A Grapheme-Aware Hybrid Framework with LLM Integration for Bengali Text-to-Chart Conversion

Anonymous ACL submission

Abstract

In domains like economics, health, and journalism, text embedded with numerical data is common, yet readers often struggle to derive insights. Converting such texts into charts enhances comprehension but is typically labor-intensive and domain-dependent. Despite progress in English, there is no prior dataset for Bengali. To fill this gap, we have introduced BETAR, a BEngali Text-to-chARt dataset comprising 3,519 annotated texts. We also propose BN-GraBERTNet, a grapheme-aware hybrid model that combines BanglaBERT, BiLSTM, and a fully connected layer to identify x-axis and y-axis entities in text. To handle complex numerical reasoning, we selectively employ open-source large language models (LLMs) to simplify sentences when necessary. These simplified sentences are then processed by our sequence tagging model. The primary goal of this work is to develop a lightweight, user-friendly, and cost-free Bengali test-to-chart system that performs competitively. Although we also evaluated open-source, purely LLM-based approaches, our proposed architecture outperformed them, achieving a weighted average F1 score of 0.93 on the test set.

1 Introduction

Texts in research, economics, and journalism often contain rich numerical data, yet extracting insights from them remains challenging due to the sparse and linear nature of numbers in text (Masson et al., 2023a). Prior studies have criticized textual presentations of numerical information (Feliciano et al., 1963; Klein, 2014; Tufte and Graves-Morris, 1983), whereas charts provide a more intuitive and concise representation of data patterns (Van Wijk, 2005), supporting the notion that "one chart is worth ten thousand words" (Larkin and Simon, 1987).

Automatic text-to-chart conversion offers a scalable solution to this challenge, explored in several English-language studies (Luo et al., 2018; Masson et al., 2023b; Dibia and Demiralp, 2019; Narechania et al., 2020; Rashid et al., 2021; Lai et al., 2020; Tian et al., 2024). However, most require structured data inputs (e.g., tables or JSON) or manual field specifications (Tian et al., 2024; Luo et al., 2018; Narechania et al., 2020; Dibia and Demiralp, 2019), making them unsuitable for direct text-to-chart conversion from natural language.

Recent works like Charagraph and ChartifyText (Masson et al., 2023a; Zhang et al., 2024) attempt full-text-based chart creation, but suffer from limited datasets, inability to handle complex reasoning, high computational cost (e.g., GPT-4), and hallucinations. Furthermore, no prior work has addressed this task in Bengali, a low resource language spoken by over 237 million people. Its rich morphology and orthographic complexity degrade the performance of standard sequence labeling models (Pal et al., 2021; Dash, 2015).

To address this, we introduce BETAR, the first Bengali benchmark dataset for text-to-chart conversion, consisting of 3,519 annotated samples. We propose BN-GraBERTNet, a hybrid model that integrates BanglaBERT embeddings with a grapheme-aware GRU encoder and BiLSTM layers. Grapheme-level encoding has shown significant improvements in Bengali NER tasks, with a 9.7-point increase in F1 score compared to word level embedding(Chaudhary et al., 2018).

To handle mathematically complex sentences, we selectively apply open-source LLMs (up to 8B parameters) during inference to simplify inputs. This approach preserves efficiency and avoids full reliance on LLMs, which are resource-intensive and often suboptimal for structured prediction tasks such as sequence labeling (Wang et al., 2023; Keraghel et al., 2024). For instance, transformer-based models like ClinicalBERT have outperformed LLMs such as GPT-3 and ChatGPT in medical NER tasks (Hu et al., 2023).

Our goal is to develop a lightweight, user-

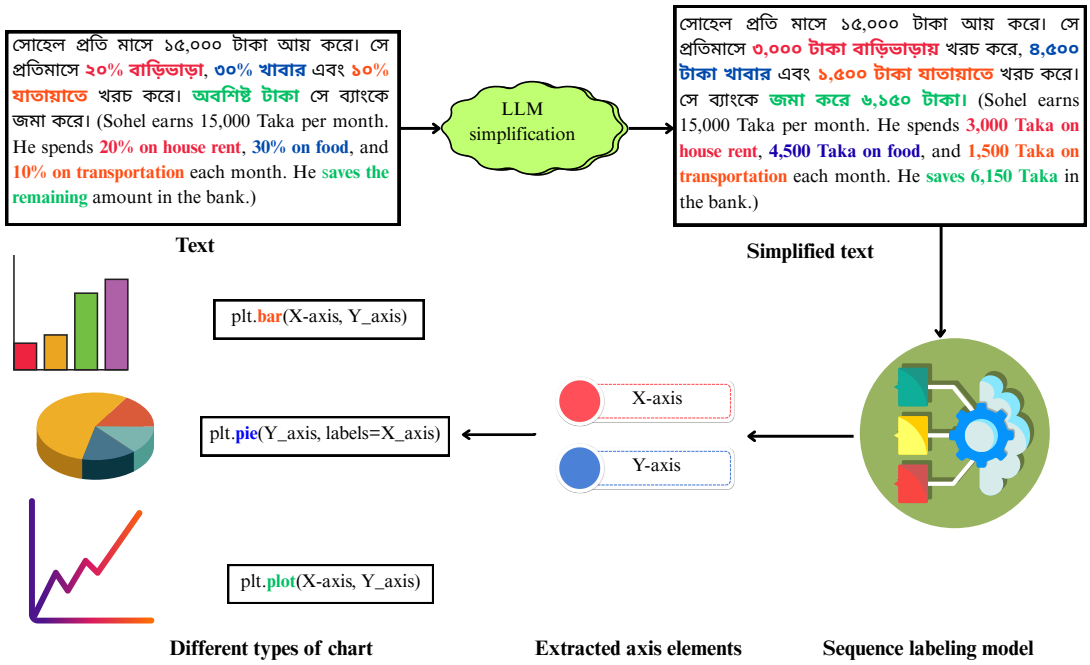

Figure 1: Natural language text to chart creation. Using LLM simplification, 20% house rent becomes 3,000 tk, 30% food cost becomes 4,500 tk, 10% transport becomes 1,500 tk, and the remainder is savings (6,150 tk).

friendly, real-time system for chart generation in Bengali, using free and open-source tools. The system's pipeline is illustrated in Figure 1.

Our main contributions are:

- We present BETAR, a 3,519-sample Bengali dataset for text-to-chart conversion.
- We propose BN-GraBERTNet, a hybrid grapheme-aware model combining BanglaBERT and BiLSTM for axis labeling.
- We selectively use LLMs during inference for simplifying complex sentences, improving downstream axis prediction.
- Our model achieves a weighted F1-score of 0.93, demonstrating strong performance with efficient resource usage.

Our dataset and code will be publicly released upon acceptance.

