PromptCast: A New Prompt-based Learning Paradigm for Time Series Forecasting

Hao Xue, Flora D. Salim

School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

Abstract—This paper presents a new perspective on time series forecasting. In existing time series forecasting methods, the models take a sequence of numerical values as input and yield numerical values as output. The existing SOTA models are largely based on the Transformer architecture, modified with multiple encoding mechanisms to incorporate the context and semantics around the historical data. Inspired by the successes of pre-trained language foundation models, we pose a question about whether these models can also be adapted to solve time-series forecasting. Thus, we propose a new forecasting paradigm: prompt-based time series forecasting (PromptCast). In this novel task, the numerical input and output are transformed into prompts and the forecasting task is framed in a sentence-to-sentence manner, making it possible to directly apply language models for forecasting purposes. To support and facilitate the research of this task, we also present a large-scale dataset (PISA) that includes three real-world forecasting scenarios. We evaluate different SOTA numerical-based forecasting methods and language generation models. The benchmark results with various forecasting settings demonstrate the proposed PromptCast with language generation models is a promising research direction. Additionally, in comparison to conventional numerical-based forecasting, PromptCast shows a much better generalization ability under the zero-shot setting.

Index Terms—time series forecasting, natural language generation, dataset and benchmark

I. INTRODUCTION

Time series forecasting is a research-intensive field, especially with the increasing of applying various deep learning frameworks for prediction such as models based on LSTM [1], Temporal Convolutional Network (TCN) [2], and Transformer [3]. More recently, we are witnessing a fast growth of large-scale pre-trained models in the Natural Language Processing (NLP) field. These models, also known as foundation models [4], are often pre-trained with an extremely large amount of data and have demonstrated good performance across various downstream tasks. For example, BERT [5] can be adapted for multiple NLP tasks, CLIP [6] and GLIP [7] are good at CV tasks. However, we also notice that this evolution seems mostly limited to the NLP and CV fields. Hence, we are particularly interested in exploring the research question of whether we can take advantage of large-scale pre-trained foundation models and adapt these models for predicting time series. To investigate this question, in this paper, we formally introduce a novel task: prompt-based time series forecasting (PromptCast). The existing forecasting methods

Hao Xue and Flora D. Salim are with the School of Computer Science and Engineering , UNSW Sydney, hao.xue1@unsw.edu.au, flora.salim@unsw.edu.au

including the state-of-the-art Transformer-based forecasting models [8], [9], [10], [11], [12], [13] can be simplified as a numerical forecasting paradigm as shown in Figure 1 (a). Numerical forecasting methods always take numerical values as input and generate numerical values as the prediction for the next time step. Instead, the input and output of the proposed prompt-based forecasting (Figure 1 (b)) are natural language sentences. This paradigm change enables the utilization of language generation models for forecasting.

This new forecasting paradigm is beneficial in multiple aspects. PromptCast presents a novel "code less" solution for time series forecasting, which could provide a new perspective rather than purely focusing on designing more and more complicated deep learning forecasting models (*e.g.*, Transformer-based Informer, Autoformer, and FEDformer). It also becomes a relatively easy-accessible and user-friendly method for non-researcher users, compared to existing forecasting models that require many tedious parameter searching and training processes, especially under a new forecasting scenario. As pointed out in a recent research [14], the benefits of using a single neural model across different tasks are significant. The PromptCast task explores the potential of using language foundation models for the forecasting task which could make it possible to broaden the language models beyond the realm of typical text-based tasks. In addition, it could inspire new research directions and new applications to better serve society. For example, as illustrated in Figure 1 (c), chatbot with forecasting ability is one of the prospective future applications driven by the research of PromptCast task. Currently, although AI-powered intelligent assistants or chatbots like Siri and Alexa can answer queries about general topics, they still fail to answer specific time series forecasting questions. With the help of PromptCast related research, they would be able to yield predictions based on the given contexts.

To the best of our knowledge, this is the first effort in approaching general time-series forecasting from a language-based perspective without any modifications on the model architectures, resulting also in the first large-scale dataset, PISA (**P**rompt based t**I**me **S**eries forec**A**sting), tailored for the task of prompt-based time series forecasting. It covers three real-world forecasting scenarios: weather temperature forecasting, energy consumption forecasting, and customer flow forecasting. We believe that the release of this dataset will not only support the research of the PromptCast task but also have a great potential to stimulate related research in the time series analysis domain. We further develop a benchmark in which we report the forecasting performance of
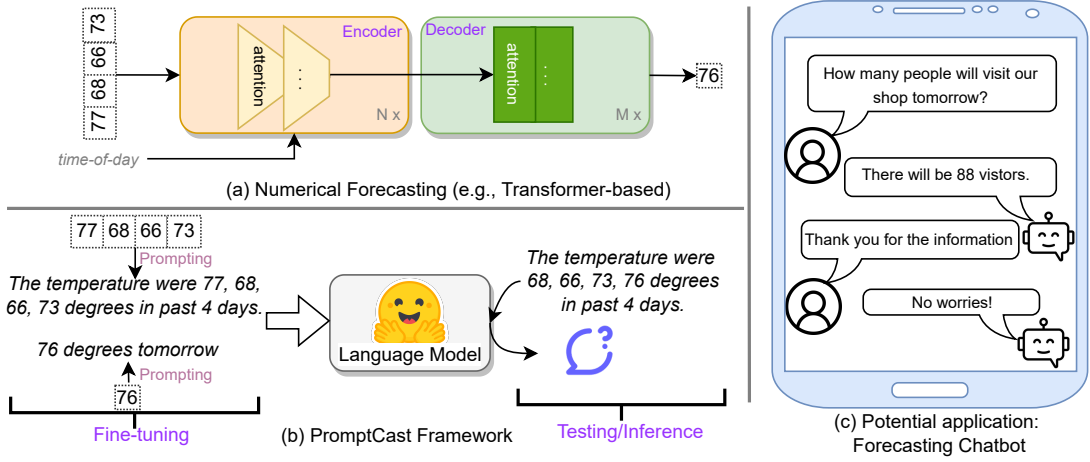
Fig. 1. Conceptual illustrations of (a) existing numerical-based forecasting; (b) the framework of the proposed PromptCast; (c) a potential forecasting chatbot application based on PromptCast.

