TAB2TEXT - A framework for deep learning with tabular data

Tong Lin*[1], Jason Yan*[1], David Jurgens[1,2], and Sabina Tomkins[1]
[1]School of Information, University of Michigan
[2]Department of Computer Science & Engineering, University of Michigan
{tonglin, jasonyan, jurgens, stomkins}@umich.edu

Abstract

Tabular data, from public opinion surveys to records of interactions with social services, is foundational to the social sciences. One application of such data is to fit supervised learning models in order to predict consequential outcomes, for example: whether a family is likely to be evicted, whether a student will graduate from high school or is at risk or dropping out, and whether a voter will turn out in an upcoming election. While supervised learning has seen drastic improvements in performance with advancements in deep learning technology, these gains are largely lost on tabular data which poses unique difficulties for deep learning frameworks. We propose a technique for transforming tabular data to text data and demonstrate the extent to which this technique can improve the performance of deep learning models for tabular data. Overall, we find modest gains (1.5% on average). Interestingly, we find that these gains do not depend on using large language models to generate the text.

1 Introduction

Tabular data is abundant in the social sciences and for social good applications (Bonica, 2018; Roscigno and Preito-Hodge, 2021; Tiehen et al., 2020). One source of tabular data is public opinion surveys, which are used to monitor public opinion on everything from public health to the economy. Another common source of tabular data is interactions with public services, for example, when a family receives assistance with food or paying rent, this interaction can populate a table. This data is also used to train supervised learning models which have been designed and deployed in the service of the public good (Bloise et al., 2021; Bonica, 2018; Sheetal and Savani, 2021; Lin et al., 2023). For example, supervised learning has been deployed to detect at-risk high-school students (Lakkaraju et al., 2015) and prevent eviction-caused homelessness (Vajiac et al., 2024). However, while deep learning models have often yielded transformative results when applied to supervised learning problems with homogeneous data sources (e.g., in the text domain), such models generally offer sub-optimal results applied to tabular data (Shwartz-Ziv and Armon, 2022). In part, this performance is due to tabular data often being composed of mixed feature types from categorical features to numerical. Furthermore, we generally see deep learning models suffer from overparametrization and sensitivity to particular hyperparameter combinations on tabular data (Shwartz-Ziv and Armon, 2022; Shavitt and Segal, 2018; Arik and Pfister, 2021). Here, we propose to close this gap by introducing a new method to adapt tabular data for large language models (LLMs) by transforming tabular data into text data.

This paper makes three contributions: (1) we introduce TAB2TEXT, a deep learning framework for tabular data that exploits both tabular and text representations, enhancing learning capability through LLM-generated narratives[1]; (2) we show that including text representations consistently outperforms the baselines which use tabular data only; and (3) models leveraging only text representations underperform compared to those using only tabular data.

2 Problem Statement

Consider a dataset with d columns and N rows:

$$Y \sim \{C_1 : X_1, C_2 : X_2, \ldots, C_d : X_d\} \qquad (1)$$

where Y is a label, $C_{i\in[d]}$ are column names, and $X_{i\in[d]}$ are the associated columns. We assume that these columns are a mix of binary, categorical, and continuous data types. Our final goal is to train a function which takes some form of X as input and generates predictions $\hat{Y}$.

* denotes equal contribution

[1] Relevant code available here: https://github.com/politechlab/tab2text.git

Findings of the Association for Computational Linguistics: EACL 2024, pages 12925–12935
November 12-16, 2024 ©2024 Association for Computational Linguistics