2 Related Work

Prior research has extensively explored tasks such as chart-to-text generation (Obeid and Hoque, 2020) extensively. In contrast, the reverse task, making charts directly from natural language—remains relatively underexplored.

Most existing text-to-chart systems rely heavily on structured inputs, such as user queries paired with tabular datasets (Luo et al., 2018; Narechania et al., 2020; Tian et al., 2024). Only a few recent works attempt to generate charts from unstructured text. For example, Chargraph, an LLM-based system (Masson et al., 2023a), uses rule-based patterns (e.g., "M=", "%") to detect numerical values. However, when these patterns are inconsistent or absent, the system requires manual user intervention. Additionally, its evaluation is limited to qualitative feedback from 12 users, with no statistical performance metrics reported.

ChartifyText (Zhang et al., 2024), which utilizes GPT-4 without dataset-specific fine-tuning, requires users to manually select relevant portions of text containing chart elements. Despite this manual effort, the system suffers from hallucinations and inconsistent outputs across runs. Moreover, it is not freely accessible and is limited to 15 user reviews in evaluation.

Text2Chart31 (Zadeh et al., 2024) integrates a feedback loop into its LLM-based pipeline and uses GPT-4, which is not open-source or free. Despite this, it still exhibits approximately 6% hallucination (Ford and Rios, 2025), highlighting limitations in semantic reliability.

In contrast, Text2Chart (Rashid et al., 2021) employs a supervised transformer trained on 717 labeled samples, achieving a 97% F1-score across three classes (X-axis, Y-axis, and No-axis). How-

ever, its dataset is relatively simple and consistently includes direct mentions of axes. However, test-to-chart data often contains numerically complex data that require analysis before charting.

Importantly, none of the aforementioned works address complex, multi-step numerical reasoning in the context of chart generation. Moreover, existing works only focus on English, with no benchmark datasets available for low-resource languages such as Bengali. Additionally, most LLM-based systems rely on paid, proprietary models like GPT-4, whereas our open-source architecture demonstrates promising results without the computational and commercial APIs.

3 BETAR: A New Benchmark Dataset

We present BETAR, a dataset designed for automatic Bengali text-to-chart conversion. This section describes the process of creating, annotating, and analyzing the dataset.

3.1 Data Collection

The texts in BETAR were collected from a wide range of Bengali sources, with source and domain distribution detailed in Appendix B. Our goal was to curate a diverse set of data-rich sentences spanning various domains such as news, education, research, health, finance, and more.

In addition, we included a subset of complex sentences requiring multi-step mathematical reasoning, as illustrated in Appendix E. These examples were added only to the test set, as they cannot be labeled through standard supervised methods and instead require manual reasoning by annotators before identifying the x and y spans. Test set construction is illustrated in Appendix F.

3.2 Tokenization and Annotation

Each sentence was first tokenized into custom tokens. These tokens were generated by splitting text based on whitespace and separating out special symbols and punctuation marks. This ensured that both words and symbols were treated as individual, meaningful units.

Three undergraduate students with substantial knowledge of both Bengali and data visualization were tasked with manually annotating the dataset. Each annotator was provided with clear class definitions and asked to assign the index of each custom token to one of the following labels:

- X-axis: Tokens that are typically nouns and represent independent entities such as categories, time points, or names. If a custom token belonged to this category, its index was recorded under the x-axis list in the dataset.
- Y-axis: Tokens indicating numerical values or measurements associated with the X-axis entities. These include quantities, counts, and percentages. If a custom token belonged to this category, its index was recorded under the y-axis list in the dataset.

Punctuation marks and special symbols were also labeled if they occurred as part of an axis-related entity. The full annotation procedure is described in Appendix C. All annotations were subsequently reviewed by an NLP expert with over 10 years of experience, who also resolved any disagreements among annotators. To assess inter-annotator consistency, we have calculated Cohen's Kappa score (Cohen, 1960), which has achieved a mean value of 0.82, indicating substantial agreement among annotators.

3.3 Dataset Statistics

Table 1 presents the custom token distribution across the x-axis, y-axis, and tokens that do not belong to either category, referred to as the no-axis class. The no-axis class is the most frequent, covering approximately 68% of all tokens.

Class	Train	Test Set-1	Total
No-axis	136,143	38,897	175,040
X-axis	34,109	9,735	43,844
Y-axis	29,794	8,503	38,297

Table 1: Number of tokens per class.

We have also analyzed the class overlap using the Jaccard similarity index (Jaccard, 1912). Table 2 shows that x-axis and no-axis tokens share some overlap, with a Jaccard score of 0.1876. This indicates that some tokens can appear in multiple roles depending on context.

Label 1	**Label 2**	**Similarity Score**
No-axis	X-axis	0.1876
No-axis	Y-axis	0.0462
X-axis	Y-axis	0.0325

Table 2: Jaccard Similarity Scores Between Different Category of Tokens.

The average sentence length is 64 tokens. The longest sentence contains 297 tokens.

3.4 Guided Format

To fine-tune LLM for token classification task, we have transformed our dataset into a guided format. Appendix D. Each instance consists of an instruction, the input sentence, and two target lines: one listing the x-axis values and the other the y-axis values. This format helps align the data with instruction-based learning for large language models.

4 Methodology

4.1 Initial Experiments

4.1.1 Large language model (LLM)

We explored both prompting and fine-tuning strategies using large language models (LLMs) for the text-to-chart task. In the zero-shot setting, only task instructions were provided, while few-shot prompting included three labeled examples to guide the model. Prompts used a context window of $n_{\text{ctx}} = 4096$, a max token limit of 512, temperature of 0.1, and top-$p = 0.9$.

For fine-tuning, we trained open-source LLMs (up to 8B parameters) on our guided dataset using LoRA (rank 8, $\alpha = 16$), 4-bit quantization, and AdamW with a learning rate of 1e-5. Generation was capped at 64 tokens with a sequence length of 512.

To ensure the system is resource-efficient, real-time usable, and cost-free for users, we focused exclusively on open-source LLMs with up to 8 billion parameters. Most existing LLMs are not specifically trained for Bengali; rather, they are multilingual and contain only a small fraction of parameters and vocabulary relevant to Bengali. Among the few language-specific models, TituLM 3.2–3B stands out as one of the most promising pure Bengali LLMs, currently available on Hugging Face.

4.1.2 Transformer based models

We have employed three pretrained transformer models which are mBERT, XLM-R, and BanglaBERT. The mBERT, fine-tuned on the Wikiann dataset (Pan et al., 2017). XLM-R (Conneau et al., 2019), with its broader multilingual training and higher capacity, offers stronger cross-lingual representation. BanglaBERT (Bhattacharjee et al., 2021), a monolingual BERT-base model pretrained on 1.5B Bengali tokens, captures rich linguistic features unique to Bengali. We initially fine-tuned the pretrained models for our task, but the performance was relatively modest with batch size=16, learning rate = 0.00001 and Adam optimizer.