multiple methods including both numerical-based forecasting methods and language generation models. To evaluate the generalization ability of PromptCast, this benchmark also explores various forecasting settings such as train-from-scratch, zero-shot prediction, multi-step forecasting, and multivariate forecasting. In summary, our contributions are three-fold: **(1)** We propose a novel prompt-based forecasting paradigm, which differs from the existing forecasting methods. This is the first time the general time series forecasting problem is addressed in a natural language generation manner. **(2)** We release a large-scale dataset (PISA) with 311,932 data instances in total for the newly introduced task. The dataset covers diverse time series forecasting scenarios. **(3)** We develop a benchmark (our data and codes for the benchmark are available at https:// github.com/HaoUNSW/PISA) on the proposed PISA datasets. It evaluates the state-of-the-art numerical-based forecasting methods and popular language generation models under the PromptCast setting.

II. PROMPT-BASED TIME SERIES FORECASTING

The prompt-based time series forecasting task is developed from the general time series task. Here we first describe the general numerical-based forecasting and then formulate the proposed PromptCast task. Let $\mathcal{U} = \{U_1, U_2, \cdots, U_M\}$ denotes a set of M *objects-of-interest*. Depending on different specific forecasting scenarios, the objects-of-interest could stand for different objects. For example, the objects could be places-of-interest (POI) such as bars and parks in human mobility forecasting [15] or cities in weather forecasting. Under the general numerical time series forecasting task setting, the input is a historical record of interested numerical data points collected on t_{obs} continuous time steps (*e.g.*, daily data): $x^m_{t_1:t_{\text{obs}}} = [x^m_{t_1}, x^m_{t_2}, \cdots, x^m_{t_{\text{obs}}} \mid x^m_t]$, where x^m_t represents the value of object-of-interest U_m observed on time step t. The forecasting target (output) is the numerical data value $x^m_{t_{\text{obs}+1}}, x^m_{t_{\text{obs}+2}}, \cdots, x^m_{t_{\text{obs}+n}}$ of the next n time steps $t_{\text{obs}+1}, t_{\text{obs}+2}, \cdots, t_{\text{obs}+n}$ (n is often referred as the prediction horizon).

The overarching goal of the PromptCast task is to leverage language foundation models to forecast time series in a sentence-to-sentence fashion. In order to achieve this goal, based on the above formulated problem of numerical time series forecasting, the numerical values need to be transferred and described as natural language sentences. This data-to-text transformation is referred to as a prompting process in this work (the details of prompting are presented in the next section). Specifically, the input numerical sequence $x^m_{t_1:t_{\text{obs}}}$ is turned into input prompts, and the forecasting target values are transformed as the output prompt. Consequently, time series forecasting can be addressed through a natural language generation paradigm, and language foundation models can be adopted as the core forecasting models in PromptCast task. For simplification, we primarily use the fundamental univariate time series single-step forecasting ($n = 1$) as the default setting to illustrate the concept of our novel PromptCast paradigm in the main paper. More detailed discussion and experimental results of PromptCast in multi-step forecasting and multivariate forecasting are reported in Sec. IV-H and Sec. IV-I, respectively.

III. DATASET DESIGN AND DESCRIPTION

In this section, we demonstrate the design and construction of the proposed PISA dataset. The overall designing guideline is: (1) to preprocess original data given in the numerical format (raw data) for the forecasting task setting (Sec. III-A) and (2) to transform the numerical data to natural language input/output formats with prompts (Sec. III-B). We also describe the features and statistics (Sec. III-C).

A. Data Sources and Processing

To create a diverse dataset, we consider three real-world forecasting scenarios (3 sub-sets of our PISA dataset) from various domains: weather forecasting, energy consumption forecasting, and human mobility forecasting. The data sources for these scenarios are:

- **City Temperature (CT)**: This data source[1] provides

[1] https://academic.udayton.edu/kissock/http/Weather/default.htm

TABLE I
TEMPLATES FOR TRANSFORMING PISA-NUMERICAL TO PISA-PROMPT.

