAP scoring method. The hybrid hierarchy is created by merging the two hierarchies at the highest level, thereby preventing the grouping of tabular features and word tokens.

Combination Methods

In the experimentation part of our study, we utilize five combination methods to merge text and tabular features, as illustrated in Figure 3: *Weighted-Ensemble* (using three different weights), *Stack-Ensemble*, and *All-Text*. These strategies are explained in this section.

Weighted-Ensemble: Tabular and text models are trained separately, and their predictions are combined with a weighted sum. We use w as the weight of the text model prediction and $(1-w)$ as the weight of the tabular model prediction. In this study, the tabular model is a LightGBM[13], and we experiment with three values of w: 0.25, 0.50, and 0.75.

Stack-Ensemble: Similar to the *Weighted-Ensemble*, text and tabular models are trained independently on their respective features. However, in the *Stack-Ensemble*, a meta-model is introduced. This model is trained on the validation set using the predictions from both the tabular and text models, along with the original tabular inputs. The goal is to learn, based on the values of the tabular features, which predictions from either the tabular or text models should be given more emphasis. This approach allows for a more nuanced and adaptive combination of the predictions, potentially capturing interdependencies between the two modalities. LightGBMs are used as the tabular and meta models in our experiments. This strategy often performed best for Shi et al.[5], although, in this study, we do not undertake k-fold cross-validation used to train the meta-model in order to reduce computation.

All-Text: In this method, all inputs, including tabular data, are treated as text, enabling us to leverage the capabilities of large language models. All features are fit to a string template and used to fine-tune a text model. In our experiments, we use a template in the form of *Column name*: *Column value*, delineated by 'l'. An example is shown in Figure 2 (top).

Text Models

The pre-trained language models that we finetune are BERT[14], DeBERTa[15], DistilBERT[16], and DistilRoBERTa[16]. In order to isolate the differences to the independent variables in question, for each TM-DS, we only train two text models. One is trained for *All-Text*, which uses all features, and one, which only uses the tabular features, for the *Weighted-Ensembles* and *Stack-Ensemble*.

Datasets

The datasets used are from Shi et al.[5], consisting of five multiclass and four binary classification tasks. We prepare the datasets in two different ways. The first, for the *All-Text* method, simply involves converting all values to strings. The second involves

Dataset ID	Train	Test	#Tab. Fts	#Text Fts	Task	Metric	Prediction Target
prod	5,091	1,273	1	1	multiclass	accuracy	sentiment associated with product review
salary	15,841	3,961	1	5	multiclass	accuracy	salary range in data scientist job listings
airbnb	18,316	4,579	35	5	multiclass	accuracy	price label of Airbnb listing
channel	20,284	5,071	16	1	multiclass	accuracy	news category to which article belongs
wine	84,123	21,031	2	3	multiclass	accuracy	which variety of wine (type of grape)
imdb	800	200	7	1	binary	roc-auc	whether film is a drama
fake	12,725	3,182	2	3	binary	roc-auc	whether job postings are fake
kick	86,502	21,626	6	3	binary	roc-auc	whether Kickstarter project will achieve funding goal
jigsaw	100,000	25,000	29	1	binary	roc-auc	whether social media comments are toxic

Table 1. Dataset information. This represents the dataset characteristics post-processing.

preparing the text and tabular features separately. Text features are converted to strings, numeric tabular features are left as they are, and categorical tabular features are ordinally encoded. For two of the datasets (*airbnb* and *imdb*), some of the features needed to be removed so as not to exceed the token limit of the text models. Preprocessing details and all final datasets are made available on our GitHub and Hugging Face repositories, respectively. The original test sets are used; the original train sets are divided into train and validation sets using an 85:15 split. After training the models, we find similar results to Shi et al.[5] and report the results in Supplementary Table S1. However, in some cases, our trained *Stack-Ensemble* models overfit and perform poorly on the test data. In order for our analysis of explanations to be valid, out of 216 experiments, we exclude 17 models that perform notably worse than others on the same dataset. For *channel*, we omit all *All-Text* and *All-Text (Unimodal)* results. For *salary* and *wine*, we omit all *Stack-Ensemble* results, and for *prod*, we omit *Stack-Ensemble* for DistilBERT. More information on the datasets can be found in Table 1.