4.2 Hybrid Models

We also experimented with hybrid models. In addition to the transformer-based encoder, which leverages self-attention for contextual representation, we incorporate a BiLSTM layer to capture sequential dependencies from both directions—an important aspect for sequence labeling tasks. A fully connected (FC) layer is used to compute logits and produce the final predictions. The models are trained using the Adam optimizer with a learning rate of 0.00001 and a batch size of 16.

4.3 Proposed Architecture

We begin by tokenizing the dataset using a custom tokenizer. Each token is labeled as 1 if it belongs to the X-axis list, 2 if it belongs to the Y-axis list, and 0 otherwise. Contextualized embeddings for each token are then extracted using the BanglaBERT model, which employs multi-head self-attention (Vaswani et al., 2017) to capture semantic dependencies across the sentence. Each input token t_i is mapped to a 768-dimensional embedding $\mathbf{e}_i^{\text{BERT}} \in R^{768}$, encoding rich contextual information.

A key challenge in Bengali is the limited vocabulary coverage of pretrained tokenizers, compounded by the language's complex morphology, inflections, and frequent spelling variations. These factors lead to a high rate of out-of-vocabulary (OOV) tokens during inference, which degrades performance.

To mitigate this, we incorporate a grapheme-aware encoder that decomposes each token t_i into its constituent graphemes—the smallest meaningful units in Bengali. This fine-grained representation reduces the impact of OOV issues and improves generalization to unseen words. Chaudhary et al. (2018) report up to a 9.7-point improvement in Bengali NER performance using grapheme-level encoding.

$$t_i = [g_{i,1}, g_{i,2}, \ldots, g_{i,n}]$$

Each grapheme $g_{i,j}$ is embedded into R^{30}, and the resulting sequence is passed through Bidirectional GRU (BiGRU) with 64 hidden units in each direction, resulting in a 128-dimensional grapheme

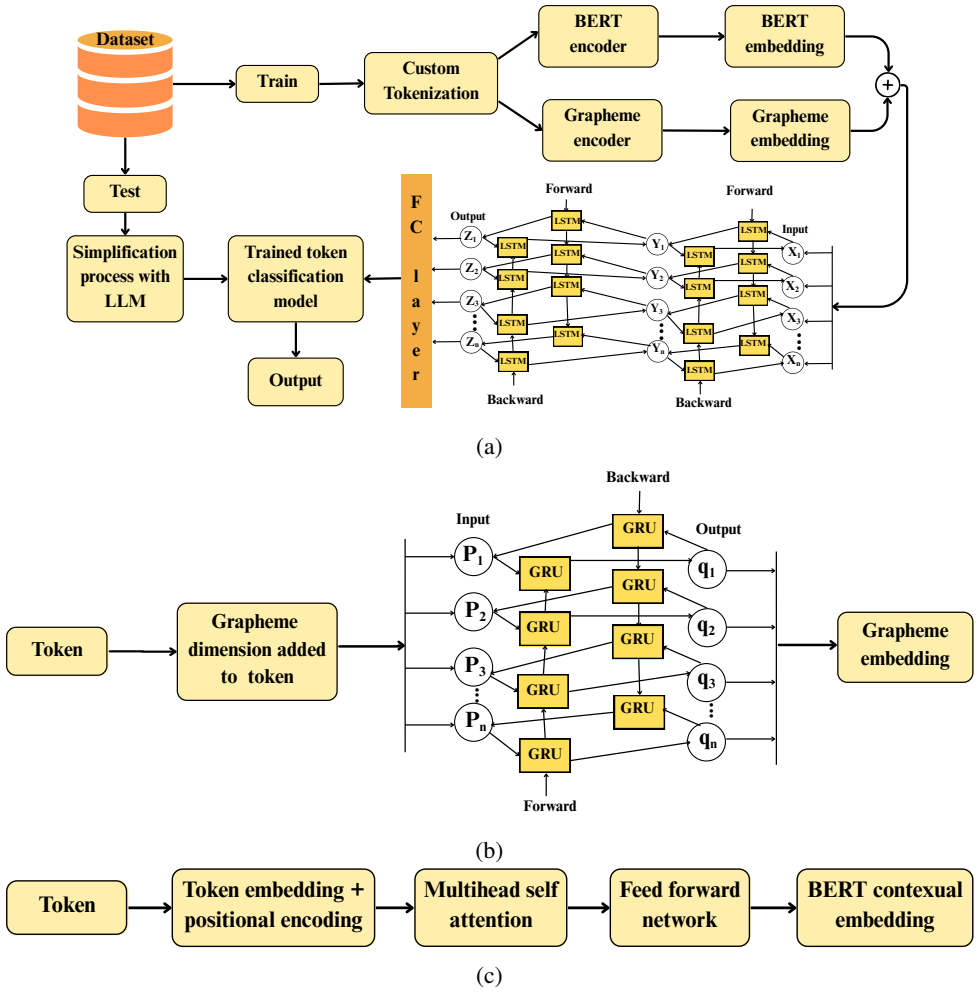

Figure 2: Overview of the (a) BN-GraBERTNet architecture (b) Grapheme encoder (c) BERT encoder

embedding per token. The final grapheme embedding is obtained by concatenating the final hidden states:

$$\mathbf{e}_i^{\text{GR}} = [\overrightarrow{h}_n; \overleftarrow{h}_1] \in R^{128}$$

We then concatenate the BERT embedding and the grapheme embedding to form a fused representation:

$$\mathbf{e}_i = [\mathbf{e}_i^{\text{BERT}}; \mathbf{e}_i^{\text{GR}}] \in R^{768+128=896}$$

This fused representation $\mathbf{e}_i \in R^{896}$ is passed through a two-layer Bidirectional LSTM with hidden size 256 in each direction and dropout rate 0.3. The BiLSTM outputs a hidden state:

$$\mathbf{h}_i \in R^{256} \quad \text{(i.e., } 2 \times 256 \text{ from BiLSTM)}$$

Finally, a fully connected (FC) layer maps the BiLSTM output for each token to a 3-dimensional logit vector corresponding to class label $y_i \in \{0, 1, 2\}$:

$$\hat{y}_i = \text{softmax}(\mathbf{W}_o \cdot \mathbf{h}_i + \mathbf{b}_o), \quad \hat{y}_i \in R^3$$

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{k=1}^{3} \exp(z_k)}$$

The FC layer acts as a token-level classifier that projects the 512-dimensional context vector into class scores. The overall architecture of our methodology illustrated in Figure 2. After generating the 0, 1, and 2 labels for each token, we grouped consecutive tokens labeled as 1 into segments referred to as x-axis, and consecutive tokens labeled as 2 into y-axis.