			Template	Example
CT	Input Prompt (Source)	Context	From $\{t_1\}$ to $\{t_{\text{obs}}\}$, the average temperature of region $\{U_m\}$ was $\{x^m_{t_1:t_{\text{obs}}}\}$ degree on each day.	From August 16, 2019, Friday to August 30, 2019, Friday, the average temperature of region 110 was 78, 81, 83, 84, 84, 82, 83, 78, 77, 77, 74, 77, 78, 73, 76 degree on each day.
		Question	What is the temperature going to be on $\{t_{\text{obs}+1}\}$?	What is the temperature going to be on August 31, 2019, Saturday?
	Output Prompt (Target)	Answer	The temperature will be $\{x^m_{t_{\text{obs}+1}}\}$ degree.	The temperature will be 78 degree.
ECL	Input Prompt (Source)	Context	From $\{t_1\}$ to $\{t_{\text{obs}}\}$, client $\{U_m\}$ consumed $\{x^m_{t_1:t_{\text{obs}}}\}$ kWh of electricity on each day.	From May 16, 2014, Friday to May 30, 2014, Friday, client 50 consumed 8975, 9158, 8786, 8205, 7693, 7419, 7595, 7596, 7936, 7646, 7808, 7736, 7913, 8074, 8329 kWh of electricity on each day.
		Question	What is the consumption going to be on $\{t_{\text{obs}+1}\}$?	What is the consumption going to be on May 31, 2014, Saturday?
	Output Prompt (Target)	Answer	This client will consume $\{x^m_{t_{\text{obs}+1}}\}$ kWh of electricity.	This client will consume 8337 kWh of electricity.
SG	Input Prompt (Source)	Context	From $\{t_1\}$ to $\{t_{\text{obs}}\}$, there were $\{x^m_{t_1:t_{\text{obs}}}\}$ people visiting POI $\{U_m\}$ on each day.	From May 23, 2021, Sunday to June 06, 2021, Sunday, there were 13, 17, 13, 20, 16, 16, 17, 17, 19, 20, 12, 12, 14, 12, 13 people visiting POI 324 on each day.
		Question	How many people will visit POI $\{U_m\}$ on $\{t_{\text{obs}+1}\}$?	How many people will visit POI 324 on June 07, 2021, Monday?
	Output Prompt (Target)	Answer	There will be $\{x^m_{t_{\text{obs}+1}}\}$ visitors.	There will be 15 visitors.

the daily average temperature (in Fahrenheit degrees) of multiple cities globally. 110 international cities are randomly selected to form the dataset.

- **Electricity Consumption Load (ECL)**: The original data includes the electricity consumption values (in Kwh) of 321 users[2]. We eliminated users with incomplete records and randomly selected 50 users with complete data for the entire collection period. Additionally, the hourly usage values for each chosen user were consolidated into daily usage data.
- **SafeGraph Human Mobility Data (SG)**: This real-world human mobility data from SafeGraph Weekly Patterns[3] contains the daily raw counts of visitors to POIs. We expanded the data collection from 5 months in [15] to almost 15 months and then randomly selected 324 POIs with full records.

The exact data collection periods are reported in Table II. Following the standard protocol [11], [15], each sub-set is divided into train/val/test at the ratio of 7:1:2 by the chronological order (Table II). The numerical sequence of each object-of-interest in each sub-set is then split into multiple instances (for training/validation/test) by applying sliding windows. The window size equals to $t_{\text{obs}} + 1$ (including t_{obs} time steps as input historical data and 1 step as the forecasting target) and the step size of the sliding window is 1 day. Specifically, following previous work [15], the observation length of the input sequence is set as 15 (i.e., $t_{\text{obs}} = 15$). To distinguish the numerical data used for numerical methods and the language-based dataset processed for language models, the numerical sequences processed by the above sliding window is referred

as PISA-numerical whereas the other is named as PISA-prompt (details given in next subsection).

Ethical Considerations. The only possible sensitive information is the identifier of the object-of-interest (e.g., the POI id in SG). To remove this, we randomly assigned object-of-interest index U_m starting from 1 to M. Given that the original data source for each subset is aggregate statistics with no personally identifiable information, the generated PISA dataset does not contain any private information that can be deciphered.

B. Template-Based Prompting

The core of the proposed PromptCast task is shaping the time series forecasting in a language generation manner. To serve this purpose, a key step in building a dataset for Prompt-Cast is to describe and transform the numerical sequential data (i.e., PISA-numerical) to natural language sentences. As demonstrated in [15], using template-based description is an effective and efficient approach to achieve the data-to-text transformation. In this work, we explicitly introduce three templates for the three sub-sets and Table I lists the templates and the corresponding examples.