Generating SHAP Values

For each dataset, we randomly select 100 instances from the test set to be explained. For experiments using the multimodal masker, a background dataset of size 100 (the default size for unimodal tabular SHAP) is randomly selected from the training set. Each TM-CM experiment on a dataset will explain the same 100 instances and use the same background dataset. We generate SHAP values for the selected instances for each TM-CM-DS combination using our multimodal masker. We also compute SHAP values for the same 100 instances for each TM-DS combination using the unimodal text masker and the *All-Text* combination method, we refer to these as *All-Text (Unimodal)*.

Results

We utilize a process similar to the SHAP package's summary plot function to account for the varying label counts and token quantities across instances. Formulaically, for each instance, there are T tokens, each belonging to one of F features. Each token has associated SHAP values for L labels. First, we sum the SHAP values for each token, $t \in T$, belonging to a feature, $f \in F$. We then take the absolute value and sum across each label, $l \in L$. Condensing results in this manner yields a single SHAP value for each feature in each instance, indicating how important the feature was to the model. We refer to this as feature importance or FI.

$$\mathrm{FI}_f = \sum_{l \in L} |\sum_{t \in f} \mathrm{SHAPValue}_{t,l}| \tag{1}$$

Comparing Ranking of Features

We wish to answer the question: Are the same features always the most important? To make an assessment on this, for each experiment, we compare the order of features, ranked by FI from most to least important. We choose to compare rankings as opposed to raw values to allow comparisons across different experiments. Therefore, we chose to use Kendall's rank correlation coefficient, or Kendall's τ[17], a non-parametric test to measure rank correlation. It is scored between -1 and 1, with identical rankings scoring 1 and opposite scoring -1.

$$\tau = \frac{\text{Number of concordant pairs} - \text{Number of discordant pairs}}{\text{Total number of pairs}} \tag{2}$$

For each TM-DS, for each of the 100 instances, we compare ordering of features ranked by FI by calculating Kendall's τ between each pairing of CM and summarize the results in Table 2a. We find that the three *Weighted-Ensembles* are the most similar to each other, as expected with their shared methodology. On the other hand, *All-Text* and *All-Text (Unimodal)*

are most dissimilar to those with the most focus on tabular features: *Stack-Ensembles*. In general, *All-Text* is more similar to the remaining CMs than *All-Text (Uni)*. Similarly, for each CM-DS, for each of the 100 instances, we calculate Kendall's τ between each pairing of TM and summarize the results in Table 2b. We see that changing TM causes less change in FI rankings than changing CM, with all pairs scoring a mean τ of 0.69-0.75. DeBERTa is the most different of the text models by a slight amount, likely because of the different way that it tokenizes, with spaces treated as part of the tokens.

	All-Text (Unimodal)	All-Text	WE (w=.25)	WE (w=.50)	WE (w=.75)
All-Text	.46 (.32)				
WE (w=.25)	.30 (.33)	.36 (.38)			
WE (w=.50)	.38 (.34)	.45 (.37)	.83 (.17)		
WE (w=.75)	.40 (.34)	.46 (.37)	.69 (.24)	.85 (.16)	
Stack Ensemble	.18 (.32)	.25 (.43)	.63 (.25)	.62 (.29)	.55 (.31)

(a)

	DistilBERT	BERT	DistilRoBERTa
BERT	.75 (.27)		
DistilRoBERTa	.73 (.29)	.71 (.29)	
DeBERTa	.70 (.28)	.69 (.28)	.69 (.30)

(b)

Table 2. Mean (SD) Kendall's τ comparing the FI rankings of each combination method (a) and each text model (b). Self-comparisons are trivially perfectly correlated ($\tau = 1$) and are omitted. Each result in (a) is an average of n3600 τs (100 instances by 9 DSs by 4 TMs), and each result in (b) is an average of n=5400 τs (100 instances by 9 DSs by 6 CMs). Note for each excluded experiment, n will be less by 100. WE refers to *Weighted-Ensemble* and w indicates the weighting of the text model.

Are Text and Tabular Features Weighted Differently?