4.4 Sentence Simplification Using LLMs

During testing, sentences were first simplified when necessary. We employed three instruction-tuned models: "Meta-Llama-3-8B", "Mistral-7B-Instruct" and "TituLM-Llama-3.2-3B-v2.0" within a few-shot learning setup consisting of three examples. The simplification pipeline operates in two stages. First, the model classifies each Bangla sentence as either simple or complex using a few-shot

prompt. Only complex sentences are passed to a second prompt for simplification, which preserves numerical values, avoids hallucination, and clarifies multi-step reasoning. All generations are configured with temperature 0.1, top_p 0.9 and a 512 token limit. This strategy improves clarity while minimizing unnecessary processing.

5 BETAR: Benchmark Evaluation

5.1 Evaluation Metrics

We have employed token level evaluation for trasformer and hybrid models. To compare performance with large language model we have also evaluated span level evaluation. For token classification, we computed the weighted F1-score across three token classes: X-axis (1), Y-axis (2), and No-axis (0). For span-based evaluation, we converted consecutive tokens labeled as 1 into X-spans, and consecutive tokens labeled as 2 into Y-spans. For LLM-based generative models, we adopted a relaxed string-matching strategy to evaluate span level axis predictions. We have used the RapidFuzz library to compute fuzzy string similarity between predicted and gold spans. A prediction was considered a correct match if its similarity score with a gold span was greater than or equal to 90%.

$$\text{match}(p_i, g_j) = \begin{cases} 1, & \text{if Similarity}(p_i, g_j) \geq 90\% \\ 0, & \text{otherwise} \end{cases}$$

Using this relaxed alignment, we calculated relaxed precision, recall, and F1-score to account for minor variation and rephrasing that commonly arise in generative model outputs.

We have also evaluated the simplification quality using a Rough Score, which quantifies how well the model-generated simplified sentence aligns with a manually created reference.

The Rough Score is defined as:

$$\text{RoughScore} = \frac{1}{|S_{\text{ref}}|} \sum_{w \in S_{\text{ref}}} 1 \left[\max_{v \in S_{\text{pred}}} \text{sim}(w, v) \geq \tau \right] \times 100$$

where:

- S_{ref} is the set of tokens in the reference simplification,
- S_{pred} is the set of tokens in the model-generated simplification,
- $\text{sim}(w, v) \in [0, 1]$ is a fuzzy similarity score between words w and v,
- τ is a similarity threshold (we use $\tau = 0.90$).

This metric captures semantic overlap rather than surface-level exactness, providing a coarse but effective approximation of sentence-level correctness during simplification.

5.2 Results

We conducted extensive experiments on our dataset using transformer-based, hybrid, and LLM-driven approaches. Among all models, our proposed hybrid architecture, BN-GraBERTNet, achieved the highest performance, attaining a weighted F1-score of 0.98 for token classification and 0.93 for span-level evaluation. The span-level evaluation focused exclusively on X and Y axis elements, excluding tokens labeled as non-axis (i.e., label 0), which are only considered during token-level evaluation. Since large language models are not well-suited for fine-grained token-level prediction due to their tendency toward overgeneralization, we adopted span-based evaluation with relaxed matching criteria for assessing LLM performance. In particular, few-shot simplification using Meta-LLaMA-3-8B (GGUF) achieved an approximate simplification accuracy of 91.3

Table 3 and Table 5 present the weighted F1-scores for token- and span-level classification, respectively. Table 6 highlights the simplification errors introduced by LLMs, while Table 4 compares the performance of our model with fine-tuned Meta-LLaMA-3-8B in a pipeline combining LLM-based simplification with axis extraction.

To evaluate computational efficiency, we report floating point operations per second (FLOPS), multiply–accumulate operations (MACs), and GPU memory consumption across all models in Appendix A. Our objective is to design a lightweight, high-performance Bengali NER model suitable for deployment on low-resource consumer hardware—a capability that remains difficult to achieve with current large-scale LLMs.

5.3 Error Analysis

X-axis and no-axis labels predominantly contain textual data, making them difficult for the model to distinguish. Long X-axis spans sometimes cause partial misclassification as no-axis. Additionally, when the X-axis contains numerical values (e.g., years), it is occasionally misclassified as Y-axis. Similarly, Y-axis elements expressed in textual numeric form (e.g., “two thousands”) are sometimes

Method	X-axis			Y-axis			No-axis			
	P	R	WF	P	R	WF	P	R	WF	WF
BanglaBERT	0.89	0.87	0.88	0.87	0.86	0.87	0.86	0.86	0.86	0.88
XLM-Roberta	0.85	0.87	0.86	0.87	0.83	0.85	0.83	0.83	0.83	0.84
mBERT	0.86	0.89	0.87	0.85	0.82	0.83	0.82	0.82	0.82	0.83
XLM-Roberta+BiLSTM+FC	0.89	0.87	0.88	0.87	0.86	0.86	0.86	0.86	0.86	0.91
BanglaBERT+BiLSTM+FC	0.91	0.90	0.91	0.90	0.89	0.90	0.89	0.89	0.89	0.93
XLM-Roberta+Grapheme+BiLSTM+FC	0.94	0.91	0.92	0.97	0.98	0.98	0.98	0.98	0.98	0.96
BanglaBERT+Grapheme+BiLSTM+FC	**0.94**	**0.91**	**0.92**	**0.97**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**
BanglaBERT+Grapheme+BiLSTM+FC	0.90	0.88	0.89	0.95	0.96	0.96	0.96	0.96	0.95	0.95

Table 3: Token level evaluation for transformer and hybrid models.

