PRE-GENERATION PREDICTION OF LARGE LANGUAGE MODEL OUTPUT TOKEN LENGTH USING PROMPT EMBEDDINGS

KIRAH SAPONG [KSAPONG@SEAS.UPENN.EDU]

ABSTRACT. Accurately estimating the number of output tokens a large language model will generate prior to inference can improve cost estimation, scheduling, and resource allocation for downstream systems. In this work, we study the task of predicting output token length before generation using only the input prompt and its tokenized context. We derive a large-scale supervised dataset from real-world conversational interactions by pairing prompt histories with corresponding assistant responses and computing token counts under the generating model's tokenizer. We train gradient-boosted regression models using prompt embeddings and input token counts as features, and compare against baselines that use prompt length alone. Our results show that semantic prompt embeddings substantially improve prediction accuracy over prompt-length-only baselines, improving R^2 from -0.09 to 0.43 and increasing rank correlation from 0.26 to 0.77. However, remaining prediction error indicates that output length remains only partially predictable from prompt context alone. These findings suggest that lightweight pre-generation token predictors can provide useful coarse-grained estimates for inference-time budgeting and scheduling, while precise deterministic forecasting remains challenging.

1. INTRODUCTION

Large language model (LLM) inference cost and latency scale directly with the number of output tokens generated during autoregressive decoding. However, output length is generally unknown before generation begins, forcing serving systems to allocate compute, estimate cost, and manage token budgets without knowing request execution time in advance. This uncertainty complicates inference scheduling and resource planning, and has been identified as a key systems challenge in modern LLM serving pipelines [1, 2].

Although users may form rough expectations about response length based on prompt complexity or desired verbosity, output length is not determined by prompt length alone. Prompts of similar size may produce substantially different outputs depending on task type, conversational context, instruction style, and stochastic decoding behavior. Consequently, predicting output token length prior to generation is a nontrivial problem.

In this work, we study the task of pre-generation output token prediction: estimating the number of output tokens an LLM will generate before inference begins using only the input conversation context. We formulate this problem as a supervised regression task over prompt/response pairs and train predictive models using semantic prompt embeddings and tokenized input length as features. Using a large-scale dataset of real conversational interactions and model outputs, we evaluate whether semantic prompt representations improve prediction accuracy beyond simpler baselines based only on prompt length.

Recent systems work has explored output-length prediction for LLM scheduling and throughput optimization, often using auxiliary proxy models or internal model representations to estimate sequence length during serving [1, 3]. In contrast, we focus on a lightweight black-box setting in which output length is predicted solely from prompt context, without access to internal activations, logits, or partial decoding states. This formulation is particularly relevant for API-based inference providers, external scheduling systems, and pre-generation cost estimation.

1.1. Contributions.

This work makes the following contributions:

(1) We formulate pre-generation LLM output token prediction as a supervised regression problem over conversational context.
(2) We derive a task-specific evaluation dataset from a large-scale corpus of real conversational interactions by extracting input conversation history, final assistant outputs, and token counts under the generating model's tokenizer.
(3) We evaluate multiple predictive feature configurations, including prompt length alone, semantic prompt embeddings alone, and the combination of both.
(4) We show that semantic prompt embeddings substantially outperform prompt-length-only baselines for output token prediction, demonstrating that output length depends strongly on prompt semantics rather than prompt size alone.

(5) We analyze the practical utility and limitations of lightweight black-box token predictors for inference-time cost estimation, token budgeting, and scheduling.

2. Background

Let x denote an input conversation context provided to an LLM, represented as the sequence of prior user and assistant messages available before generation begins. Let y denote the number of output tokens generated by the model in response to x.

The objective of this work is to learn a predictor $f(x)$ that estimates the expected output token count prior to generation. Formally, we seek to approximate

$$f(x) \approx \mathbb{E}[y \mid x],$$

where prediction is performed using only the input context x, without access to the generating model's internal activations, logits, or partial decoding states.

Because language generation is stochastic, the true output length for a fixed prompt should be viewed as a random variable conditioned on the input context. As a result, deterministic prediction can at best estimate the expected output length under the model's decoding distribution.

Output token counts are nonnegative and follow a long-tailed distribution in practice. To stabilize variance and reduce the influence of large outliers during optimization, we model the transformed target

$$\tilde{y} = \log(1 + y)$$

during training, and transform predictions back to token-count space for evaluation.