In a nutshell, the template consists of two main parts: input prompt and output prompt. The input prompt covers the description of the historical observation and the indicators of the prediction target time step (i.e., $t_{\text{obs}+1}$). The output prompt handles the desired prediction value ($x^m_{t_{\text{obs}+1}}$) which is used as the ground truth label for training or evaluation. This input/output prompt setting is similar to the source/target sentence in machine translation. For researchers who are more familiar with the open question answering setting, our PISA dataset can also be interpreted as a question answering task setting. The input prompt can be broken into the context part and the question part. The context provides the historical information for forecasting and the question part can be seen as

---

[2]https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014
[3]https://docs.safegraph.com/docs/weekly-patterns#section-weekly-patterns-schema

		CT	ECL	SG
Objects-of-interest		110 cities	50 Users	324 POIs
Collection Period		From 2017/01/01 To 2020/04/30	From 2012/01/01 To 2014/12/31	From 2020/06/15 To 2021/09/05
Training Set		From 2017/01/01 To 2019/04/30 850 days 91850 instances	From 2012/01/01 To 2017/01/31 762 days 37350 instances	From 2020/06/15 To 2021/04/23 313 days 96552 instances
Validation Set		From 2019/05/01 To 2019/08/31 123 days 11880 instances	From 2014/02/01 To 2014/05/31 120 days 5250 instances	From 2021/04/24 To 2021/06/07 45 days 9720 instances
Test Set		From 2019/09/01 To 2020/04/30 243 days 25080 instances	From 2014/06/01 To 2014/12/31 214 days 9950 instances	From 2021/06/08 To 2021/09/05 90 days 24300 instances
Value Range Average Value		[-44, 104] 58.070	[2799, 24906] 11479.120	[3, 383] 29.355

the input query about the future. Naturally, the output prompt is the ground truth answer that responds the question. Based on the templates and the processed numerical sequences, PISA-prompt is then generated. Note that the instances in PISA-numerical map the instances in PISA-prompt. For example, the first instance in PISA-prompt is transferred from the first instance in PISA-numerical. This is to ensure that they can be used to compare the performance of numerical forecasting methods and the language-based forecasting methods in the benchmark. Our PISA-prompt provides the input prompts and the corresponding output prompts in separate files (*e.g.*, val_x_prompt.txt and val_y_prompt.txt).

C. Statistics Overview

To highlight the diversity of PISA, we analyze its key statistics as given in Table II. PISA dataset comprises a total of 311,932 instances from three different forecasting application domains. Each sub-set has its distinct statistical characteristics. The last row of Table II lists the value range distributions and the average value of each sub-set. Furthermore, Figure 2 shows the distribution plots of three sub-sets in our PISA dataset. It can be observed that the distributions of the three sub-sets are different. As shown in the figure and according to the value ranges reported in Table II, these three selected data sources ensure the data diversity of our PISA datasets. Specifically, the proposed PISA covers a range of negative numerical values (the CT sub-set), large values (the ECL sub-set), and small/regular values (the SG sub-set). This diversity of data in our dataset ensures the representativeness of our PISA dataset and provide a comprehensive benchmark for the performance of different models under different scenarios.

IV. BENCHMARK

In this section, we present our benchmarking study and analysis for the proposed PromptCast task. By conducting experiments on the established PISA dataset, we aim to address the following two main research questions: **RQ1**: Can we use language generation models to predict time series under the PromptCast task setting? Compared to the conventional numerical-based time-series forecasting methods, what is the performance of language generation-based forecasting models? **RQ2**: Can forecasting time series with prompts as well as using language generation models achieve better generalization ability?

A. Evaluation Metrics

While the proposed PromptCast task is a language generation task that aims to generate the target output prompts, we are particularly interested in evaluating the time series forecasting performance. To this end, the first step of the evaluation protocol is to extract the predicted numerical values from the generated sentences. Given that the output prompts follow the same template for each sub-set (*e.g.*, "There will be ..." in the SG sub-set), the numerical value can be easily extracted by simple string parsing. However, in practice, due to the uncertainty of the inference process, it is not guaranteed that the numerical value can be extracted from the generated output for every testing instance. To reflect this in the evaluation, we explicitly introduce the Missing Rate as one evaluation metric. It is defined as $(n_{\text{test}} - n_{\text{decoded}})/n_{\text{test}} \times 100\%$ where n_{test} and n_{decoded} are the total number of instances in the test set and the number of generated instances that can successfully decode the predicted value, respectively. A lower Missing Rate indicates better performance.

After extracting the predicted numerical values, the evaluation of the PromptCast task will be similar to the evaluation of traditional numerical-based forecasting methods. Therefore, we also use two widely-used metrics, the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE), to evaluate the prediction performance in our benchmark. For each deep learning method, we report the average performance and the standard deviation over five runs with different random seeds. The Missing Rate is not considered when evaluating numerical-based methods as it is not applicable.