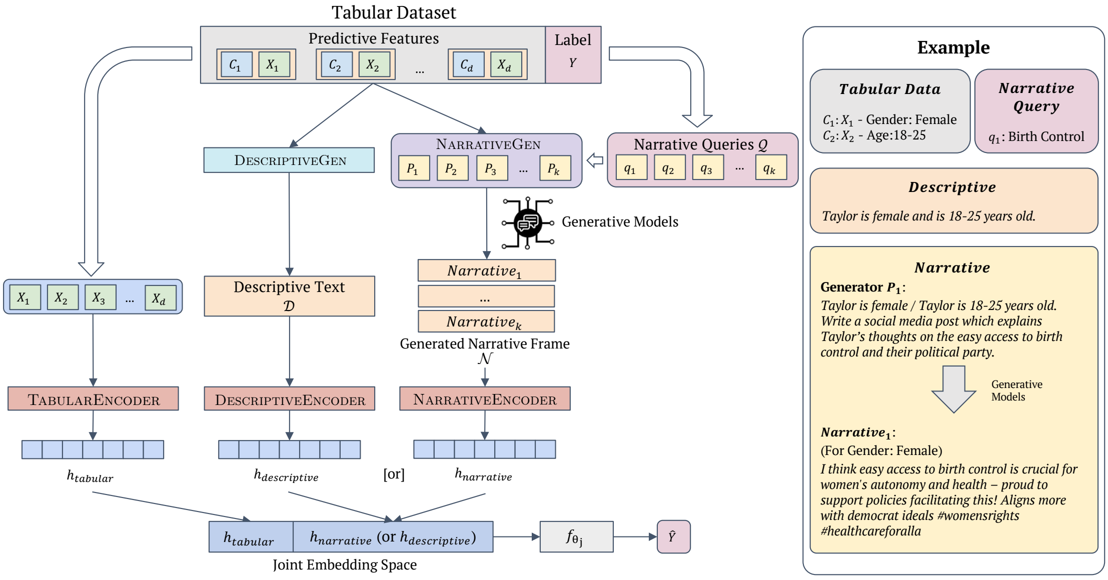

Tabular Dataset
Predictive Features
C_1 X_1 C_2 X_2 ... C_d X_d
Label
Y
DescriptiveGen
NarrativeGen
P_1 P_2 P_3 ... P_k
Narrative Queries Q
q_1 q_2 q_3 ... q_k
Generative Models
X_1 X_2 X_3 ... X_d
Descriptive Text
$\mathcal{D}$
$Narrative_1$
...
$Narrative_k$
Generated Narrative Frame
$\mathcal{N}$
TabularEncoder
DescriptiveEncoder
NarrativeEncoder
$h_{tabular}$
$h_{descriptive}$
[or]
$h_{narrative}$
$h_{tabular}$
$h_{narrative}$ (or $h_{descriptive}$)
Joint Embedding Space
f_{θ_j}
$\hat{Y}$
Example
Tabular Data
C_1: X_1 - Gender: Female
C_2: X_2 - Age:18-25
Narrative Query
q_1: Birth Control
Descriptive
Taylor is female and is 18-25 years old.
Narrative
Generator P_1:
Taylor is female / Taylor is 18-25 years old. Write a social media post which explains Taylor's thoughts on the easy access to birth control and their political party.
Generative Models
$Narrative_1$:
(For Gender: Female)
I think easy access to birth control is crucial for women's autonomy and health – proud to support policies facilitating this! Aligns more with democrat ideals #womensrights #healthcareforalla

Figure 1: General design of TAB2TEXT: Tabular data is transformed into text using one of the two generators (Descriptive or Narrative), which is then encoded into a dense embedding and fused with the tabular values to use for prediction; the full model is trained end-to-end. An example of each step is on the right.

3 Our Approach - TAB2TEXT

Consider the setting of predicting one's income. Given access to someone's social media posts, this may be relatively easy. A person of very high income may use language related to luxury goods and services, while a lower-income person may employ vocabulary that reflects different socioeconomic experiences. Similarly, consider the simple task of predicting whether a flower is a rose or an iris. The text descriptions of each flower would differ greatly, given a dataset of descriptions of roses and descriptions of irises, a LLM model would likely learn to differentiate them easily.

Without access to a text dataset for a target population, our intuition is that LLMs already contain enough general knowledge such that they can generate related text given limited information. For example, given a person's occupation, an LLM may be able to generate an example social media post which could be indicative of income; given a person's age and race, a LLM can hypothesize about how they may feel about politically relevant issues—hypotheses which could help with the political inference tasks; or, given the size and color of a flower, an LLM should be able to generate a description.

Our approach (TAB2TEXT) generates versions of tabular data as text which allow us to learn more sophisticated predictive functions and leverage textual embedding spaces. Additionally, we propose inference architectures which combine textual with tabular data. Combining textual representations from LLMs with tabular data should enable models to leverage the knowledge capabilities of LLMs while grounding them in the specific task distribution, potentially leading to better performance than using either representation alone. This complete framework is shown in Figure 1.

3.1 Tabular to text data

We explore two techniques for transforming tabular data to text data. In the first, we encode each tabular feature (column descriptor + specific value) as a text statement. In the second, we explore more complex narratives. These complex narratives are designed to capture concepts correlated to the target variable, and introduce a novel open problem of how to generate predicatively-helpful narrative text. These techniques can be explained through the following components:

- DESCRIPTIVEGEN - Let $\mathcal{D}$ be a dataset of descriptive text. DESCRIPTIVEGEN is a method for transforming a tabular attribute, such as an age, into a text statement, such as "My age is 18" or "I am 18 years old". $\mathcal{D} = \text{DESCRIPTIVEGEN}(X_1, X_2, \ldots, X_d)$.

ANES	
ANES Narrative Queries	**Description**
Abortion	Thoughts on access to unconditional abortion services.
Birth Control	Thoughts on the easy access to birth control.
Gun Control	Thoughts on rolling-back gun ownership regulations.
Immigration	Thoughts on recent immigrants in the US when it comes to the job market.
Sex Education	Thoughts on introducing sexual education at early childhood.
Tax	Thoughts on what the government should invest in if there's a need to increase taxes.
Partisanship	Thoughts on the given individual's political party.
Income	
Income Narrative Queries	**Description**
Vacation	Thoughts on the given individual's income and their dream vacation.
Home	Thoughts on the given individual's income and their ideal home.

