Multimodal-Toolkit: A Package for Learning on Tabular and Text Data with Transformers

Ken Gu
Georgian
ken.gu@georgian.io

Akshay Budhkar
Georgian
akshay@georgian.io

Abstract

Recent progress in natural language processing has led to Transformer architectures becoming the predominant model used for natural language tasks. However, in many real-world datasets, additional modalities are included which the Transformer does not directly leverage. We present Multimodal-Toolkit,[1] an open-source Python package to incorporate text and tabular (categorical and numerical) data with Transformers for downstream applications. Our toolkit integrates well with Hugging Face's existing API such as tokenization and the model hub[2] which allows easy download of different pre-trained models.

1 Introduction

In recent years, Transformers (Vaswani et al., 2017) have become popular for model pre-training (Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2019) and have yielded state-of-the-art results on many natural language processing (NLP) tasks. In addition, well-documented Transformer libraries such as Hugging Face Transformers (Wolf et al., 2020), and AllenNLP (Gardner et al., 2018) have democratized NLP, making it easier to productionize and experiment on Transformers.

However, there are not a lot of comprehensive tools for Transformers to work with tabular data. Often in real-world datasets, there are tabular data as well as unstructured text data which can provide meaningful signals for the task at hand. For instance, in the small example in Figure 1, each row is a data point. Columns `Title` and `Review Text` contain text features, columns `Division Name`, `Class Name`, and `Department Name` contain categorical features, and the `Age` column is a numerical feature. To the best of our knowledge, no tool exists that makes it simple for Transformers to handle this extra modality. Therefore, given the advances of Transformers for natural language tasks and the maturity of existing Transformer libraries, we introduce Multimodal-Toolkit, a lightweight Python package built on top of Hugging Face Transformers. Our package extends existing Transformers in the Hugging Face's Transformers library to seamlessly handle structured tabular data while keeping the existing tokenization (including subword segmentation), experimental pipeline, and pre-trained model hub functionalities of Hugging Face Transformers. We show the effectiveness of our toolkit on three real-world datasets.

Age	Title	Review Text	Division Name	Class Name	Department Name
21	Pretty but not for me	This sweater is very pretty, i love the knit a...	General Petite	Sweaters	Tops
36	Beautiful	As beautiful as in the picture. couldn't go wr...	General	Skirts	Bottoms
47	Adorable and comfortable!	Just bought this in black at my local store an...	General	Knits	Tops
29	Must have, elegant, chic	This top! i was hesitant to try this on becaus...	General	Blouses	Tops
38	Very flattering fit	This is a great pair of trousers for work but ...	General Petite	Pants	Bottoms

Figure 1: An example of a clothing review classification dataset. Each row is a data point consisting of text, categorical features, and numerical features.

2 Related Work

There have been several proposed Transformer models that aim to handle text features and additional features of another modality. For pre-trained Transformers on images and text, models such as ViLBERT (Lu et al., 2019) and VLBERT (Su et al., 2020) are mainly the same as the original BERT model but treat the extra image modality as additional tokens to the input. These models require pre-training on multimodal image and text data. On the other hand, while treating image features

[1] Github: https://git.io/JO5a6
[2] https://huggingface.co/docs

Proceedings of the Third Workshop on Multimodal Artificial Intelligence, pages 69–73

as additional input tokens, MMBT (Kiela et al., 2019) proposes to use pre-trained BERT directly and fine-tune on image and text data. This is similar to Multimodal-Toolkit in which no pre-training on text and tabular data is needed.

Likewise, Transformers have been adapted to align, audio, visual, and text modalities in which there is a natural ground truth alignment. MulT (Tsai et al., 2019) is similar to ViLBert in which co-attention is used between pairs of modalities but also includes temporal convolutions so that input tokens are aware of their temporal neighbors. Meanwhile, Rahman et al. (2020) injects cross modality attention at certain Transformer layers via a gating mechanism.

Finally, knowledge graph embeddings have also been effectively combined with input text tokens in Transformers. Ostendorff et al. (2019) combines knowledge graph embeddings on authors with book titles and other metadata features via simple concatenation for book genre classification. On the other hand, for more general language tasks, ERNIE (Zhang et al., 2019) first matches the tokens in the input text with entities in the knowledge graph. With this matching, the model fuses these embeddings to produce entity-aware text embeddings and text-aware entity embeddings.