Furthermore, we investigate whether the weighting of text and tabular features differ depending on the TM or CM used. As there is no guarantee of equal numbers of each feature type, we evaluate the difference between the average text FI and tabular FI, referring to this difference as Δ. We choose the median as FI is not normally distributed across instances.

$$\Delta = Median(\text{Text FI}) - Median(\text{Tabular FI}) \tag{3}$$

For each DS-TM combination, we test to see whether Δ differed between CMs. Similarly, for each DS-CM combination, we test to see whether Δ differed between TMs. Δ follows a non-normal distribution so we use the Kruskal-Wallis test[18]. The Kruskal-Wallis test, otherwise known as a one-way ANOVA on ranks, is a non-parametric test for hypothesis testing between subjects for a non-normal continuous variable. The null hypothesis is that all population medians are equal. $\hat{\varepsilon}^2_{\text{ordinal}}$ is the effect size of the Kruskal-Wallis test and indicates the percentage of the variation in the dependent variable that is explained by the independent variable. In Figure 4, we compare Δ's for each CM when DS=*fake* and TM=BERT (see Supplementary Fig. S1-S86 online for other experiments). The significant Kruskal-Wallis test ($p = 4.03e-50$) indicates that all medians are not equal, as is clear visually. Additionally, $\hat{\varepsilon}^2_{\text{ordinal}} = 0.40$ indicates that 40% of the variation in Δ is explained by the changes in the combination method. Figure 4 also shows non-significant pairwise tests, indicated with a bar. As Kruskal-Wallis tests if a

	airbnb	channel	fake	imdb	jigsaw	kick	prod	salary	wine
CM	.46 (.01)	.52 (.02)	.44 (.02)	.57 (.06)	.40 (.07)	.26 (.04)	.63 (.08)	.34 (.04)	.58 (.07)
TM	.11 (.05)	.15 (.05)	.24 (.08)	.05 (.04)	.32 (.06)	.08 (.02)	.19 (.10)	.09 (.07)	.09 (.04)

Table 3. We report the Mean (SD) effect size of the Kruskal-Wallis test ($\hat{\varepsilon}^2_{\text{ordinal}}$) when we test to see what proportion of the variance in the dependent variable, Δ, can be explained by changes in the independent variable. **Top row**: We set the independent variable to Combination Method (CM) and average $\hat{\varepsilon}^2_{\text{ordinal}}$ over the 4 KW tests (one for each Text Model (TM)), each containing n=600 instances. **Bottom row**: We set the independent variable to TM and we average $\hat{\varepsilon}^2_{\text{ordinal}}$ over the 6 KW tests (one for each CM), each containing n=400 instances. Note for each excluded experiment, n will be less by 100.

group of populations are the same, in order to compare two groups, Dunn's test[19] is appropriate. These tests have the Holm multiple-comparison adjustment applied. Considering $\hat{\varepsilon}^2_{\text{ordinal}}$ for each experiment, we report the summary statistics in Table 3. The [first/second] row indicates the proportion of variance in Δ explained by the [combination methods/text models], averaged over [text models/combination methods] to get a mean (standard deviation).