To represent the input context x, we consider two classes of predictive features:

(1) **Input Token Count:** The total number of input tokens in the formatted prompt under the generating model's tokenizer.
(2) **Prompt Embeddings:** A dense semantic embedding of the formatted input conversation produced by a pretrained sentence embedding model.

Using these features, we formulate output token prediction as a supervised regression problem over a dataset

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N},$$

where each sample consists of an input conversation context x_i and the observed output token count y_i.

We evaluate predictive models over different feature subsets in order to determine whether semantic prompt representations provide signal beyond prompt length alone.

3. Related Work

Prior work on efficient LLM serving has studied methods for improving inference throughput, reducing latency, and optimizing scheduling under variable request workloads. Modern LLM serving systems exploit request characteristics such as sequence length to improve batching efficiency, memory allocation, and scheduling decisions during autoregressive decoding [4, 5]. However, many such systems assume output length is known in advance or rely on information only available after generation has partially begun.

Several recent works have explored sequence-length prediction for improving resource allocation in LLM serving pipelines. Proxy-model and uncertainty-aware scheduling approaches estimate generation length in order to improve batching and reduce latency during inference [1, 3]. These methods typically leverage internal model representations, auxiliary proxy models, partial decoding information, or tightly integrated serving-time instrumentation. While effective in systems where such information is available, they do not address the black-box pre-generation setting considered in this work.

Related ideas also appear more broadly in predictive systems and machine learning literature, where models are trained to estimate runtime, execution cost, or computational complexity from input features in order to improve scheduling and resource allocation. These works motivate the general framing of predictive resource estimation, but do not specifically study pre-generation output token forecasting for LLMs.

Our work differs from prior approaches by focusing on a lightweight black-box formulation: predicting total output token count before generation begins using only the prompt context, without access to internal activations, logits, partial decoding states, or serving-time instrumentation. This setting is particularly relevant for API-based inference providers,

external scheduling systems, and pre-generation cost estimation where the underlying model is accessed only through an opaque interface.

4. Approach

We formulate output token prediction as a supervised regression problem over prompt/response pairs derived from real conversational data. Each training example consists of an input conversation context and the corresponding number of output tokens generated by the assistant.

4.1. **Dataset Construction.** We derive our supervised dataset from the `re-align/lmsys-chat-outputs` corpus, which contains multi-turn conversational interactions generated by the `Zephyr-7B-Beta` language model [6, 7].

Because all examples are generated by a single underlying model under a fixed decoding configuration, the learned predictor estimates output length behavior specific to that generation regime.

For each conversation, we treat the final assistant message as the prediction target and all preceding messages as the input context. Formally, given a conversation

$$[m_1, m_2, \ldots, m_{T-1}, m_T],$$

where m_T denotes the final assistant response, we construct:

- **Input Context:** $x = [m_1, \ldots, m_{T-1}]$
- **Target Output:** $y = \text{token\_count}(m_T)$

Only conversations whose final message is authored by the assistant are retained.

4.2. **Prompt Formatting and Tokenization.** Input conversation histories are formatted using the official chat template associated with the `Zephyr-7B-Beta` tokenizer in order to preserve the prompt structure used during the original generation process. Input and output token counts are then computed using the same tokenizer to ensure consistency with the generating model.

4.3. **Feature Extraction.** For each formatted input prompt, we extract the following predictive features:

(1) **Input Token Count:** The total number of tokens in the formatted prompt.
(2) **Prompt Embedding:** A dense semantic embedding of the formatted prompt produced by the pretrained `all-mpnet-base-v2` sentence embedding model [8, 9].

We evaluate three feature configurations in order to measure the relative predictive value of prompt length and semantic information:

- **Baseline:** Input token count only
- **Embedding-Only:** Prompt embedding only
- **Full Model:** Input token count and prompt embedding combined

4.4. **Model Training.** Our primary predictive model is a gradient-boosted regression model implemented using XGBoost [10], trained to predict the log-transformed output token count

$$\tilde{y} = \log(1 + y).$$

Gradient boosting is chosen due to its strong empirical performance on structured tabular data and its ability to model nonlinear interactions between prompt length and semantic embedding features.

Models are trained using a 70/15/15 train/validation/test split. Early stopping is applied based on validation-set performance to reduce overfitting and select the final model checkpoint.