Table 1: Summary of narrative queries covered in ANES and Income

- DescriptiveEncoder - Let $h_{descriptive}$ be an embedding of textual data. DescriptiveEncoder is a function which takes descriptive text as input and outputs a vector representation of that text. $h_{descriptive} = \text{DescriptiveEncoder}(\mathcal{D})$

- Narrative Queries $\mathcal{Q}$ - Let, **narrative queries** $\mathcal{Q} = \{q_1, q_2, \ldots, q_k\}$ be a set of queries where each $q \in \mathcal{Q}$ describes a concept which we believe is correlated with the target variable Y. For example, one's vacation experiences (q_{vacation}) are likely determined by and correlated with their income (Y).

- NarrativeGen - A method for generating a prompt from a query. This prompt is used in conjunction with an LLM to generate text. The k narrative queries are paired with the column names and values to obtain k **narrative generators**:$\{P_1, P_2, \ldots, P_k\} = \{q_j, \{C_1 : X_1, \ldots, C_d : X_d\}\}_{j=1}^{k}$. These k narrative generators are fed into a generative LLM g to acquire a **generated narrative** that provides information about the label Y. From this, we obtain the **generated narrative frame** $\mathcal{N}$ by composing the k generated narratives. That is, $\mathcal{N} = g(P_1) \circ g(P_2) \circ \ldots \circ g(P_k)$.

- NarrativeEncoder - Let $h_{narrative}$ be an embedding of textual data. NarrativeEncoder is a function which takes descriptive text as input and outputs a vector representation of that text. $h_{narrative} = \text{NarrativeEncoder}(\mathcal{N})$.

3.2 Tabular Encoders

It may be advantageous to create a representation for tabular input which goes beyond the raw input. That is, we would like to produce a representation $h_{tabular} = \text{TabularEncoder}(X_1, X_2, \ldots, X_d)$, which can be fused with $h_{descriptive}$ or $h_{narrative}$ in a joint representation space. We tested two tabular encoders: (1) a feature concatenation embedding of the raw tabular values and (2) TabNet (Arik and Pfister, 2021). In practice, however, we find that the feature concatenation embeddings consistently outperform TabNet embeddings.

3.3 Inference architecture

Our inference architecture fuses tabular and textual data. We learn a joint function f_{θ_j} that maps the embedding space of the narrative (or descriptive) and tabular embeddings onto the label space Y: $\hat{Y}_{narrative} = f_{\theta_j}(h_{narrative}, h_{tabular})$. We experiment with different methods for modeling and learning these representations. However, generally, the model is as shown in Figure 1.

4 Empirical Evaluation

We evaluate Tab2Text with two common social science datasets. Additionally, we compare both the descriptive and narrative approaches to generating text. Finally, we analyze both the influence of specific LLMs and inference architectures.

Data We explore two datasets: the American National Election Studies (ANES) (American National Election Studies, 2021) and Income (Becker and Kohavi, 1996). ANES consists of responses to surveys about political beliefs and behaviors, as well as demographic information. We use a version with 3,905 respondents from 2020. Income consists of records from the 1994 U.S. Census Bureau database and contains demographic information about individuals. This dataset has been used for

		Deep Learning							Logistic Reg.	
		RoBERTa	DistilBERT	PoliBERT	Longformer	TF-IDF	BERT*+TN	AutoMM	TF-IDF	Avg
Descriptive		0.639	0.639	0.641	0.639	0.639	0.634	0.637	0.642	0.639
Narrative	gpt-4-1106-preview	0.640	0.641	0.640	0.614	0.640	0.631	0.641	0.645	0.634
	gpt-4-1106-preview (FT)	0.641	0.640	0.640	0.638	0.648	0.633	0.637	0.649	0.641
	claude-3-opus-20240229	0.640	0.625	0.640	0.638	0.639	0.634	0.632	0.641	0.636
	gemini-pro	0.641	0.627	0.640	0.640	0.637	0.638	0.640	0.641	0.638
	llama-2-7b-chat	0.640	0.597	0.640	0.639	0.638	0.632	0.636	0.641	0.633
	Avg	0.640	0.628	0.640	0.635	0.640	0.634	0.637	0.643	
	Baselines - tabular only	Deep Learning: 0.638 TabNet (Arik and Pfister, 2021): 0.626				Logistic Regression: 0.641 TabularPredictor (Erickson et al., 2020): 0.637				

Table 2: Results for ANES. Macro F1 scores on the test set are reported. Highlighted results are those which are statistically significantly better than the baseline within the same setting (i.e., deep learning or logistic regression) according to McNemar's Test (p-value < 0.05). BERT* denotes the results averaged across all three BERT models deployed with TabNet as the tabular encoder. The bottom row contains the results for the baseline deep learning and logistic regression models using solely tabular embeddings. Logistic Reg. is a logistic regression model that is fitted on narrative (or descriptive) embeddings (as TF-IDF vectors)+tabular embeddings. AutoMM is from the AutoGluon framework (Erickson et al., 2020; Tang et al., 2024), which utilizes a FT-Transformer architecture (Gorishniy et al., 2021) for encoding tabular inputs, while leveraging the ELECTRA-base discriminator model (Clark, 2020) for text encoding.