Text	Simplified text using LLM	Token identification and pair extraction
ঢাকা থেকে অস্ট্রেলিয়া যেতে ভ্রমণকারীদের ৩টি বিমানবন্দরে বিরতি দিতে হয়। ঢাকা থেকে কানাডা যেতে তাদের ২টি বিমানবন্দরে বিরতি দিতে হয়, এবং ঢাকা থেকে যুক্তরাষ্ট্র যেতে তাদের ৪টি বিমানবন্দরে বিরতি দিতে হয়। (***Travelers from Dhaka to Australia have to stop at 3 airports. To go from Dhaka to Canada, they have to stop at 2 airports, and to go from Dhaka to the United States, they have to stop at 4 airports.***) **Gold pairs** অস্ট্রেলিয়া → ৩টি, কানাডা →২টি , যুক্তরাষ্ট্র →৪টি (Australia → 3, Canada → 2, USA → 4)	Already simplified, no further modification required	**LLM (Meta Llama 3-8B finetuned)** ঢাকা থেকে কানাডা → ৩টি, ঢাকা থেকে যুক্তরাষ্ট্র → ২টি ***(Dhaka to Canada → 3, Dhaka to USA → 2)*** **Predicts two X-axis where three X axis values are present. Corresponding Y-axis values are also incorrect.** **BN-GraBERTNet** অস্ট্রেলিয়া → ৩টি, কানাডা → ২টি , যুক্তরাষ্ট্র → ৪টি ***(Australia → 3, Canada → 2, USA → 4)***
একটি শহরে ২০১৯ সালে জনসংখ্যা ছিল ৫০,০০০। ২০২০ সালে এটি ১০% বৃদ্ধি পায়, কিন্তু ২০২১ সালে ৫% কমে যায়। (***In a city, the population was 50,000 in 2019. In 2020, it increased by 10%, but in 2021, it decreased by 5%***) **Gold pairs** ২০১৯ → ৫০,০০০ জন, ২০২০ → ৫৫,০০০ জন, ২০২১ → ৫২,২৫০ জন ***(2019 → 50,000 people, 2020 → 55,000 people, 2021 → 52,250 people)***	**Complex sentence found, simplifying....** একটি শহরে ২০১৯ সালে জনসংখ্যা ছিল ৫০,০০০। ২০২০ সালে জনসংখ্যা হয়েছে ৫৫,০০০। ২০২১ সালে জনসংখ্যা হয়েছে **৫২,৫০০**। (***A city's population was 50,000 in 2019. It increased to 55,000 in 2020, then decreased to 52,500 in 2021.)*** **The simplification wrongly predict 52,500 which should be 52,250**	**LLM (Meta Llama 3-8B finetuned)** ২০১৯ → ৫০, ২০২০ → ০০০, ২০২১ → ৫৫ ***(2019 → 50 people, 2020 → 000 people, 2021 → 55 people)*** **Numerical values are fabricated also not all X axis are captured** **BN-GraBERTNet** ২০১৯ → ৫০,০০০ জন, ২০২০ → ৫৫,০০০ জন, ২০২১ → ৫২,৫০০ জন ***(2019 → 50,000 people, 2020 → 55,000 people, 2021 → 52,500 people)*** **Wrong generated 52,500 which should be 52,250 from simplification process**

Table 4: Complex and simple texts were processed through both our proposed architecture and fine-tuned LLMs. The comparison highlights that, while fine-tuned LLMs achieve strong performance, they exhibit certain degrees of hallucination during generation. In contrast, our hybrid transformer-based model demonstrates more direct output, effectively mitigating such issues.

labeled as no-axis. These misclassifications are illustrated in the confusion matrix in Figure 3. During simplification, the LLM occasionally misinterpreted cases requiring long-term hierarchical reasoning, leading to incorrect outputs for complex sentences.

6 Conclusion

n this work, we introduced a new benchmark dataset comprising 3,519 Bengali texts for the task of chart element extraction. As baselines, we evaluated three pretrained transformer-based models: BanglaBERT, XLM-R, and mBERT. However, our proposed hybrid model, BN-GraBERTNet, achieved the best performance. This model integrates Bengali grapheme-level features with contextual embeddings from BanglaBERT and feeds the combined representation into a two-layer BiLSTM for sequential modeling.

BN-GraBERTNet achieved a token-level weighted F1-score of 0.98 on the test set and the highest span-level weighted F1-score of 0.93. Error analysis revealed that class imbalance occasionally led the model to misclassify x-axis or y-axis tokens as no-axis.

6.1 Limitations

First, the system assumes a linear one-to-one mapping between x and y-axis elements. Specifically, it pairs the first x value with the first y value. This assumption breaks down in cases where a single x corresponds to multiple y values, which ideally

Method	X-axis			Y-axis			
	P	R	WF	P	R	WF	WF
mBERT+Grapheme+BiLSTM+FC	0.900	0.880	0.890	0.930	0.945	0.940	0.915
XLM-Roberta+Grapheme+BiLSTM+FC	0.940	0.910	0.910	0.938	0.952	0.946	0.927
BanglaBERT+Grapheme+BiLSTM+FC	**0.904**	**0.913**	**0.909**	**0.944**	**0.965**	**0.954**	**0.934**
Meta-LLaMA-3-8B(FT)	0.880	0.874	0.877	0.873	0.862	0.868	0.838
Gemma-3-4B(FT)	0.850	0.861	0.856	0.868	0.834	0.849	0.812
TituLM-3.2-3B(FT)	0.865	0.854	0.859	0.861	0.838	0.847	0.825
Phi-4-mini-Instruct(FT)	0.842	0.829	0.835	0.838	0.821	0.824	0.798
Qwen 1.5B-Instruct(FT)	0.839	0.826	0.832	0.834	0.818	0.820	0.795
BLOOMZ-1b7(FT)	0.838	0.825	0.831	0.834	0.819	0.821	0.794
Meta-LLaMA-3-8B (zero shot)	0.800	0.790	0.795	0.792	0.781	0.783	0.752
TituLM-3.2-3B (zero shot)	0.811	0.801	0.806	0.794	0.785	0.787	0.760
Meta-LLaMA-3-8B (few shot)	0.872	0.864	0.868	0.861	0.852	0.858	0.827
TituLM-3.2-3B (few shot)	0.842	0.831	0.836	0.835	0.824	0.826	0.794

Table 5: Span-level evaluation across hybrid and large language models. We evaluated the top three models from both the transformer-based and hybrid model series based on their token classification performance.

Model	Rough score
TituLM-Llama-3.2-3B-v2.0	78.2%
Mistral-7B-Instruct	86.1%
Meta-LLaMA-3-8B	**91.3%**

Table 6: LLM simplification time rough score.

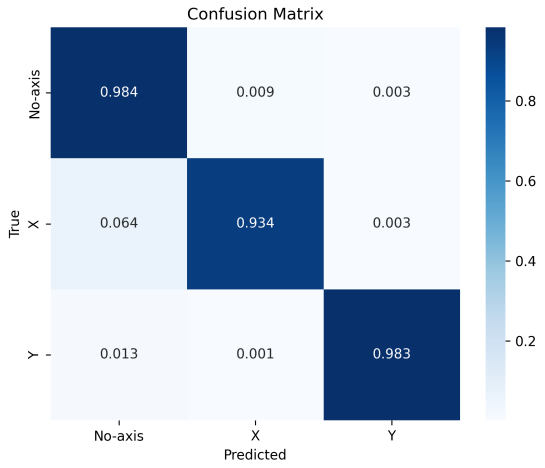

Figure 3: Confusion matrix for token label classification of our proposed model.

require stacked bar charts, functionality not yet supported by the system. Additionally, the model fails to handle non-linear relationships in the text, where the order of x and y elements does not align sequentially.