B. Baselines

In order to provide a comprehensive benchmark for the proposed PromptCast task, we test the performance of 10 popular natural language generation models on our PISA dataset (*i.e.*, PISA-prompt). These language models are T5 [16], Bart [17], BERT [5], RoBERTa [18], Electra [19], Bigbird [20], ProphetNet [21], LED [22], Blenderbot [23], and Pegasus [24]. Furthermore, for the comparison purpose (RQ1) and providing strong baselines for forecasting with prompts methods, we also include the performance of conventional numerical paradigm forecasting methods on the PISA-numerical. We consider 3 naive forecasting methods: Copy Yesterday (CY), Historical Average (HA), and Copy Last Week (CLW). We also consider 3 basic numerical forecasting methods: AutoARIMA, LSTM, and temporal convolutional network (TCN). Transformer-based time series forecasting methods including the vanilla Transformer [3], the state-of-the-art Informer [9], Autoformer [11], and FEDformer [12] are also

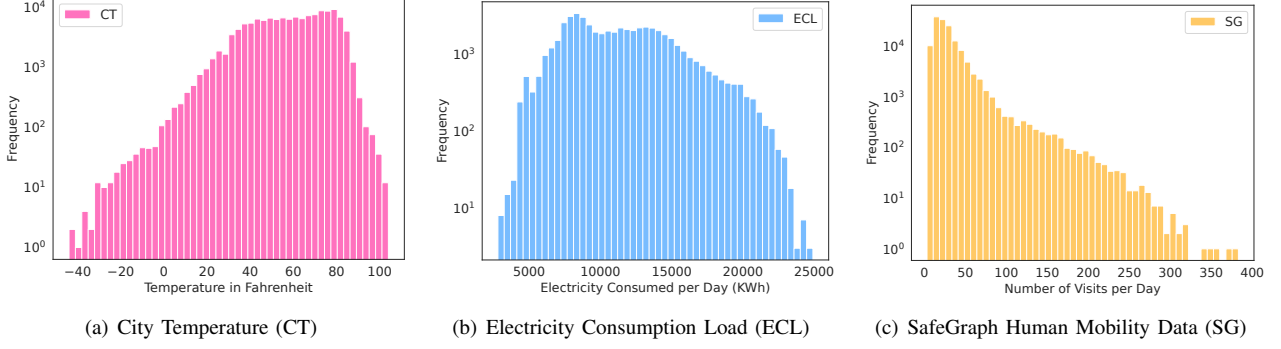(a) City Temperature (CT)　　　(b) Electricity Consumption Load (ECL)　　　(c) SafeGraph Human Mobility Data (SG)

Fig. 2. The distribution plots of three sub-sets: (a) the City Temperature (CT) sub-set includes negative values; (b) the Electricity Consumption Load (ECL) sub-set covers large numbers spanning a wide range; and (c) the sub-set of SafeGraph Human Mobility Data (SG) involves relatively small values ranging from 0 to 400.

included. Overall, we consider 20 different methods (10 numerical methods and 10 language models) in the current benchmark.

C. Implementation Details

For the evaluated numerical forecasting methods, the implementations are based on the official FEDformer[4] repository which also includes the implementation of Transformer, Informer, and Autoformer. We follow the default hyperparameter settings except for the *pred_len* parameter (used in Informer, FEDformer, and Autoformer). For numerical methods, the data normalization process is also included when we processed the numerical data. In our experiments, the *pred_len* parameter is set to 7 (around half of the observation length, which is 15 in PISA). As for the hyperparameter searching, the factor c (used in Informer and Autoformer) is the one that could affect the forecasting performance of numerical forecasting methods [11]). The default factor given by the official implementation is 3 and this default factor has also been used for different datasets according to the official implementations. This justifies that it is a reasonable choice when we evaluate Informer and Autoformer performance on our PISA dataset.

For the language models in the benchmark, their implementations can be grouped into two categories. The first category follows the *EncoderDecoderModel*[5] framework. Three language models are implemented under this category: BERT, RoBERTa, Electra. The rest 7 language models are accomplished through the second category, which is the *ConditionalGeneration* in HuggingFace (*e.g.*, *BartForConditionalGeneration* class[6]). The fine-tuning process in PromptCast is based on the standard Trainer provided by HuggingFace. Specifically, the sequence-to-sequence trainer is applied as our downstream time series forecasting task in PromptCast is a sequence-to-sequence task. Additionally, no modifications are introduced to the loss function during fine-tuning. For decoding the generation from the language models, the standard tokenizers provided by HuggingFace are used to detokenize

the direct output tokens to yield sentences and there are no extra regularization steps on the yielded sentences to acquire numerical predicted outputs. We simply decode the numerical values through string parsing.

We would like to emphasize that the language models and numerical forecasting methods are treated and processed equally and fairly in our benchmark. There is also no specific hyperparameter tuning for language models under PromptCast. For both language models and numerical models, we all use the default settings provided/recommended by the official implementations. Thus, we believe that the comparison is fair and convincing. This hyperparameter searching-free characteristic could also reflect another benefit of PromptCast in real-world applications, that is, no need to conduct complicated and time-consuming hyperparameter tuning processes. The forecasting models can then be deployed more quickly for new forecasting scenarios.

All the above scripts are provided in our GitHub repository[7]. The experiments were performed with PyTorch on a Linux server equipped with Nvidia V100 GPUs. Note that in our experiments, only 1 GPU is enabled per run for each method.