5. Experimental Results

We evaluate the proposed approach on the `re-align/lmsys-chat-outputs` dataset, which yields 304,246 prompt/response pairs after preprocessing. Each example is constructed by taking the full conversation history up to the final assistant message as the input context and the token count of the final assistant response as the prediction target.

5.1. **Data Processing.** All prompts are formatted using the official chat template of the `Zephyr-7B-Beta` tokenizer to ensure consistency with the original generation format. Input and output token counts are computed using the same tokenizer. Examples with zero output tokens are removed, as these correspond to degenerate or empty responses.

For each input context, we extract two classes of predictive features:

- Input token count
- Dense semantic prompt embeddings

Prompt embeddings are produced using the pretrained `all-mpnet-base-v2` sentence embedding model.

5.2. **Models and Evaluation Setup.** Our primary predictive model is a gradient-boosted regression model implemented using XGBoost. We evaluate three feature configurations:

- **Input-Only Baseline:** Input token count only
- **Embedding-Only Model:** Prompt embeddings only
- **Full Model:** Input token count and prompt embeddings combined

All models are trained using a 70/15/15 train/validation/test split with early stopping on the validation set.

We report the following metrics:

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- Median Absolute Error (MedAE)
- Coefficient of Determination (R^2)
- Spearman Rank Correlation
- Percentage of predictions within relative error thresholds (10%, 20%, 30%, 40%, 50%)

Absolute-error metrics quantify token-count prediction quality directly, while relative-error thresholds provide a scale-aware measure of forecasting usefulness across a broad range of output lengths. Spearman correlation measures the extent to which the model correctly ranks prompts by expected output length.

TABLE 1. Performance across feature configurations on the held-out test set.

Model	MAE	RMSE	MedAE	R^2	Spearman	10%	20%	30%	40%	50%
Input-Only Baseline	304.69	515.47	163.42	-0.09	0.26	7.20%	15.35%	23.94%	32.83%	42.27%
Embedding-Only	185.04	374.32	85.16	0.42	0.77	17.69%	33.43%	46.96%	57.86%	66.75%
Full Model	184.05	373.41	84.24	0.43	0.77	18.01%	33.72%	47.19%	58.13%	66.89%

5.3. **Quantitative Results.** The input-only baseline performs poorly across all metrics, achieving negative R^2, weak rank correlation, and low relative-error accuracy. This indicates that prompt length alone is not a reliable predictor of output token count and performs worse than a constant-mean predictor under squared error.

In contrast, models using semantic prompt embeddings substantially improve prediction performance. The embedding-only model improves R^2 from -0.09 to 0.42 and Spearman correlation from 0.26 to 0.77, demonstrating that semantic information contained in the prompt is strongly predictive of output length.

The full model, which combines embeddings and input token count, performs only marginally better than the embedding-only model. This suggests that semantic embeddings already encode nearly all useful information provided by prompt length and that explicit token-count features add minimal additional signal.

Despite these gains, prediction accuracy remains moderate in absolute terms. The best-performing model achieves an MAE of 184 tokens, with 18% of predictions falling within 10% relative error and 34% within 20%. However, performance is stronger at coarser tolerances: 47% of predictions fall within 30%, 58% within 40%, and 67% within 50%. This suggests that the model is more useful for coarse-grained length estimation than for precise point prediction.

5.4. **Prediction Calibration and Error Behavior.** Figure 1 plots predicted versus true output token counts for the full model. The model captures the overall monotonic relationship between prompt context and output length, consistent with the strong Spearman correlation reported in Table 1. However, the scatter also reveals substantial dispersion around the diagonal, particularly for longer outputs. Predictions for very long responses tend to lie below the diagonal, indicating systematic underprediction in the upper tail.

Figure 2 shows the distribution of residual errors. Residuals are concentrated near zero, but the distribution exhibits heavy tails and a noticeably longer negative tail, indicating that the model occasionally underpredicts output length by