Second, it struggles with numerically ambiguous or structurally complex inputs. For instance, in phrases like "5–7 million people," the model identifies components but fails to generate a representative value suitable for plotting.

Third, when the input covers multiple distinct topics, the system produces only one chart, rather than decomposing the input into multiple subcharts.

Fourth, while the LLM-based simplifier generates fluent outputs, it occasionally misinterprets or hallucinates information, affecting chart element extraction.

Fifth, the system currently supports only monolingual Bengali. Extending it to multilingual or code-mixed inputs is a promising direction.

Future work will explore learning-based axis alignment, inclusion of multilingual data, improved hallucination mitigation, and the use of larger, more capable language models beyond the open-source options used in this study.

7 Ethical Considerations

We constructed the BETAR dataset in accordance with ethical and fair data usage guidelines. All texts were sourced from publicly available Bengali content, including social media, websites, educational materials, and news articles. No personally identifiable information (PII) was collected, and no data came from private or restricted sources. All sentences were anonymized and factual. Annotators followed clear guidelines to ensure consistent labeling and minimize bias. The dataset covers diverse domains to maintain balance and representativeness.

References

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Kazi Samin, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2021. Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. *arXiv preprint arXiv:2101.00204*.

Aditi Chaudhary, Chunting Zhou, Lori Levin, Graham Neubig, David R Mortensen, and Jaime G Carbonell. 2018. Adapting word embeddings to new languages with morphological and phonological subword representations. *arXiv preprint arXiv:1808.09500*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Niladri Sekhar Dash. 2015. Frequency of use of words in bengali.

Victor Dibia and Çağatay Demiralp. 2019. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE computer graphics and applications*, 39(5):33–46.

Gloria D Feliciano, Richard D Powers, and Bryant E Kearl. 1963. The presentation of statistical information. *Audio Visual communication review*, 11(3):32–39.

James Ford and Anthony Rios. 2025. Does it run and is that enough? revisiting text-to-chart generation with a multi-agent approach. *arXiv preprint arXiv:2506.06175*.

Yan Hu, Iqra Ameer, Xu Zuo, Xueqing Peng, Yujia Zhou, Zehan Li, Yiming Li, Jianfu Li, Xiaoqian Jiang, and Hua Xu. 2023. Zero-shot clinical entity recognition using chatgpt. *arXiv preprint arXiv:2303.16416*.

Paul Jaccard. 1912. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50.

Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. 2024. A survey on recent advances in named entity recognition. *arXiv preprint arXiv:2401.10825*.

Torsten L Klein. 2014. Communicating quantitative information: tables vs graphs.

Chufan Lai, Zhixian Lin, Ruike Jiang, Yun Han, Can Liu, and Xiaoru Yuan. 2020. Automatic annotation synchronizing with textual description for visualization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Jill H Larkin and Herbert A Simon. 1987. Why a diagram is (sometimes) worth ten thousand words. *Cognitive science*, 11(1):65–100.

Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. Deepeye: Towards automatic data visualization. In *2018 IEEE 34th international conference on data engineering (ICDE)*, pages 101–112. IEEE.

Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. 2023a. Charagraph: Interactive generation of charts for realtime annotation of data-rich paragraphs. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–18.

Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. 2023b. Statslator: Interactive translation of nhst and estimation statistics reporting styles in scientific documents. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14.

Arpit Narechania, Arjun Srinivasan, and John Stasko. 2020. Nl4dv: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):369–379.

Jason Obeid and Enamul Hoque. 2020. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. *arXiv preprint arXiv:2010.09142*.

Alok Ranjan Pal, Diganta Saha, Sudip Kumar Naskar, and Niladri Sekhar Dash. 2021. In search of a suitable method for disambiguation of word senses in bengali. *International Journal of Speech Technology*, 24:439–454.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1946–1958.

Md Mahinur Rashid, Hasin Kawsar Jahan, Annysha Huzzat, Riyasaat Ahmed Rahul, Tamim Bin Zakir, Farhana Meem, Md Saddam Hossain Mukta, and Swakkhar Shatabda. 2021. Text2chart: A multi-staged chart generator from natural. *arXiv preprint arXiv:2104.04584*.

Yuan Tian, Weiwei Cui, Dazhen Deng, Xinjing Yi, Yurun Yang, Haidong Zhang, and Yingcai Wu. 2024. Chartgpt: Leveraging llms to generate charts from abstract natural language. *IEEE Transactions on Visualization and Computer Graphics*.

Edward R Tufte and Peter R Graves-Morris. 1983. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT.

Jarke J Van Wijk. 2005. The value of visualization. In *VIS 05. IEEE Visualization, 2005.*, pages 79–86. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Fatemeh Pesaran Zadeh, Juyeon Kim, Jin-Hwa Kim, and Gunhee Kim. 2024. Text2chart31: Instruction tuning for chart generation with automatic feedback. *arXiv preprint arXiv:2410.04064*.

Songheng Zhang, Lei Wang, Toby Jia-Jun Li, Qiaomu Shen, Yixin Cao, and Yong Wang. 2024. Chartifytext: Automated chart generation from data-involved texts via llm. *arXiv preprint arXiv:2410.14331*.

Appendix

A Comparison of Model Resource Consumption

LLM	Sequence Length	FLOPS (GFLOPS)	MACs (GMACS)	GPU Memory (MB)
LLAMA-3 (8B)	128	1,920	960.6	32,630.63
	256	3,840	1,920	34,260.63
	512	7,690	3,840	38,032.63
	1024	15,370	7,680	44,080.63
	2048	–	–	>48,000
XLM-RoBERTa-large (559.89M)	128	78.96	39.46	2,688.63
	256	161.15	80.53	2,962.63
	512	335.24	167.5	3,640.63
BanglaBERT (110.03M)	128	22.36	11.17	842.63
	256	45.94	22.95	1,012.63
	512	96.72	45.91	1,432.63

Table 7: Comparison of FLOPS, MACs, and GPU memory usage across models for different sequence lengths.

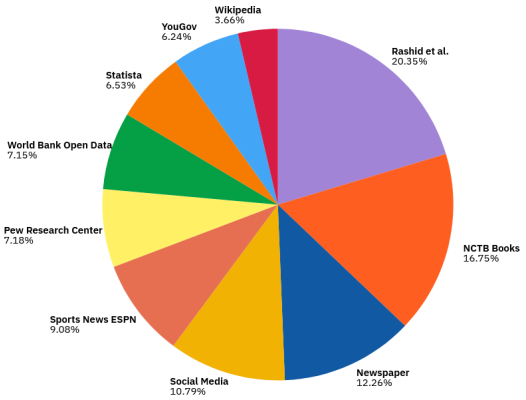

Figure 4: Illustration of our proposed dataset's source distribution.