D. Experimental Performance

1) Numerical-Based Methods: This part focuses on evaluating the typical numerical-based forecasting methods with our PISA dataset. Normally, the position embeddings used in the Transformer architecture (and its variants) only contain the limited position information (*e.g.*, the first time step in each input sequence). This kind of position information remains the same for all different input data instances. However, for time series data, temporal information (*e.g.*, *day-of-week* and *month-of-year*) is an important cue for predicting future data and reflects the global position relations. For example, the first time step of instance A could correspond to Monday whereas the first time step (same position) of instance B could be Friday. Thus, appending the temporal embeddings to the basic position embedding becomes popular in Transformer-based time series forecasting methods. This is equivalent to providing temporal context in the input prompt (*i.e.*, From

---

[4]https://github.com/MAZiqing/FEDformer

[5]https://huggingface.co/docs/transformers/model_doc/encoder-decoder

[6]https://huggingface.co/docs/transformers/v4.19.2/en/model_doc/bart#transformers.BartForConditionalGeneration

[7]https://github.com/HaoUNSW/PISA

TABLE III
RESULTS OF NUMERICAL-BASED FORECASTING METHODS ON PISA-NUMERICAL.

Method	Temporal Embedding	CT		ECL		SG	
		RMSE	MAE	RMSE	MAE	RMSE	MAE
CY	N/A	6.710	4.991	680.142	381.247	10.945	7.691
HA	N/A	8.089	6.321	694.658	455.288	9.198	6.221
CLW	N/A	10.352	7.950	835.590	553.485	10.387	7.381
AutoARIMA	N/A	6.904	5.234	644.253	387.608	9.290	6.383
LSTM	N/A	6.511±0.053	4.956±0.056	598.962±2.027	367.798±2.088	8.994±0.032	6.107±0.011
TCN	N/A	6.397±0.089	4.876±0.072	589.785±6.280	368.682±6.077	8.389±0.029	5.927±0.039
Transformer	timeF	6.790±0.072	5.238±0.058	612.102±25.081	400.182±24.956	8.230±0.029	5.851±0.023
	fixed	6.603±0.177	4.989±0.137	557.813±22.754	357.253±6.875	8.274±0.035	5.856±0.036
	learned	6.873±0.143	5.294±0.108	567.307±10.261	394.226±8.900	8.408±0.274	5.940±0.103
Informer	timeF	6.778±0.085	5.195±0.075	597.011±15.373	383.704±21.694	8.167±0.049	5.832±0.032
	fixed	6.457±0.268	4.922±0.209	**536.921±33.375**	**349.331±11.916**	**8.151±0.068**	5.868±0.049
	learned	6.844±0.106	5.307±0.083	561.661±19.709	394.813±13.871	8.403±0.281	5.914±0.133
Autoformer	timeF	6.681±0.094	5.040±0.081	608.499±9.051	384.782±9.361	8.180±0.020	**5.831±0.017**
	fixed	6.438±0.064	4.909±0.064	588.466±9.446	375.703±8.107	8.239±0.053	5.898±0.025
	learned	6.812±0.091	5.200±0.072	593.071±3.476	393.695±2.385	8.392±0.220	6.044±0.158
FEDformer	timeF	6.567±0.158	5.015±0.130	633.060±7.646	401.925±7.186	8.314±0.081	5.941±0.055
	fixed	**6.358±0.050**	**4.841±0.029**	596.240±13.169	403.764±12.324	8.214±0.013	5.913±0.024
	learned	6.650±0.049	5.108±0.036	539.039±2.878	387.422±1.611	8.374±0.051	6.049±0.049

$\{t_1\}$ to $\{t_{\text{obs}}\}$ in Table I) in PromptCast. Based on the implementations of Informer[8] and Autoformer[9], we fully investigate and benchmark three different embedding approaches, namely, *timeF*, *fixed*, and *learned*. For the three different temporal embeddings used in numerical methods benchmarking, we also follow the Autoformer implementations[10]. Basically, the *timeF* embedding is accomplished via *nn.Linear()* function and the *fixed* and *learned* are based on *nn.Embedding()* function. The *fixed* embedding has fixed non-trainable parameters (similar to the original sin/cos position embedding weight calculation in the vanilla Transformer) for the *nn.Embedding()* layer, whereas the parameters for the *learned* embedding are trainable. In this way, the three different embedding methods can be used for numerical forecasting methods to capture the temporal information in the input data and this offers a comprehensive evaluation of the temporal embeddings for the forecasting task.

Table III presents the performance of different methods with different temporal embedding policies. In general, FEDformer, Informer and Autoformer achieve the best performance across different sub-sets. On most of cases, these advanced time series forecasting frameworks outperform the vanilla Transformer, naive methods, and non-Transformer methods. Naive methods demonstrate worse forecasting performance compared to other methods, which is as expected. When comparing different embeddings, the *fixed* embedding demonstrates overall good performance. This embedding leads to good predictions on 5 out of 6 metrics and the *timeF* is the best performer of the remaining metric (MAE on SG). The *learned* embedding has the worst performance on CT and SG, but it beats the *timeF* on the ECL sub-set. These results suggest that the *fixed* embedding is a favorable approach for incorporating temporal cues.