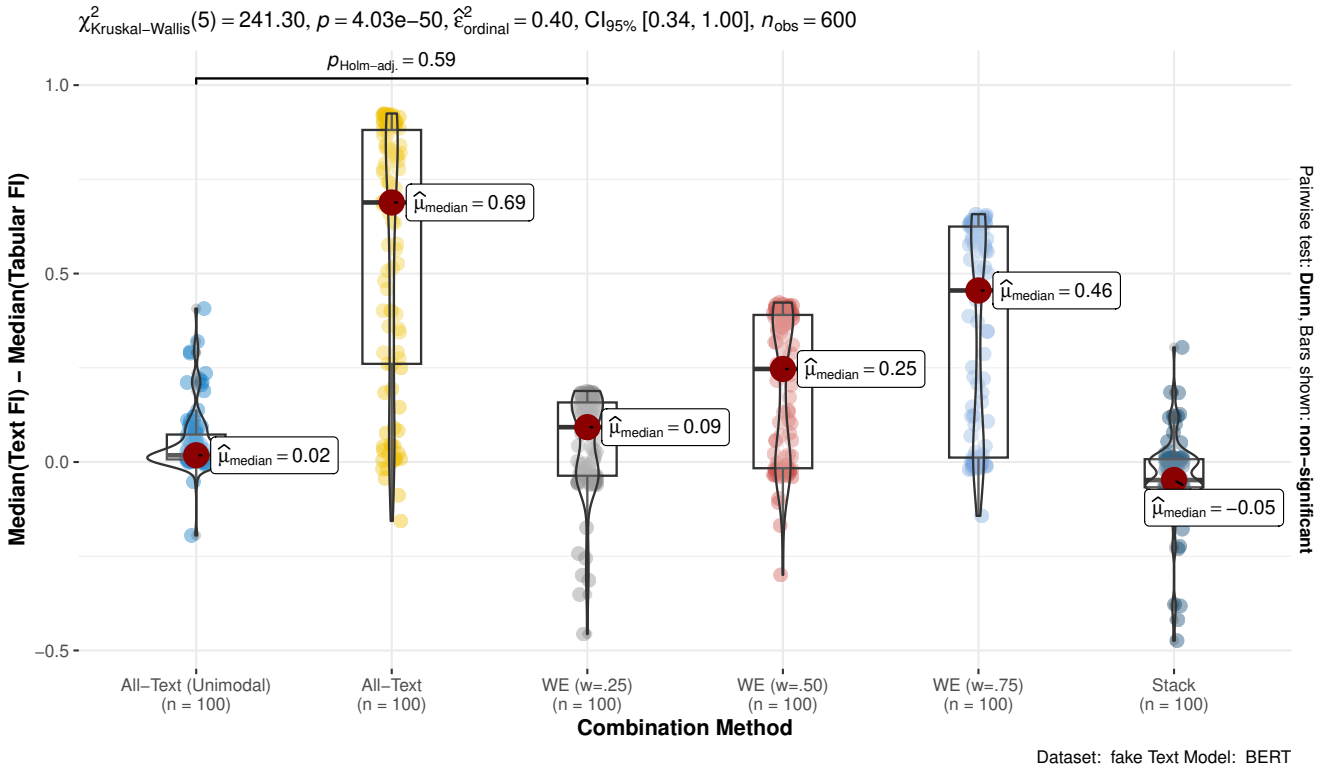

Figure 4. In order to compare which combination method assigns more relative importance to text or tabular features, we plot Δ = Median(Text FI) - Median(Tabular FI) on the y-axis and combination method on the x-axis. This example sets dataset=*fake* and text model=BERT. Each differently colored plot represents the distribution of Δ's for a single combination method, each with 100 colored dots representing Δ for each of the 100 instances. For each plot, the distribution is also represented by a violin plot, which emphasizes the non-normal distributions, a box plot, which indicates the spread of the data, and a labeled red dot, which indicates the median. A higher Δ indicates that a higher relative importance is assigned to text features over tabular features. The Kruskal-Wallis test statistic is significant ($p = 4.03e - 50$), meaning that the null hypothesis of all medians being equal fails. Of the pairwise tests, only [WE (w=.25), *All-Text*] is non-significant. WE refers to *Weighted-Ensemble* and *w* indicates the weighting of the text model.

Combination Method	Median Δ
All-Text (Unimodal)	.08 (.29)
All-Text	.15 (.37)
Weighted Ensemble (w=.25)	-.03 (.16)
Weighted Ensemble (w=.50)	.09 (.14)
Weighted Ensemble (w=.75)	.22 (.19)
Stack Ensemble	-.06 (.23)

(a)

Text Model	Median Δ
BERT	.05 (.29)
DistilBERT	.10 (.21)
DistilRoBERTa	.10 (.28)
DeBERTa	.06 (.25)

(b)

Table 4. To represent a single experiment, we take the median Δ and report the Mean (SD) for each (a) Combination Method and (b) Text Model. Higher values of Δ indicate a higher weighting of text features compared to tabular features. The median Δ for an experiment represents 100 instances. In (a) each result is an average of *n*=36 experiments (4 TMs by 9 DSs) and in (b) each is an average of *n*=54 experiments (6 CMs by 9 DSs). In (a), *w* indicates the weighting of the text model.