However, these models do not capture categorical and numerical data explicitly. Hugging Face does include LXMERT (Tan and Bansal, 2019) to handle language and vision modality but this can not be easily adapted for categorical and numerical data. Nevertheless, existing multimodal Transformer models do give good insights into how to combine categorical and numerical features. ViLBERT and VLBERT for example include image modality as input tokens which lead to one of our simple baseline of categorical and numerical features as additional token inputs to the model. Likewise, the gating mechanism Rahman et al. (2020), attention, and different weighting schemes have all been shown to be useful in combining different modalities.

3 Design

The goal of Multimodal-Toolkit is to allow users to quickly adapt state-of-the-art Transformer models for situations involving text and tabular data which occur often in real-world datasets. Moreover, we want to bring the benefits of Transformers to more use cases while making it simple for users

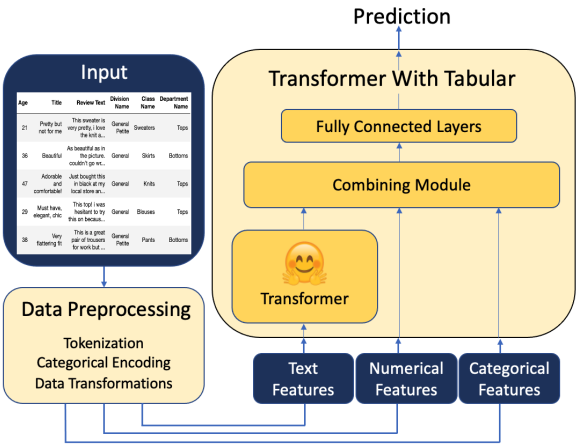

Figure 2: The framework of Multimodal-Toolkit. There is a data processing module that outputs processed text, numerical, and categorical features that are then fed as input to our Transformer With Tabular module consisting of a Hugging Face Transformer and our combining module.

of Hugging Face Transformers to adopt. Therefore, we maintain the existing interface of the popular Hugging Face Transformers library.

This design enables us to easily include more Transformer models, leverage strengths of specific models, use a feature-rich training pipeline, and integrate the thousands of community trained models on Hugging Face's model hub. We support a variety of Transformers (e.g. BERT, ALBERT, RoBERTa, XLNET) for both classification and regression tasks. All together, this becomes a reusable Transformer With Tabular component. We also provide a data preprocessing module for categorical and numerical features. An overview of the system is shown in Figure 2. Currently, the library supports PyTorch Transformers implementations.

3.1 Combining Module

We implement a combining module that is model agnostic that takes as input, $\boldsymbol{x}$, the text features outputted from a Transformer model and preprocessed categorical ($\boldsymbol{c}$) and numerical ($\boldsymbol{n}$) features, and outputs a combined multimodal representation $\boldsymbol{m}$. Although existing multimodal Transformers incorporate cross-modal attention inside middle Transformer layers, we choose the design in which the modality combination comes after the Transformer because this module can be easily included without much adaptation of the existing Hugging Face Transformer interface and can be easily extended to new Transformers included in the future.