2) Pre-trained Language Models: For language models investigated in the benchmark, the ready-to-use pre-trained weights provided by HuggingFace [25] are used for initialization. The configuration details are listed in Table VI and with the model key given in the table, the corresponding pre-trained model can be accessed and downloaded from HuggingFace. It is worth noting that the pre-trained weights are trained with general English-language corpora datasets such as BookCorpus [26], CC-News [18], and OpenWebText [27]. These common language datasets are about general articles and do not include specific time series sequences orientated data. Although three data sources in PISA are open available, only the original numerical data (in the csv format) can be acquired online instead of the text format with our prompts. This also guarantees that the data from these three datasets are not used in pre-training and prevents the potential data leakage. In the experiments, each language model is fine-tuned with the training set of each sub-set in PISA.

The prediction results (RMSE and MAE) of using different language generation models on PISA dataset are listed in Table IV. According to the table, the top performers (shown in bold) include Bigbird, Bart, and RoBERTa. Bigbird achieves the best performance on 4 out of 6 metrics. When we jointly consider Table III and Table IV, it can be seen that using language models perform reasonably well on the CT and ECL sub-sets. For ECL, although the MAE of using language models is slightly worse than the best performer in Table III, the RMSE has a relatively large improvement. Compared against numerical methods, using language models can also yield comparable results on SG. This benchmark answers RQ1 and indicates that prompt-based forecasting with language models is a promising direction for the time series forecasting research.

Table V reports the Missing Rate metric. It is clearly noticeable that only three methods (ProphetNet, Electra, and

---

[8]https://github.com/zhouhaoyi/Informer2020/blob/main/models/embed.py

[9]https://github.com/thuml/Autoformer/blob/main/layers/Embed.py

[10]https://github.com/thuml/Autoformer/blob/main/layers/Embed.py

TABLE IV
RESULTS (RMSE AND MAE) OF USING LANGUAGE MODELS FOR PROMPTCAST ON PISA-PROMPT.

	CT				ECL				SG			
	RMSE		MAE		RMSE		MAE		RMSE		MAE	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
T5	6.499	0.065	4.830	0.038	527.425	10.280	353.450	2.696	8.450	0.037	5.879	0.020
Bart	6.432	0.040	4.759	0.027	527.350	10.608	355.390	2.751	8.279	0.053	**5.785**	0.023
Blenderbot	6.667	0.048	4.828	0.025	541.713	10.838	355.846	4.154	8.429	0.080	5.798	0.022
LED	6.376	0.036	4.730	0.025	540.924	16.542	367.276	6.742	8.277	0.072	5.787	0.036
Pegasus	6.379	0.023	4.727	0.014	537.186	11.296	361.135	4.728	8.289	0.016	5.817	0.013
ProphetNet	6.375	0.063	4.740	0.052	584.814	4.124	356.632	2.712	8.466	0.135	5.847	0.071
Bigbird	**6.351**	0.016	**4.707**	0.019	**519.665**	3.440	**350.699**	1.953	8.326	0.048	5.841	0.031
Electra	6.397	0.011	4.740	0.013	576.506	3.789	352.187	3.413	8.311	0.084	5.820	0.046
BERT	6.388	0.081	4.758	0.052	577.076	3.608	354.653	2.169	8.395	0.040	5.823	0.030
RoBERTa	6.450	0.081	4.786	0.070	659.874	23.218	448.902	19.320	**8.260**	0.031	**5.785**	0.009

TABLE V
THE MISSING RATE PERFORMANCE OF LANGUAGE MODELS.

Missing Rate	ProphetNet	Electra	BERT
on CT (%)	0.412 ± 0.045	0.319 ± 0.068	0.244 ± 0.151

TABLE VI
DETAILS OF HUGGINGFACE PRE-TRAINED MODELS

Model	HuggingFace Key	Pretrained Model Size
T5	t5-base	891.7 MB
Bart	facebook/bart-base	557.8 MB
Blenderbot	facebook/blenderbot_small-90M	350.4 MB
LED	allenai/led-base-16384	647.7 MB
Pegasus	google/pegasus-xsum	2.3 GB
ProphetNet	microsoft/prophetnet-large-uncased	1.6 GB
Bigbird	google/bigbird-pegasus-large-arxiv	2.3 GB
Electra	google/electra-base-generator	135.0 MB
BERT	bert-base-uncased	440.5 MB
RoBERTa	roberta-base	501.2 MB

BERT) have a tiny amount (less than 0.5%) of missing cases and all cases appear on the CT sub-set. For other methods that are not listed in Table V , the missing rates are all 0. We further investigate the output sentences that cannot be decoded and find out that the failure cases are related and potentially caused by negative values. For example, the failure generated sentences are like *the temperature will be - - - -* where the models fail to generate the tokens after "-". Our PISA dataset is valuable in supporting the research directions to address this limitation in the future. For more qualitative examples, we provide all generated examples of language models in our GitHub repository[11] including all the three test sets with various settings such as with pre-trained weights, train-from-scratch, zero-shot, prompt ablations, and multi-step forecasting.

E. Training From Scratch

Another question we aim to explore is whether the language models in PromptCast can still be used for time series forecasting without the use of pre-trained weights. Hence, we disable the pre-trained weights and train language models from scratch

[11]https://github.com/HaoUNSW/PISA/PromptCast%20Generated%20Examples

with the training set of PISA dataset. We select three language models that exhibit excellent forecasting performance on all three subsets (according to Table IV) for examination in this part of the experiment: Bart, Pegasus, and Bigbird. The results are reported in the middle section of Table VII. Comparing the results of using pre-training weights (Table IV), we note that there is a performance decrease for each method when pre-trained weights are not loaded, suggesting that utilizing pre-trained weights could result in further performance improvement. However, even training from scratch, the language generation models (especially Bart) still can yield comparable prediction results compared to numerical-based methods (Table III). This demonstrates that our proposed PromptCast is robust and not entirely reliant on pre-trained weights.