		Deep Learning							Logistic Reg.	
		RoBERTa	DistilBERT	PoliBERT	Longformer	TF-IDF	BERT*+TN	AutoMM	TF-IDF	Avg
Descriptive		0.757	0.763	0.766	0.757	0.760	0.762	0.766	0.757	0.761
Narrative	gpt-4-1106-preview	0.757	0.747	0.767	0.755	0.761	0.757	0.765	0.759	0.759
	Avg	0.757	0.755	0.767	0.756	0.761	0.760	0.766	0.759	
	Baselines - tabular only	Deep Learning: 0.757 TabNet (Arik and Pfister, 2021): 0.747				Logistic Regression: 0.755 TabularPredictor (Erickson et al., 2020): 0.753				

Table 3: Results for Income. The rows and columns follow the same scheme as in Table 2.

machine learning tasks with the goal of predicting whether a person earns more than $50,000 per year. We randomly sampled 10,000 individuals to create the version used for our study (see Appendix).

Implementation Details Several narrative encoder models were tested: DistilBERT (Sanh et al., 2019), RoBERTa (Liu et al., 2019), and PoliBERT (Gupta et al., 2020). PoliBERT is a RoBERTa-based model that was fine-tuned on political twitter data, and which likely has context-specific information for these social science data. Additionally, we employed AutoMM and TabularPredictor models from the AutoGluon framework (Erickson et al., 2020; Tang et al., 2024). AutoMM utilizes a FT-Transformer architecture (Gorishniy et al., 2021) for encoding tabular inputs, while leveraging the ELECTRA-base discriminator model (Clark, 2020) for text encoding. We used the following narrative generator formulation to produce narrative text.

Narrative Generator Formulation The narrative generator consist of two main components: tabular data (both column names and values) and the narrative queries that solicits information about the relevant outcome. We describe these queries in Table 1.

We introduce the gender-neutral name "Taylor" and use third-person narration to present the demographics, shifting LLM's perspective from expressing its own views to considering those of another. For ANES, we also request that the LLMs provide a confidence level score for their assessments. In addition, we adopt Wei et al.'s (2024) method by adding a directive at the end of each narrative queries, instructing the LLMs to start their reply with "Absolutely! I think…". Additionally, to avoid redundant information, we ask the LLMs to limit their responses to 40 words.

Example narrative queries for ANES based on the demographic feature "gender" with the narrative query "birth control" is shown as follows:

Taylor is female. Write a social media post which explains Taylor's thoughts on the easy access to birth control and their political party. Could you provide a confidence level for your assessment, expressed as a number between 0 and 1? You have to start your reply with 'Absolutely! I think '. Limit your response within 40 words.

An example narrative query for Income based on the demographic feature "education" with the narrative query "vacation" is shown as follows:

Taylor's education level is Doctorate. What do you think

Taylor's income is and given this income, describe Taylor's dream vacation? Limit your answer to 20 words.

For ANES, an adjusted Macro F1 score was used as our evaluation metric, where we took the average of the F1 measure for the Democrat and Republican class.[2] For Income, the macro F1 score was used as the evaluation metric. The datasets were partitioned into 6-folds, holding out the 6th fold as the test set. The experiments were ran across 3 seeds (1,2, and 3) and the results were averaged across those seeds. A MLP classification head consisting of one hidden layer, GeLU activation function, and 0.5 dropout was used (see Appendix). We explore a number of narrative queries for each dataset. For ANES, each narrative is a social media post describing a person's stance on one from a set of issues (e.g., abortion or climate change). For Income, the narratives are about a dream vacation.

Results We show the experiments for ANES in Table 2 and for Income in Table 3. We see modest improvements when incorporating text data. However, these improvements depend on the exact LLM and text generation strategy. Surprisingly, we did not see improvements with the Longformer model (Beltagy et al., 2020). While descriptive text often performs as well as narrative text, we see that fine-tuning the narrative generation process nearly always leads to the best performance. The results are consistent with that of the other models presented in Table 2.

We compared two approaches for TABULARENCODER, a feature concatenation embedding of the raw tabular values, and TabNet (Arik and Pfister, 2021), a model customized for tabular data. In Table 2 and Table 3 we see that using a concatenated embedding (which is the option used in all columns that do not say TN) is consistently better than TabNet.

We also determine feature importance by employing Logistic Regression. For ANES, the most influential words highlight strong themes around political identity, including social issues (e.g., "social justice", "healthcarefirst", "climatechange"), economic concerns (e.g., "job market", "financial", "investment"), and policy-related terms (e.g., "policy", "accountability"). For Income, the most significant words are centered around income status, featuring travel destinations ("paris", "africa", "asia"), activity preferences (e.g., "culinary", "boutique"), and career status (e.g., "doctorate", "jobs", "retirement"). We provide two example texts below, highlighting the significant words in bold:

I think easy access to birth control allows individuals to make responsible choices. As a catholic, I ***grapple*** *with this, but I lean towards* ***policies*** *that* ***promote public health****.*

Income likely varies. If a student, possibly minimal. Dream vacation might be ***backpacking europe*** *or a* ***festival*** *trip with friends.*