Taking the median Δ from each of the 199 experiments (the labeled dark red circles in Figure 4 show six of them), we average over text model and combination method and report the summary statistics in Table 4. We see that changing the text model has much less of an impact than changing the combination method. Looking at Table 4b, the three *Weighted-Ensembles* follow a predictable pattern, with the preference for text over tabular features as w increases from 0.25 through to 0.75. Specifically, we see Δ increasing as the weighting of the text model predictions increases. When comparing Δ for *All-Text* and *All-Text (Unimodal)*, we observe that unimodal explanations put a higher weight on tabular features (reflecting a lower Δ). *Stack-Ensembles*, on average, placed the lowest relative emphasis on text features. We hypothesize this to be a consequence of tabular features appearing twice in the training process, first at the initiation of the tabular model and then during the meta-model.

Explainability Consistency

In order to compare the quality of our multimodal masker with the unimodal masker, we first point out that both retain all the qualities of a SHAP explanation, namely local accuracy, missingness, and consistency. However, another method of explanation quality was proposed by Watson et al.[20], who looked at how consistent entire explanations were when regenerating SHAP values after retraining the same model architecture with a different random seed. A simple linear model, M, is trained to classify between the SHAP values of models a and b, and α is the number of comparisons made. The metric is scaled to be between 0 and 1, with a perfectly confused linear model with 50% accuracy scoring 1.

$$C = 1 - \frac{\Sigma_{(a,b)} 2 * |M(a,b) - 0.5|}{\alpha} \tag{4}$$

We choose to experiment with TM=DistilRoBERTa and DS=*fake* for strong performance and a similar number of text and tabular features. We finetune the text model four times with four different random seeds, then generate SHAP values from the same 1000 test set instances using both maskers using CM=*All-Text*. For each masker, a linear model is trained with 10-fold cross-validation to distinguish each of the six unique pairings ($\alpha = 6$), Table 5 shows the results.

Combination Method	Explanation Consistency
All-Text (Unimodal)	.659
All-Text	.853

Table 5. Using CM=*All-Text*, TM=DistilRoBERTa, and DS=*fake*, we compare explanation consistency when using multimodal and unimodal maskers.

Conclusion

In this study, we enable the generation of SHAP values for any text-tabular combination method for the first time. Our novel multimodal masker facilitates the masking of text and tabular features without first requiring conversion into a single modality. This opens up new avenues for explainability in multimodal models, allowing us to gain insights into new combination methods for the first time. We addressed the limitations of the existing unimodal masker, which restricted the generation of SHAP explanations to the All-Text combination method. By deferring input formation until after the masking stage and treating text and tabular features consistently regardless of how they are combined, we avoid the pitfalls of the unimodal approach.

Through extensive experimentation across nine datasets, we explore how changing text models and combination methods affect the resulting explanations. In particular, we find *All-Text* models favor textual features, whereas *Weighted-Ensembles* with a low text weighting w, and *Stack-Ensembles* - which see each tabular feature twice - favor tabular features. Across all datasets, changing the combination method had a greater impact on feature importance rankings than changing the text model. Finally, we retrain and regenerate SHAP values for CM=*All-Text*, TM=DistilRoBERTa, and DS=*fake* and find that our multimodal masker produces more consistent explanations than the unimodal masker.

Although we experimented with many datasets, text models, and several combination methods, there are still other factors that were omitted and could be experimented with. For *All-Text*, only a single style of string template was used. Future work could explore how explanations are affected when templates are varied. Furthermore, this work focuses on the initial implementation and subsequent analysis; further work could target speed and efficiency gains with specific configurations for certain model types and combination methods, as have been developed for the original SHAP library.

Data availability

Unprocessed datasets from Shi et al.[5] are available on GitHub at https://github.com/sxjscience/automl_multimodal_benchmark. We upload all processed datasets and all trained models to Hugging Face, available at https://huggingface.co/james-burton.

Code availability

Our code is available on GitHub here: https://github.com/jameswburton18/TextNTabularExplanations.