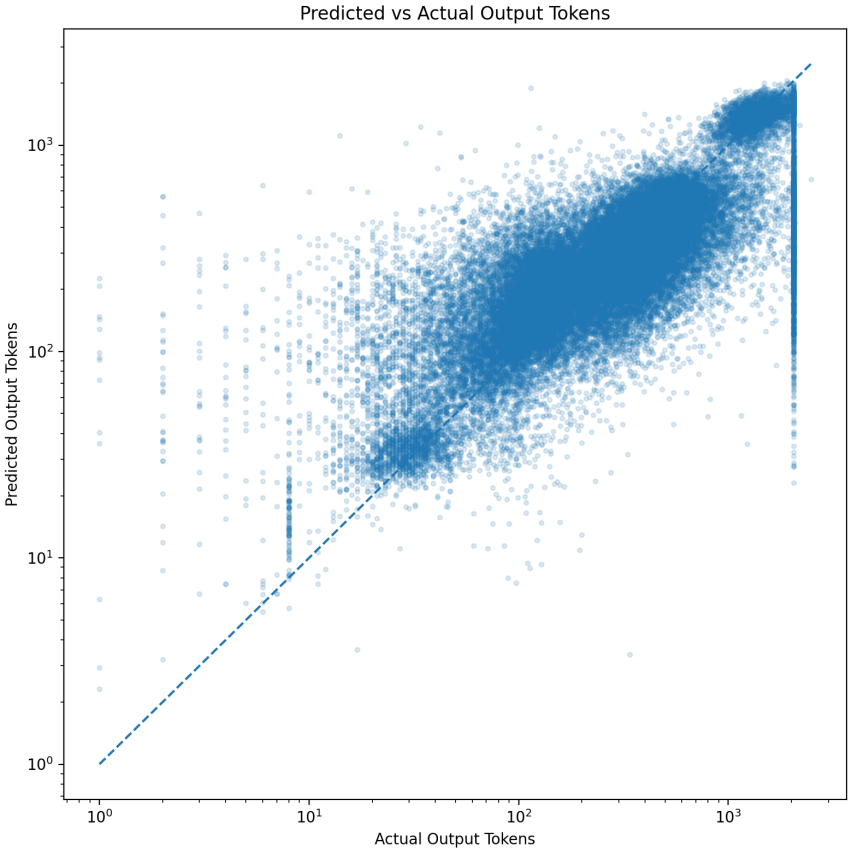

FIGURE 1. Predicted versus true output token counts for the full model on the test set.

a large margin. This is consistent with the behavior observed in Figure 1, where extremely long responses are often predicted too conservatively.

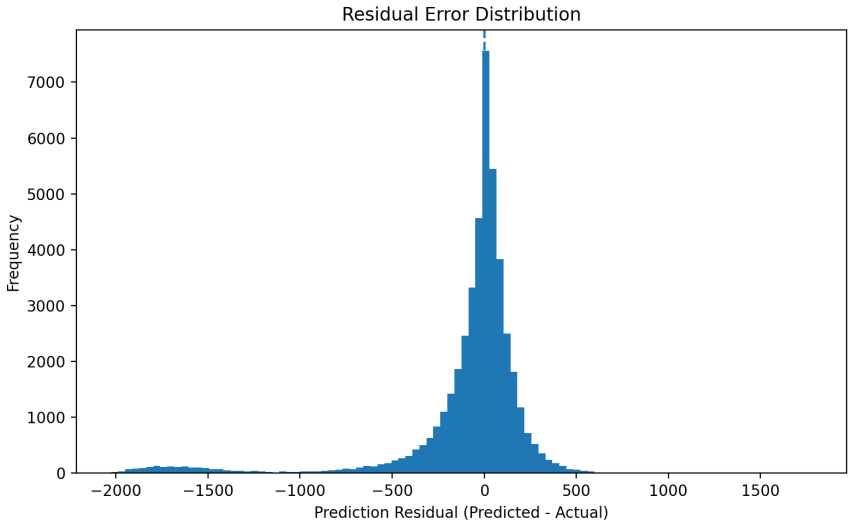

FIGURE 2. Distribution of residual prediction errors for the full model.

Figure 3 plots absolute prediction error as a function of true output length. Absolute error increases with output length, indicating longer responses are harder to predict in exact token counts. This helps explain why MAE remains relatively large even when rank correlation is strong.