B Dataset Statistics

We curated the dataset from a mix of Bengali and English sources, with English texts translated into Bengali using a blend of manual annotation and Google Translate. Most examples are numerically rich and designed for chart element extraction. To ensure generalization, we also included non-numeric sentences without identifiable X- or Y-axis elements.

The test set contains 519 examples, including 212 complex and the rest simple. In total, the dataset spans 10 sources and 10 domains, as shown in Figure 4 and Figure 5.

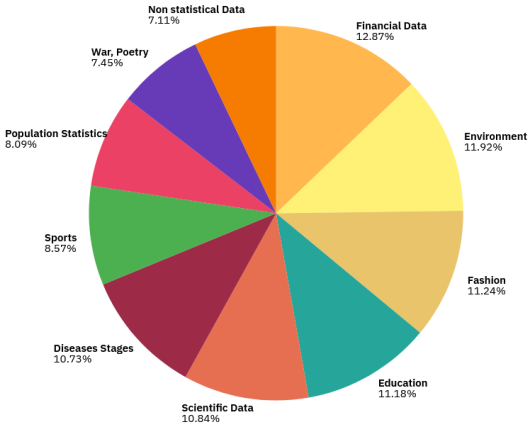

Figure 5: Illustration of our proposed dataset's domain distribution.

C Annotation Process

The annotation procedure is summarized in Table 8. Annotators were provided with a Python script that tokenizes each sentence into whitespace-separated units, including punctuation and special symbols, and assigns indices to each token. They then reviewed the indexed tokens and manually marked those corresponding to X-axis and Y-axis elements. The selected indices were recorded in separate X-axis and Y-axis lists.

D Guided dataset format.

The guided dataset format is illustrated in Table 9, showing an example sentence alongside its corresponding guided version. Unlike the original format, which provides index spans for X-axis and Y-axis elements, the guided format replaces index with the actual entity values. This structure is tailored for fine-tuning large language models (LLMs), offering explicit supervision through natural language instructions and structured outputs.

Text	Text with Indexed Token	Label
বিপিএল মিউজিক ফেস্টের জন্য এরই মধ্যে অনলাইনে টিকিট ছাড়া হয়েছে। তবে দর্শকদের তেমন সাড়া না থাকায় টিকিটের মূল্য কমানোর সিদ্ধান্ত নিয়েছে বিসিবি। প্রথমে প্লাটিনাম টিকিটের মূল্য ১২ হাজার টাকা। গোল্ড, সিলভার, গ্র্যান্ড স্ট্যান্ড যথাক্রমে ৮ হাজার টাকা, ৬ হাজার টাকা, ৪ হাজার টাকা। ক্লাব হাউজ টিকিটের মূল্য নির্ধারণ করা হয়েছিল আনুমানিক ২ হাজার ৫০০ টাকা। [***translate***: *Tickets for the BPL Music Fest have already been released online. However, due to a lack of significant response from the audience, the BCB has decided to reduce the ticket prices. Initially, the price of a Platinum ticket was set at 12,000 BDT. The prices for Gold, Silver, and Grand Stand tickets were 8,000 BDT, 6,000 BDT, and 4,000 BDT respectively. The Club House ticket was approximately priced at 2,500 BDT.*]	বিপিএল(0) মিউজিক(1) ফেস্টের(2) জন্য(3) এরই(4) মধ্যে(5) অনলাইনে(6) টিকিট(7) ছাড়া(8) হয়েছে(9)।(10) তবে(11) দর্শকদের(12) তেমন(13) সাড়া(14) না(15) থাকায়(16) টিকিটের (17) মূল্য(18) কমানোর(19) সিদ্ধান্ত(20) নিয়েছে(21) বিসিবি(22)।(23) প্রথমে(24) প্লাটিনাম(25) টিকিটের(26) মূল্য(27) ১২(28) হাজার(29) টাকা(30)।(31) গোল্ড(32),(33) সিলভার(34),(35) গ্র্যান্ড(36) স্ট্যান্ড(37) যথাক্রমে(38) ৮(39) হাজার(40) টাকা(41),(42) ৬(43) হাজার(44) টাকা(45),(46) ৪(47) হাজার(48) টাকা(49)।(50) ক্লাব(51) হাউজ(52) টিকিটের(53) মূল্য(54) নির্ধারণ(55) করা(56) হয়েছিল(57) আনুমানিক(58) ২(59) হাজার(60) ৫০০(61) টাকা(62)।(63) [*BPL(0) Music(1) Fest(2) tickets(3) have(4) already(5) been(6) released(7) online(8).(9) However(10),(11) due(12) to(13) the(14) lack(15) of(16) response(17) from(18) the(19) audience(20),(21) BCB(22) has(23) decided(24) to(25) reduce(26) ticket(27) prices(28).(29) Initially(30),(31) the(32) price(33) of(34) Platinum(35) tickets(36) was(37) 12(38) thousand(39) taka(40).(41) Gold(42),(43) Silver(44),(45) and(46) Grand(47) Stand(48) tickets(49) were(50) priced(51) at(52) 8(53) thousand(54) taka(55),(56) 6(57) thousand(58) taka(59),(60) and(61) 4(62) thousand(63) taka(64).(65) The(66) Club(67) House(68) ticket(69) price(70) was(71) set(72) at(73) approximately(74) 2(75) thousand(76) 500(77) taka(78).(79)*	**X-axis:** {25, 32, 34, 36-37, 51-52} **Y-axis** : {28-30, 39-41, 43-45, 47-49, 59-62} **Label for translated text** **X-axis:** {35,42,44,47-48, 66-68} **Y-axis:** {38-40, 53-55, 57-59. 62-64, 75-78}

Table 8: Annotation procedure.

Normal dataset	Guided dataset

{
text: "ব্ল্যাক ফ্রাইডের সময়, সারা বিশ্বে হার্ডওয়্যার খরচ এক বা দুই দিনের জন্য অত্যন্ত হ্রাস পায়। গত বছর, টার্গেট, ইবি গেমস, ওয়ালমার্ট - এই ৩টি খুচরা বিক্রেতা তাদের সমস্ত পণ্যে ৬৫% ছাড় দিয়েছে, যেখানে অন্য ১২টি দোকান তাদের পণ্যে মাত্র ২৫% ছাড় দিয়েছে।",
 	
(***translation:*** *During Black Friday, hardware costs across the world drop significantly for one or two days. Last year, three major retailers — Target, EB Games, and Walmart — offered a 65% discount on all their products, whereas twelve other stores provided only a 25% discount on their items.)*
 	
X-axis: {20-25, 41-42}
Y-axis: {34-35, 47-48}
}
	কাজ: প্রদত্ত বাংলা বাক্য থেকে চার্টের উপাদানগুলি বের করো।
	(***translation:*** *Task: Extract chart components from the given Bengali sentence.)*
	