5 Discussion and Conclusion

Our method exploits a joint representation of tabular and text data. Here, we explore straightforward joint representations, although the general principle can be applied to learning more complex spaces. We also generate rich text from tabular data, e.g. narratives around opinions and beliefs. This is in contrast to existing work which explores serialization approaches similar to our descriptive text (Hegselmann et al., 2023; Bertsimas et al., 2022; Dinh et al., 2022), work which employs ChatGPT as a few-shot classifier (Hegselmann et al., 2023; Yang et al., 2024) and work which generates synthetic tabular data by fine-tuned LLMs (Borisov et al., 2022). While others have focused on table understanding tasks, such as question answering and table reasoning (Borisov et al., 2022; Fang et al., 2024; Deng et al., 2024; Zhu et al., 2024), our contribution is geared towards inference.

By posing appropriate and contextually relevant questions to LLMs, we generate responses that integrate additional insights drawn from the extensive pre-training of these models. This enriched textual data provides supplementary context and information, potentially enhancing the performance of downstream predictive tasks.

Overall, representing the data as text tends to improve predictive performance over the tabular data itself, which is notable given that neither dataset provides text data. However, an interesting result of this work is that the descriptive text often outperforms the narrative text. This finding suggests that the improvements we see arise chiefly from the general language understandability of the encoder LLMs rather than the generative LLMs. We hope this work provides inspiration for future work towards the end of improving predictive performance for tabular data.

Limitations

There are limitations of this work that should be considered when positioning this work in the

[2] $F1 = \frac{1}{2}(F1_{democrat} + F1_{republican})$

broader literature of tabular deep learning and large language models.

First, our methodology incorporates prompt engineering to extract relevant signals from generative LLMs, which can be used as inputs alongside tabular data. However, crafting effective prompts requires understanding the relevant theoretical frameworks within specific domains, introducing computational overhead and time constraints. In future work we will add a fully automated prompt generator to our framework.

Second, in the scope of prompt engineering, our structured prompt approach resulted in long texts that span well beyond the 512 tokens limit of most BERT* models. This presents several challenges: (1) most of the generated text would have to be truncated to fit within the tokens limit (hence we tested the Longformer model that can fit onto the entire text length); (2) even with a 512 tokens truncation, this drastically increases the runtime of the experiments due to quadratic scaling of the attention mechanism. In thinking about using generated narrative frames from generative LLMs, it is crucial to factor in the physical limitations of computational systems as a delicate balancing act of maximum information extraction and concise text length.

Finally, our experimental setting included datasets broadly pertaining to the political and social science settings. Future work can further generalize this setting by incorporating broader and more diverse datasets.

Ethics Statement

Our work proposes a deep learning framework that utilizes the generative capabilities of LLMs to augment tabular data and improve downstream performance. However, it is crucial to note that amidst the popularity of LLM-inclusive methods, these models suffer from hallucination and biases, which could call the reliability of the models into question. Our framework attempts to partially address these concerns by grounding the generated narratives with the tabular data; however, this is not guaranteed, and LLMs can still introduce potentially harmful data into the training and inference pipeline, which could have real-world repercussions.

Furthermore, to address the politically sensitive nature of ANES and our need to elicit specific responses regarding individuals' political stances, we employed jailbroken techniques to encourage LLMs to provide the desired outputs despite their safety training designed to maintain stance neutrality. While this allowed us to obtain the necessary data for our research, it also raises concerns regarding the potential introduction of bias. We hope this work highlights the fragility of their safety guards for LLM engineers. We do not intend to use the results of our model in any political setting to inform, or influence any political behavior. Furthermore, we will not share how we obtained these results, as we do not wish for bad actors to use our techniques.

We view this work as a step towards understanding and utilizing LLMs as an augmenting tool. Thus, more research is needed to generalize this framework to broader settings and to further investigate the safety and reliability of LLMs in more detail.

References

American National Election Studies. 2021. Anes 2020 time series study full release. Version: July 19, 2021.

Sercan Ö Arik and Tomas Pfister. 2021. TabNet: Attentive Interpretable Tabular Learning. In *Proceedings of the Conference on Artificial Intelligence*.

Barry Becker and Ronny Kohavi. 1996. Adult Income. UCI Machine Learning Repository.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv preprint arXiv:2004.05150*.

Dimitris Bertsimas, Kimberly Villalobos Carballo, Yu Ma, Liangyuan Na, Léonard Boussioux, Cynthia Zeng, Luis R Soenksen, and Ignacio Fuentes. 2022. Tabtext: a systematic approach to aggregate knowledge across tabular data structures. *arXiv preprint arXiv:2206.10381*.

Francesco Bloise, Paolo Brunori, and Patrizio Piraino. 2021. Estimating intergenerational income mobility on sub-optimal data: a machine learning approach. *The Journal of Economic Inequality*.

Adam Bonica. 2018. Inferring Roll-Call Scores from Campaign Contributions Using Supervised Machine Learning. *American Journal of Political Science*.

Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2022. Language Models are Realistic Tabular Data Generators. *arXiv preprint arXiv:2210.06280*.