Inside the combining module, we implement var-

Combine Feature Method	Equation
Text only	$\boldsymbol{m} = \boldsymbol{x}$
Concat	$\boldsymbol{m} = \boldsymbol{x}\|\|\boldsymbol{c}\|\|\boldsymbol{n}$
Individual MLPs on categorical and-numerical features then concat (MLP + Concat)	$\boldsymbol{m} = \boldsymbol{x}\|\|\text{MLP}(\boldsymbol{c})\|\|\text{MLP}(\boldsymbol{n})$
MLP on concatenated categorical and numerical features then concat (Concat + MLP)	$\boldsymbol{m} = \boldsymbol{x}\|\|\text{MLP}(\boldsymbol{c}\|\|\boldsymbol{n})$
Attention on categorical and numerical features (Attention)	$\boldsymbol{m} = \alpha_{x,x}\boldsymbol{W}_x\boldsymbol{x} + \alpha_{x,c}\boldsymbol{W}_c\boldsymbol{c} + \alpha_{x,n}\boldsymbol{W}_n\boldsymbol{n}$ $\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(\boldsymbol{a}^T[\boldsymbol{W}_i\boldsymbol{x}_i\|\|\boldsymbol{W}_j\boldsymbol{x}_j]))}{\sum_{k\in\{x,c,n\}}\exp(\text{LeakyReLU}(\boldsymbol{a}^T[\boldsymbol{W}_i\boldsymbol{x}_i\|\|\boldsymbol{W}_k\boldsymbol{x}_k]))}$
Gating on categorical and numerical features and then sum (Rahman et al., 2020) (Gating)	$\boldsymbol{m} = \boldsymbol{x} + \alpha\boldsymbol{h}$ $\boldsymbol{h} = \boldsymbol{g}_c \odot (\boldsymbol{W}_c\boldsymbol{c}) + \boldsymbol{g}_n \odot (\boldsymbol{W}_n\boldsymbol{n}) + b_h$ $\alpha = \min(\frac{\|\|\boldsymbol{x}\|\|_2}{\|\|\boldsymbol{h}\|\|_2} * \beta, 1)$ $\boldsymbol{g}_i = \text{R}(\boldsymbol{W}_{g_i}[\boldsymbol{i}\|\|\boldsymbol{x} \ + b_i)$ where β is a hyperparameter and R is an activation function
Weighted feature sum on text, categorical, and numerical features (Weighted Sum)	$\boldsymbol{m} = \boldsymbol{x} + \boldsymbol{w}_c \odot \boldsymbol{W}_c\boldsymbol{c} + \boldsymbol{w}_n \odot \boldsymbol{W}_n\boldsymbol{n}$

Table 1: The included combining methods in the combining module. Uppercase bold letters represent 2D matrices, lowercase bold letters represent 1D vectors. b is a scalar bias, $\boldsymbol{W}$ represents a weight matrix, and $||$ is the concatenation operator. Please see Rahman et al. (2020) for details on the gating mechanism.

Dataset	Task	Size	T	C	N
Airbnb	Regression	64k	3	74	15
Clothing	Classification	15k	2	3	3
PetFinder	Classification	28k	2	14	5

Table 2: Statistics of the datasets involved in experiments. T is the number of text columns. C is the number of categorical features, and N is the number of numerical features.

ious methods of combining the different representations in their respective feature spaces into one unified representation. These methods are inspired by the related work in multimodal Transformers as well as straightforward reasonable baselines such as concatenation and multi-layer perceptron (MLP) concatenation. Given a pre-trained Transformer, the parameters of the combining module and Transformer are trained based on the supervised task. In other words, the Transformer is further fine-tuned. The included methods are shown in Table 1.

4 Experiments

In this section, we study the effectiveness of leveraging tabular features on data with text and tabular data. We evaluate Multimodal-Toolkit on three real-world datasets from Kaggle.

4.1 Datasets

Regression: For regression, we use the Melbourne Airbnb Open Data (Airbnb) dataset (Xie, 2019) for the task of listing price prediction. Each data example is an Airbnb listing. Text features include the name of the listing, the summary of the listing, and a host description.

Binary Classification: For binary regression, we use Women's E-Commerce Clothing Reviews (Clothing) (Brooks, 2018). The source of the reviews is anonymous. Data examples consist of a review, a rating, the clothing category of the product etc. The goal is to predict if the review is recommending the product.

Multiclass Classification: Finally, we also include the PetFinder.my Adoption Prediction (PetFinder) dataset (PetFinder.my, 2018). Given the listing information of a pet set for adoption, the goal is to predict the speed at which a pet will be adopted, represented as 5 classes. Text features include the listing description and the pet name.