References

1. Yuan, W., Neubig, G. & Liu, P. BARTScore: Evaluating Generated Text as Text Generation. *Adv. Neural Inf. Process. Syst.* **33**, 27263–27277 (2021).
2. Rai, A. Explainable AI: from black box to glass box. *J. Acad. Mark. Sci.* **48**, 137–141, DOI: 10.1007/s11747-019-00710-5 (2020).
3. European Commission. Proposal for a Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts. *Com(2021)* **0106**, 1–108 (2021).
4. Sleeman, W. C., Kapoor, R. & Ghosh, P. Multimodal Classification: Current Landscape, Taxonomy and Future Directions. *ACM Comput. Surv.* **55**, 1–31, DOI: 10.1145/3543848 (2023).
5. Shi, X., Mueller, J., Erickson, N., Li, M. & Smola, A. J. Benchmarking Multimodal AutoML for Tabular Data with Text Fields. *Proc. Neural Inf. Process. Syst. Track on Datasets Benchmarks* **1** (2021).
6. Liu, P. *et al.* Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.* **55**, 1–35, DOI: 10.1145/3560815 (2023).
7. Gu, K. & Budhkar, A. Multimodal-Toolkit: A Package for Learning on Tabular and Text Data with Transformers. In *Multimodal Artificial Intelligence, MAI Workshop 2021 - Proceedings of the 3rd Workshop*, 69–73, DOI: 10.18653/v1/2021.maiworkshop-1.10 (Association for Computational Linguistics (ACL), 2021).
8. Parcalabescu, L. & Frank, A. MM-SHAP: A Performance-agnostic Metric for Measuring Multimodal Contributions in Vision and Language Models & Tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4032–4059, DOI: 10.18653/v1/2023.acl-long.223 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2023).
9. Lyu, Y., Liang, P. P., Deng, Z., Salakhutdinov, R. & Morency, L. P. DIME: Fine-grained Interpretations of Multimodal Models via Disentangled Local Explanations. *AIES 2022 - Proc. 2022 AAAI/ACM Conf. on AI, Ethics, Soc.* 455–467, DOI: 10.48550/arxiv.2203.02013 (2022).
10. Lundberg, S. & Lee, S.-I. A Unified Approach to Interpreting Model Predictions. *Adv. Neural Inf. Process. Syst.* **2017-Decem**, 4766–4775 (2017).
11. Shapley, L. S. *A Value for N-Person Games* (RAND Corporation, 1952).
12. Molnar, C. Interpretable Machine Learning. A Guide for Making Black Box Models Explainable. *Book* 247 (2020).
13. Ke, G. *et al.* LightGBM: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017-Decem**, 3147–3155 (2017).
14. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. for Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.* **1**, 4171–4186 (2018).
15. He, P., Liu, X., Gao, J. & Chen, W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *ICLR 2021 - 9th Int. Conf. on Learn. Represent.* (2020).
16. Sanh, V., Debut, L., Chaumond, J. & Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv* **abs/1910.0**, DOI: https://doi.org/10.48550/arXiv.1910.01108 (2019).
17. Kendall, M. G. A New Measure of Rank Correlation. *Biometrika* **30**, 81, DOI: 10.2307/2332226 (1938).
18. Kruskal, W. H. & Wallis, W. A. Use of Ranks in One-Criterion Variance Analysis. *J. Am. Stat. Assoc.* **47**, 583–621, DOI: 10.1080/01621459.1952.10483441 (1952).

19. Dunn, O. J. Multiple Comparisons Among Means. *J. Am. Stat. Assoc.* **56**, 52, DOI: 10.2307/2282330 (1961).

20. Watson, M., Hasan, B. A. S. & Moubayed, N. A. Agree to Disagree: When Deep Learning Models with Identical Architectures Produce Distinct Explanations. *Proc. - 2022 IEEE/CVF Winter Conf. on Appl. Comput. Vision, WACV 2022* 1524–1533, DOI: 10.1109/WACV51458.2022.00159 (2022).

Acknowledgements

This work was supported by Innovate UK grant number 10027358. Figures 1, 2, 3 were created with BioRender.com.

Author contributions

J.B. and N.A. conceived the idea. J.B. produced the code and conducted the experiments. Both authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- TextNTabularNatureSciRep1.pdf