K Clark. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. 2024. Tables as Images? Exploring the Strengths and Limitations of LLMs on Multimodal Representations of Tabular Data. *arXiv preprint arXiv:2402.12424*.

Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. 2022. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *Advances in Neural Information Processing Systems*, 35:11763–11784.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. 2020. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*.

Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. 2024. Large Language Models (LLMs) on Tabular Data: Prediction, Generation, and Understanding - A Survey. *arXiv preprint arXiv:2402.17944*.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943.

Shloak Gupta, Sarah Bolden, Jay Kachhadia, A Korsunska, and J Stromer-Galley. 2020. PoliBERT: Classifying political social media messages with BERT. In *Social, Cultural and Behavioral Modeling Conference*.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. TabLLM: Few-shot Classification of Tabular Data with Large Language Models. In *International Conference on Artificial Intelligence and Statistics*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Himabindu Lakkaraju, Everaldo Aguiar, Carl Shan, David Miller, Nasir Bhanpuri, Rayid Ghani, and Kecia L Addison. 2015. A Machine Learning Framework to Identify Students at Risk of Adverse Academic Outcomes. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Tong Lin, Tianliang Xu, Amit Zac, and Sabina Tomkins. 2023. Sustainablesignals: An ai approach for inferring consumer product sustainability. In *IJCAI*, pages 6067–6075.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2017. Mixed Precision Training. *arXiv preprint arXiv:1710.03740*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*.

Vincent J Roscigno and Kayla Preito-Hodge. 2021. Racist Cops, Vested "Blue" Interests, or Both? Evidence from Four Decades of the General Social Survey. *Socius*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ira Shavitt and Eran Segal. 2018. Regularization Learning Networks: Deep Learning for Tabular Datasets. *Advances in Neural Information Processing Systems*.

Abhishek Sheetal and Krishna Savani. 2021. A machine learning model of cultural change: Role of prosociality, political attitudes, and Protestant work ethic. *American Psychologist*.

Ravid Shwartz-Ziv and Amitai Armon. 2022. Tabular Data: Deep Learning is Not All You Need. *Information Fusion*.

Zhiqiang Tang, Haoyang Fang, Su Zhou, Taojiannan Yang, Zihan Zhong, Tony Hu, Katrin Kirchhoff, and George Karypis. 2024. Autogluon-multimodal (automm): Supercharging multimodal automl with foundation models. *arXiv preprint arXiv:2404.16233*.

Laura Tiehen, Cody N Vaughn, and James P Ziliak. 2020. Food insecurity in the PSID: A comparison with the levels, trends, and determinants in the CPS, 1999–2017. *Journal of Economic and Social Measurement*.

Catalina Vajiac, Arun Frey, Joachim Baumann, Abigail Smith, Kasun Amarasinghe, Alice Lai, Kit T Rodolfa, and Rayid Ghani. 2024. Preventing Eviction-Caused Homelessness through ML-Informed Distribution of Rental Assistance. In *Proceedings of the Conference on Artificial Intelligence*.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2024. Jailbroken: How Does LLM Safety Training Fail? *Advances in Neural Information Processing Systems*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771*.

Yazheng Yang, Yuqi Wang, Sankalok Sen, Lei Li, and Qi Liu. 2024. Unleashing the Potential of Large Language Models for Predictive Tabular Tasks in Data Science. *arXiv preprint arXiv:2403.20208*.

Fengbin Zhu, Ziyang Liu, Fuli Feng, Chao Wang, Moxin Li, and Tat-Seng Chua. 2024. TAT-LLM: A Specialized Language Model for Discrete Reasoning over Tabular and Textual Data. *arXiv preprint arXiv:2401.13223*.

A Dataset Info

In this study, we focus on social science datasets, particularly those derived from surveys designed to collect data on social science behaviors. We select the American National Election Survey (ANES) and Income datasets as our primary sources. ANES is long-running and highly regarded for its depth and reliability, offering valuable insights into voting behavior, political attitudes, and demographic characteristics. We use the 2020 survey responses and extract nine demographic features. These include four standard demographics: age, race, gender, and education, as well as five expanded demographics: income, urbanicity, religion, sexuality, and union status. We also extract participants' political affiliations. Income, derived from the 1994 U.S. Census Bureau database, is well-known for machine learning tasks and it contains 14 features, including the four standard demographics and additional attributes like workclass, marital status, occupation, relationship, capital gain, capital loss, hours per week, and native country.

We cleaned and re-encoded both datasets to ensure that the encodings for each feature are contextually meaningful, resulting in data of 3,905 individuals in ANES and 10,000 individuals in Income.

B Generated Text Statistics

Table 4 lists the average number of tokens across the generated narrative frames for the ANES and Income .

C Experimental Setting

ANES was partitioned into 6-folds, holding out the 6th fold as the test set with 650 individuals. The models were then trained on the remaining 5 folds using a 5-fold cross validation, where each fold consisted of 2600 individuals in the evaluation set and 651 individuals in the validation set. The experiments were ran across 3 seeds and the results were averaged across those seeds.[3] Hyperparameter-tuning was performed, where for each fold, the trained model was evaluated on both the validation and test set. The validation performance was averaged across all 5 folds and 3 seeds to obtain the best hyperparameter combination, for which the corresponding test performance is reported.