4.2 Experimental Setting

For experiments, we test each combining feature method described in Table 1. In addition, as mentioned in Section 2 we test a baseline in which the categorical and numerical features are also treated

Method	Airbnb		Clothing		PetFinder	
	RMSE	MAE	F1	AUPRC	$F1_{macro}$	$F1_{micro}$
Text Only	254.0	82.74	0.957	0.992	0.088	0.281
Unimodal	245.2	79.34	**0.968**	**0.995**	0.089	0.283
Concat	239.3	**65.68**	0.958	0.992	0.199	0.362
MLP + Concat	**237.3**	66.73	0.959	0.992	0.244	0.352
Concat + MLP	238.0	65.66	0.959	0.992	0.176	0.344
Attention	246.3	74.72	0.959	0.992	0.254	0.375
Gating (Rahman et al., 2020)	237.8	66.64	0.961	0.994	**0.275**	0.375
Weighted Sum	245.2	71.19	0.962	0.994	0.266	**0.380**

Table 3: Comparison of combining methods with results on regression and classification tasks. For each metric, the best performing model is in bold. For regression we use Root-mean-squared Error (RMSE) and MAE (Mean Absolute Error). In both cases, lower is better. For binary classification, we report F1 score and area under the precision-recall curve (AUPRC). Meanwhile, for multiclass classification, we use $F1_{macro}$ and $F1_{micro}$. In all classification metrics, higher is better.

as text columns. For example, for the situation in Figure 1, the text representing categorical features in `Division Name`, `Class Name`, and `Department Name` as well the numerical value in `Age` would all be tokenized and be treated as additional inputs to the Transformer. We denote this baseline as Unimodal.

For the Clothing Review dataset, we use `bert-base-uncased` as our Transformer and tokenizer. For the Airbnb dataset and Pet Adoption datasets, because there are some data points containing non-English text, we use `bert-base-multilingual`. We keep the training settings consistent for a given dataset. We train for 5 epochs and perform 4-fold-cross-validation, reporting the mean performance. For regression, we use a learning rate of 3e-3 while for classification tasks we use a learning rate of 5e-5. We report the results in Table 3.

4.3 Results

From Table 3, we observe the effectiveness of incorporating tabular features across different tasks and datasets. For each real-world dataset, the text-only baseline is the worst performing model. This shows using only text data with Transformers may be insufficient when extra tabular data is available.

However, how much the performance improves by leveraging Tabular features depends on the dataset. In the case of the Clothing Review dataset, the text of the review was already a very strong signal to the prediction, extra tabular features did not improve the performance much. We hypothesize the strong performance of the text only baseline may be due to the task of classifying review recommendation simplifying to sentiment classification, which the text modality provides the strongest signals. On the other hand, for the PetFinder dataset, the text description of the animal may not be sufficient to predict adoption speed. Rather, it is tabular features such as the age or the breed of the pet. Furthermore, the relative low raw performance of PetFinder dataset could be attributed to the difficulty of the task as a forecasting problem.

Additionally, although the Unimodal baseline is the best for the clothing dataset, this method does not appear to scale well when the number of categorical and numerical features increases or when the extra features' text representation does not reveal obvious semantic meaning.

5 Conclusion

This paper presents Multimodal-Toolkit, an open-source Python library powered by Hugging Face Transformers to learn on data that contains both text and tabular data. We show the effectiveness of incorporating tabular data and treating it as a separate modality with the already powerful Transformers. The modular design and shared API with Hugging Face allow users quick access to Hugging Face's community uploaded Transformer models.

For future work, we aim to include support for more Transformers and integrate the combining module at earlier layers in the Transformer. We hope the toolkit brings more research attention to this data scenario and we welcome open-source contributions to the project.

References

Nick Brooks. 2018. Women's e-commerce clothing reviews.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Ethan Perez, and Davide Testuggine. 2019. Supervised Multimodal Bitransformers for Classifying Images and Text. *arXiv e-prints*, page arXiv:1909.02950.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, volume 32, pages 13–23. Curran Associates, Inc.

Malte Ostendorff, Peter Bourgonje, Maria Berger, Julian Moreno-Schneider, Georg Rehm, and Bela Gipp. 2019. Enriching BERT with Knowledge Graph Embeddings for Document Classification. *arXiv e-prints*, page arXiv:1909.08402.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

PetFinder.my. 2018. Petfinder.my adoption prediction.

Wasifur Rahman, Md Kamrul Hasan, Sangwu Lee, AmirAli Bagher Zadeh, Chengfeng Mao, Louis-Philippe Morency, and Ehsan Hoque. 2020. Integrating multimodal information in large pretrained transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2359–2369, Online. Association for Computational Linguistics.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6558–6569, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tyler Xie. 2019. Melbourne airbnb open data.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.