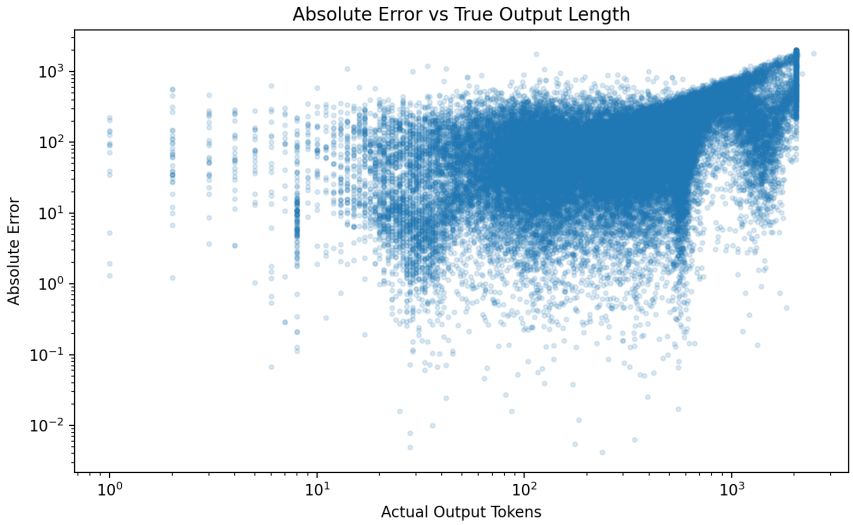

FIGURE 3. Absolute error versus true output token count for the full model.

Figure 4 shows relative error versus true output length. Relative error is highest for very short outputs and decreases as target length grows. This indicates that short responses are the hardest to predict proportionally, while longer responses, although still difficult in absolute terms, are easier to estimate at the level of relative scale.

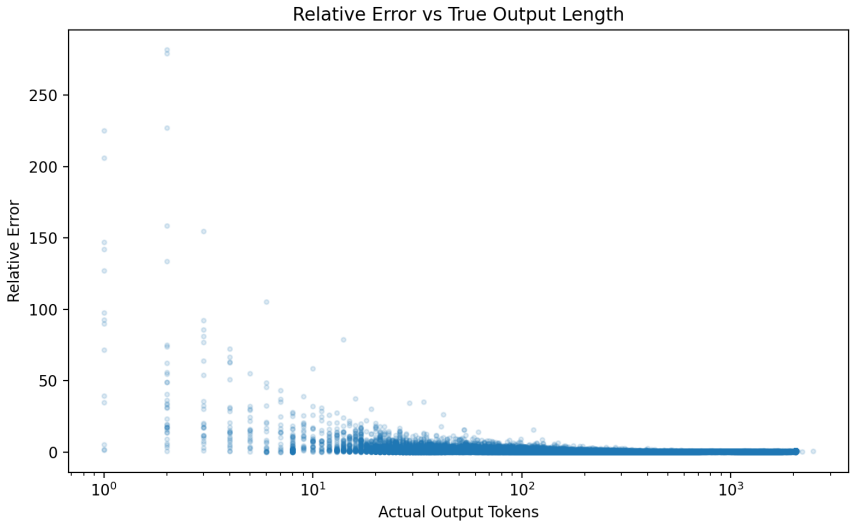

FIGURE 4. Relative error versus true output token count for the full model.

Taken together, these plots indicate that the model is better at identifying whether a prompt will elicit a short, medium, or long response than at predicting exact token counts. The predictor is therefore better characterized as a coarse response-length estimator than a precise token-count forecaster.

5.5. **Qualitative Examples.** Table 2 presents representative examples from the full model on hand-selected prompts spanning a range of expected output lengths.

TABLE 2. Example predictions from the full model on representative prompts.

Prompt	Predicted Tokens	Actual Tokens
What is the capital of Japan?	44.2	87
Explain how quicksort works in simple terms.	217.2	279
Write a detailed essay on the causes of World War I.	500.0	726
Write a Python implementation of a trie with insert, search, and delete.	759.4	855
Tell me about AI.	215.8	553

These examples reinforce the quantitative findings. For prompts with relatively constrained outputs, such as short factual or medium explanatory questions, the model produces predictions that are directionally reasonable, though often somewhat conservative. For longer prompts, the model generally captures that the response should be large, but still tends to underpredict the final token count. The prompt *"Tell me about AI"* is short, but open-ended, permitting a wide range of valid response lengths, and the model underestimates the assistant's verbosity by a large margin. This highlights a core limitation of black-box prompt-only prediction: prompts that admit many valid responses remain difficult to forecast precisely.

5.6. **Interpretation.** These results suggest three primary conclusions.

First, output token count is not well explained by prompt length alone, contradicting simplistic heuristics that assume a direct relationship between input and output size.

Second, semantic representations of the prompt provide substantial predictive power, indicating that task type, instruction style, and conversational context strongly influence response length.