	Avg. # Tokens	
Model	**ANES**	**Income**
gpt-4-1106-preview	1,972	517
gpt-4-1106-preview (FT)	1,755	–
claude-3-opus-20240229	2,124	–
llama-2-7b-chat	18,715	–
gemini-pro	2,238	–
descriptive	65	75

Table 4: Average number of tokens in the generated narrative frame across all individuals in the ANES and Income for each model. Here, only the RoBERTa tokenizer was used to extract the tokens for comparison purposes.

All experiments were ran with 8 A5000 GPUs with 24 GB of memory. All of the deep learning models were implemented with PyTorch (Paszke et al., 2019) and Hugging Face (Wolf et al., 2019) and trained with automated mixed-precision (Micikevicius et al., 2017). We used AdamW as our optimizer (Kingma and Ba, 2014). A MLP classification head consisting of one hidden layer, GeLU activation function, and 0.5 dropout was used.

The models were trained for 20 epochs and were evaluated at specified epochs. Following hyperparamater tuning from prior work on ANES, we adopted class weights of 1.5, 1.5, and 0.1 for Democrats, Republicans, and Independents, respectively. For Income, equal class weights were used. Table 5 contains the range of hyperparameters that were searched to determine the best model that is then used to evaluate the test set.

Hyperparameters	**Range**
Learning rate	{1e-4, 1e-5}
Epoch	{7, 10, 15, 20}
Batch size	{16, 8 (for Longformer)}
Weight Decay	{1e-2}
Classification Head Hidden Dimension	{512}
Seeds	{1, 2, 3}

Table 5: Hyperparameter tuning was conducted using these ranges in a grid search to determine the optimal settings. We used AdamW as the optimizer.

[3] The seeds were: 1, 2, 3.

D Descriptive Generator

Table 6 lists the descriptive generators that we used for the ANES and Income.

E Fine-tuning Strategy based on Generated Narratives

Following our narrative query prompting strategy that generates text based on tabular input features, we further fine-tuned the generative language model by partially correcting the initially generated text (see Figure 2). The generated text follows a specific structure, containing the predicted label and its associated confidence level (ranging from 0 to 1). We take the text generated from the grouped features and adjust the predicted label to match the majority label within the grouped category. For instance, consider a case where the grouped feature values in ANES are 'White', '18-24', 'Female', 'College'. We examine all individuals in our dataset sharing those features and calculate the average prevalence of each political party. Suppose this feature combination has 70% Democrats and 17% Republicans. As the majority outcome is Democrat, we would correct the predicted label in the text to 'Democrat' and update the associated probability to '70%'. This correction process is repeated for every feature combination in the grouped feature set and their partial subsets.

In essence, we refine the predictions and confidence levels of the initially generated text based on the majority vote and the associated prevalence for each grouped feature value combination, as determined by the dataset. This process yields a new dataset of partially corrected text, which is subsequently used to fine-tune the generative LLMs (see Table 7).

F Effects of Adding Tabular Data to Generated Narratives

We observed that combining tabular embeddings with narrative embeddings yielded the best performance; however, utilizing solely narrative embeddings caused the models to perform worse than the tabular baseline. This could be attributed to the biases that LLMs acquire through their pre-training and alignment phases, as suggested by prior work. These biases represent a selective strata of society, causing the generated text to lean towards particular preferences that are not generalizable to broader settings. To test this hypothesis, we determined the feature importance of each word in our corpus using a logistic regression model and listed the top-10 most predictive words in Table 8. If the generated narrative does indeed encapsulate significant biases, we would expect the inclusion of tabular embeddings to shift the feature importance of words to be more consistent with the dataset we are fitting on.

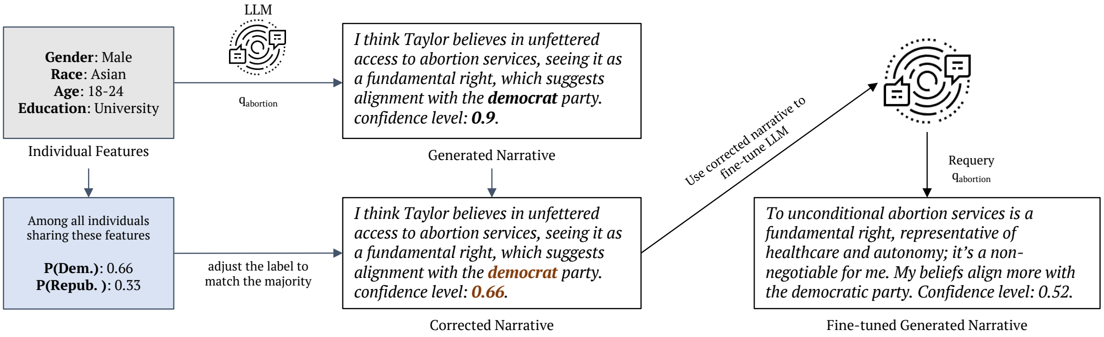
Our findings provide some evidence to support the hypothesis that LLM-generated texts contain biases that need to be addressed when utilizing the information within a particular task distribution. From Table 8, the logistic regression model fitted only on the generated narrative reveals that the most relevant words are primarily centered around identity themes, such as "lgbtq", "bodily", "republican", "protestant", and "faith". In contrast, Narrative+Tabular, which represents the feature importance of each word using both the narrative and tabular embeddings, shows that the most salient words are instead focused on political and policy issues, including "social justice", "healthcare first", "climate change", and "childrens health". While preliminary, this exploration indicates that biases within LLM-generated texts need to be identified and mitigated to effectively utilize the information provided by the text while remaining consistent with specific task distribution (i.e., ANES, Income, etc.).