	ইনপুট: ব্ল্যাক ফ্রাইডের সময়, সারা বিশ্বে হার্ডওয়্যার খরচ এক বা দুই দিনের জন্য অত্যন্ত হ্রাস পায়। গত বছর, টার্গেট, ইবি গেমস এবং ওয়ালমার্ট - এই ৩টি খুচরা বিক্রেতা তাদের সমস্ত পণ্যে ৬৫% ছাড় দিয়েছে, যেখানে অন্য ১২টি দোকান তাদের পণ্যে মাত্র ২৫% ছাড় দিয়েছে।
	(***translation:*** *Input: During Black Friday, hardware costs across the world drop significantly for one or two days. Last year, Target, EB Games, and Walmart — these three retailers offered a 65% discount on all their products, while another 12 stores offered only a 25% discount.)*
	
	আউটপুট:
	এক্স-অক্ষ: টার্গেট, ইবি গেমস এবং ওয়ালমার্ট, অন্য ১২টি দোকান
	ওয়াই-অক্ষ: ৬৫ %, ২৫ %
	
	(***translation: Output:***
	X-axis: Target, EB Games, Walmart, Other 12 stores
	Y-axis: 65%, 25%)

Table 9: Guided dataset construction. An instruction is provided to help the LLM understand its task. The input sentence is presented as-is, while span labels are converted into actual X-axis and Y-axis values.

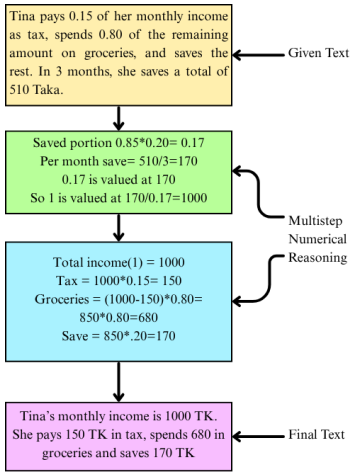

Figure 6: Multistep reasoning process for a complex example. In the dataset, for sentence simplification, the LLM is provided with the first and last sentences as examples in a few-shot setting. The intermediate calculation steps are intentionally omitted from the prompt.

E Multistep complex numerical reasoning of data

Our dataset contains 212 complex samples in the test set, out of a total of 519. These samples require simplification before being passed to the axis labeling model. In many cases, multi-step reasoning is necessary to interpret the data correctly. An illustration of this type of reasoning is shown in Figure 6.

F Test dataset annotation

Annotators did not mark individual token indices, as this was only required during the training phase for token classification. Since no token-level learning was involved during testing, they directly identified the X-axis and Y-axis spans. For complex cases, annotators first simplified the sentence by performing necessary mathematical reasoning, and then determined the corresponding X and Y spans.

Text	Simplified Text	Label
উত্তর আমেরিকায় এই বছর প্রচণ্ড গরম পড়েছে, যা গত দশকের তুলনায় ২.৫ ডিগ্রি সেলসিয়াস বেশি। গত দশকে তাপমাত্রা ছিল ৩৫ ডিগ্রি সেলসিয়াস। *(translation: This year, North America experienced extreme heat, which is 2.5°C higher compared to the last decade. The average temperature during the last decade was 35°C.)*	উত্তর আমেরিকায় এই বছর প্রচণ্ড গরম পড়েছে। গত দশকে তাপমাত্রা ছিল ৩৫ ডিগ্রি সেলসিয়াস। এই বছর তাপমাত্রা হয়েছে ৩৭.৫ ডিগ্রি সেলসিয়াস। *(translation: North America experienced extreme heat this year. The temperature during the last decade was 35°C. This year, the temperature reached 37.5°C.)* ***Simplified by annotator***	X-axis: গত দশকে, এই বছর Y-axis: ৩৫ ডিগ্রি সেলসিয়াস, ৩৭.৫ ডিগ্রি সেলসিয়াস (translation: X-axis: Last decade, This year Y-axis: 35°C, 37.5°C)
১৯০৫ সালে আরএফএল ব্যবসায় ৫০% লাভের মাধ্যমে সফলতা অর্জন করে। পরবর্তী তিন বছর ধরে তারা প্রতি বছর পূর্ববর্তী বছরের তুলনায় ৫% হারে লাভ বৃদ্ধি করে। *(translation: In 1905, RFL achieved business success with a 50% profit. Over the next three years, they increased their profit by 5% each year compared to the previous year.)*	১৯০৫ সালে আরএফএল ৫০% লাভ করে। পরের বছরগুলোতে প্রতি বছর আগের বছরের তুলনায় ৫% করে লাভ বাড়ে। ফলে ১৯০৬ সালে ৫২.৫%, ১৯০৭ সালে ৫৫.১২%, এবং ১৯০৮ সালে প্রায় ৫৭.৮৮% লাভ করে। *(translation: RFL made a 50% profit in 1905. In the following years, the profit increased by 5% over the previous year. As a result, the profit was 52.5% in 1906, 55.12% in 1907, and approximately 57.88% in 1908.)* ***Simplified by annotator***	X-axis: ১৯০৫, ১৯০৬, ১৯০৭, ১৯০৮ Y-axis: ৫০%, ৫২.৫%, ৫৫.১২%, ৫৭.৮৮% (translation: X-axis: 1905, 1906, 1907, 1908 Y-axis: 50%, 52.5%, 55.12%, 57.88%)
তালিকাভুক্ত বহুজাতিক কোম্পানির আর্থিক রিপোর্টে দেখা গেছে যে, জুলাই-সেপ্টেম্বর ত্রৈমাসিকে বিক্রয় রাজস্ব ১০.৩৯ শতাংশ কমেছে, একই সময়ে মুনাফা ৪৪.৭৪ শতাংশ কোটি কমেছে। *(translation:The financial report of the listed multinational company shows that in the July–September quarter, sales revenue decreased by 10.39%, and profit dropped by 44.74%.)*	তালিকাভুক্ত বহুজাতিক কোম্পানির আর্থিক রিপোর্টে দেখা গেছে যে, জুলাই-সেপ্টেম্বর ত্রৈমাসিকে বিক্রয় রাজস্ব ১০.৩৯ শতাংশ কমেছে, একই সময়ে মুনাফা ৪৪.৭৪ শতাংশ কোটি কমেছে। (translation:The financial report of the listed multinational company shows that in the July–September quarter, sales revenue decreased by 10.39%, and profit dropped by 44.74%.) **Already *simplified***	X-axis: বিক্রয় রাজস্ব, মুনাফা Y-axis: ১০.৩৯ শতাংশ , ৪৪.৭৪ শতাংশ (translation: X-axis: Sales Revenue, Profit Y-axis: 10.39%, 44.74%)

Figure 7: Test dataset annotation process.