Third, even with rich semantic features, prediction accuracy remains limited, suggesting that output length is noisy and influenced by stochastic aspects of the generation process. The plots further show that this uncertainty is not uniform. Short outputs are difficult in relative terms, while long outputs are difficult in absolute terms.

Overall, the results indicate that pre-generation token prediction is feasible but imperfect. Semantic prompt embeddings enable meaningful improvements over naive baselines, and the model is effective at coarse ranking and broad response-length estimation. However, a significant portion of output-length variability remains uncertain in the black-box setting.

6. DISCUSSION

Our results indicate that pre-generation output token prediction based on input context is feasible, but fundamentally limited in precision. While semantic prompt embeddings substantially improve performance over naive baselines based on input length, a large portion of output-length variance remains unexplained by prompt representations alone.

The poor performance of the input-length-only baseline demonstrates that prompt size alone is insufficient for forecasting output length. Instead, the substantial performance gap between length-only and embedding-based models suggests that semantic properties of the prompt are far more predictive than raw prompt length.

At the same time, the embedding-based model achieves only moderate absolute accuracy. The predictor is considerably better at coarse-grained length estimation than precise token-count forecasting. It reliably distinguishes between prompts likely to produce short, medium, and long responses, but struggles to predict exact output lengths.

Several limitations of the current approach remain. First, the current formulation predicts only a point estimate of expected output length. Given the stochasticity of the task, future work may benefit from predicting output-length distributions, quantiles, or confidence intervals rather than a single deterministic estimate. Such probabilistic forecasts may be more useful for downstream scheduling, budgeting, and admission-control systems.

Second, this study conditions prediction only on prompt context and ignores explicit decoding configuration. In practice, output length is strongly influenced by generation hyperparameters such as temperature, top-p, repetition penalties, stop sequences, and maximum token limits. Future work could incorporate such decoding parameters directly as model inputs in order to estimate $P(y \mid x, c)$, where c denotes generation configuration, potentially improving predictive performance when inference parameters are known in advance.

Third, the current approach relies on frozen pretrained sentence embeddings. While these embeddings capture general semantic information, they are not optimized specifically for token-length prediction. Future work could

explore jointly learned task-specific representations, finetuned embedding models, or direct encoder architectures trained end-to-end for this objective.

Finally, broader improvements may require larger and more diverse training corpora spanning multiple language models, decoding settings, and application domains. The current study focuses on outputs from a single underlying language model under a fixed generation configuration, and performance may differ when generalizing across heterogeneous decoding regimes.

Overall, these findings suggest that lightweight black-box pre-generation token predictors can provide meaningful signal for inference planning, coarse-grained budgeting, and request ranking, but precise deterministic forecasting of output length remains challenging in the black-box setting.

REFERENCES

[1] Qiu, H., et al. *Efficient Interactive LLM Serving with Proxy Model-Based Sequence Length Prediction.* 2024.
[2] Fu, Y., et al. *Efficient LLM Scheduling by Learning to Rank.* 2024.
[3] Zheng, H., et al. *Scheduling LLM Inference with Uncertainty-Aware Output Length Predictions.* 2026.
[4] Yu, G., et al. *Orca: A Distributed Serving System for Transformer-Based Generative Models.* 2022.
[5] Kwon, W., et al. *Efficient Memory Management for Large Language Model Serving with PagedAttention.* 2023.
[6] Tunstall, L., et al. *Zephyr: Direct Distillation of LM Alignment.* 2023.
[7] re-align. *LMSYS Chat Outputs Dataset.* Hugging Face Datasets, 2024.
[8] Reimers, N. and Gurevych, I. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.* 2019.
[9] Song, K., et al. *MPNet: Masked and Permuted Pre-training for Language Understanding.* 2020.
[10] Chen, T. and Guestrin, C. *XGBoost: A Scalable Tree Boosting System.* 2016.

APPENDIX A. SUPPORTING CODE

The following source files are included with the submission to support reproducibility of the experimental pipeline:

- `preprocess_data.py`: Preprocesses raw conversational data into prompt/response token-count training examples.
- `train_token_predictor.py`: Trains XGBoost-based token prediction models and evaluates performance on held-out data.
- `plot_results.py`: Generates evaluation plots from model prediction outputs.
- `check_predictions.py`: Runs qualitative inference examples on the trained predictor for manual inspection.