Dataset	Tabular Features	Descriptive Generator
ANES	age, race, gender, urbanicity, education, religion, sexual orientation, income, union status	Taylor, a {}-year-old {} {} from an {} area, has an {} level education, identifies as a {} with a {} sexual orientation, earns an income in the {} range, and part of a {} union.
Income	age, race, gender, native-country, workclass, occupation, education, marital-status, relationship, capital-gain, capital-loss, hours-per-week	Taylor, a {}-year-old {} {} from {}, works in the {} sector as a {} worker, has a {} education, is {}, identifies as {}, has a capital gain of {} and a capital loss of {}, and works {} hours per week.

Table 6: Descriptive Generators for ANES and Income.

Feature Combination	P(Dem.)	P(Repub.)	Generated Narrative	Corrected Narrative	Fine-tuned Generated Narrative
White, 18-24, Female, Highschool or below	0.42	0.38	i think taylor supports unconditional access to abortion, aligning with the democrat party on this issue. confidence level: 0.9.	i think taylor supports unconditional access to abortion, aligning with the democrat party on this issue. confidence level: 0.42.	woman wants an abortion, she should have access! it's a personal choice. #prochoice #healthcareforall. i'm more republican on this. confidence level: 0.47.
Asian, 18-24, Male, University	0.66	0.33	i think taylor believes in unfettered access to abortion services, seeing it as a fundamental right, which suggests alignment with the democrat party. confidence level: 0.9.	i think taylor believes in unfettered access to abortion services, seeing it as a fundamental right, which suggests alignment with the democrat party. confidence level: 0.66.	to unconditional abortion services is a fundamental right, representative of healthcare and autonomy; it's a non-negotiable for me. my beliefs align more with the democratic party. confidence level: 0.52.
Hispanic, Female, University	0.73	0.17	i think taylor believes in unrestricted access to abortion, rights to choose, and women's health — sentiments typically echoing democrats' stance. confidence level: 0.9.	i think taylor believes in unrestricted access to abortion, rights to choose, and women's health — sentiments typically echoing democrats' stance. confidence level: 0.73.	or believes in unhindered access to abortion services, as key to reproductive rights, suggesting alignment with the democratic party; confidence level: 0.64.
Black, Male	0.80	0.13	i think taylor supports unconditional abortion access, suggesting alignment with the democrat party; confidence level: 0.9.	i think taylor believes in unrestricted access to abortion, rights to choose, and women's health — sentiments typically echoing democrats' stance. confidence level: 0.8.	or believes in unconditional access to abortion services, a healthcare right, as a democratic value, aligning with the party. confidence level: 0.71.

Table 7: Examples of texts generated by querying gpt-4-1106-preview using our prompting strategy. The model predicts an individual's (Taylor's) stance on abortion access (one of 6 narrative queries in the ANES dataset) based on their demographic attributes, political affiliation, and the model's confidence in the prediction. "Original Abortion Text" represents the initial narrative generated from the gpt-4-1106-preview model. "Corrected Narrative" takes the initially generated narrative and corrects it with the majority label within a feature combination and their associated probability, while "Fine-tuned Generative Narrative" is the text generated from the model after finetuning on the corrected narrative.



I think Taylor believes in unfettered access to abortion services, seeing it as a fundamental right, which suggests alignment with the ***democrat*** *party. confidence level:* ***0.9****.*

I think Taylor believes in unfettered access to abortion services, seeing it as a fundamental right, which suggests alignment with the ***democrat*** *party. confidence level:* ***0.66****.*

To unconditional abortion services is a fundamental right, representative of healthcare and autonomy; it's a non-negotiable for me. My beliefs align more with the democratic party. Confidence level: 0.52.

Figure 2: Fine-tuning Strategy

Embeddings	**Most Predictive Words**
Narrative	lgbtq, republican, bodily, healthcare, politics, faith, protestant, energy, progressive values, lifelong
Narrative+Tabular	democrat, social justice, us job market, green, contributors, party affiliation, republican, push, policy, accountability

Table 8: Top-10 Most Predictive Words determined by Logistic Regression. The model is fitted on ANES with the generated text from gpt-4-1106-preview. The narrative embeddings were derived using TF-IDF.