F. Zero-shot Performance

To further explore the language models for prompt-based forecasting, we conduct an experiment under the zero-shot setting (see Table VII). Specifically, we fine-tune each method on two sub-sets and test the fine-tuned model on the test set of the left sub-set (*e.g.*, fine-tune with the training sets of CT and ECL, test on the test set of SG). For the comparison purpose, we also evaluate the Transformer-based numerical methods under the same zero-shot setting and the results are given in Table VII (the upper part). Similar to the results given in Table III (the normal training setting), the *fixed* embedding has better performance (5 out of 6 metrics) than the other two embeddings under this challenging zero-shot setting. Since the date range of the three sub-sets are dissimilar (Table II), learnable *learned* and *timeF* temporal embeddings would lead to inferior performance when transferred to unseen scenarios as the learned embeddings may not be appropriate for the new scenarios.

Except for the Autoformer and FEDformer on ECL, the numerical-based methods fail to produce satisfactory predictions under the zero-shot setting. Given the distinct characteristics of the three subsets, such poor performance is as expected for numerical methods. However, for the language models, although the performance is lower than the standard setting and the train-from-scratch setting, prompt-based forecasting can still generate reasonable predictions. This demonstrates a strong generalization ability when time series forecasting is

TABLE VII
RESULTS OF NUMERICAL FORECASTING METHODS AND LANGUAGE MODELS UNDER DIFFERENT SETTINGS.

Method	Temporal Embedding	CT				ECL				SG			
		RMSE		MAE		RMSE		MAE		RMSE		MAE	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Transformer	timeF	75.465	1.330	73.238	1.473	11866.762	40.561	11288.860	41.504	29.010	2.554	18.903	1.087
	fixed	67.964	12.021	65.991	12.311	5780.931	1432.223	5055.838	1836.453	52.461	17.611	47.150	21.680
	learned	48.691	14.586	40.968	17.008	7938.621	550.239	6982.758	647.932	28.238	1.348	18.719	1.743
Informer	timeF	67.783	15.014	64.901	16.422	11887.368	30.596	11306.690	32.765	34.927	3.421	25.205	3.983
	fixed	69.109	8.656	67.065	9.090	11180.022	296.532	10649.465	259.677	26.761	2.290	15.930	1.857
	learned	45.517	17.482	38.000	17.228	11509.084	113.513	10923.215	114.072	27.417	2.241	17.310	1.471
Autoformer	timeF	52.814	5.002	39.577	5.842	694.693	2.715	455.658	2.188	38.710	11.207	30.857	9.751
	fixed	47.691	5.329	34.531	2.996	674.641	1.845	440.564	1.678	36.801	3.523	28.637	1.927
	learned	83.349	9.332	59.951	7.855	693.810	0.719	454.691	0.644	56.787	3.050	40.890	2.004
FEDformer	timeF	63.851	4.729	46.117	4.608	693.017	2.127	454.284	1.983	50.252	8.780	40.091	8.115
	fixed	77.699	3.711	54.176	4.005	655.196	3.142	424.823	2.603	64.622	5.056	45.391	2.996
	learned	239.426	24.961	146.535	21.858	694.019	0.832	454.866	0.842	108.169	8.851	85.243	6.055
Train-from-Scratch PromptCast Setting													
Bart	N/A	6.886	0.052	5.130	0.039	**546.881**	8.904	**371.102**	4.369	**8.923**	0.466	**6.000**	0.174
Pegasus		6.791[a]	0.190	5.043	0.114	632.351	14.745	411.915	6.316	9.484	0.372	6.180	0.078
Bigbird		**6.643**	0.076	**4.964**	0.085	639.889	8.340	416.529	4.021	10.529	0.437	6.365	0.031
Zero-Shot PromptCast Setting													
Bart	N/A	7.379	0.086	5.501	0.067	660.082	16.205	493.035	18.166	**8.592**	0.075	**5.961**	0.038
Pegasus		**6.918**	0.022	**5.178**	0.031	**643.483**	16.536	446.876	5.822	9.293	0.160	6.116	0.041
Bigbird		7.070	0.074	5.248	0.044	665.191	55.176	**417.634**	4.815	9.439	0.020	6.289	0.027

[a] Pegasus Missing Rate (%) on CT: 2.482±3.754

addressed with prompts (RQ2). This strong zero-shot ability could bring advantages in real-world forecasting applications such as rapid deployment for new forecasting scenarios and cold-start forecasting for scenarios without any historical data. In the future, prompt-based forecasting could also enable the exploration of more complex forecasting scenarios such as predicting energy consumption based on weather temperature or forecasting customer traffic with temperature trends.

G. Prompts Ablation Study

To further investigate the prompts in the PromptCast task, we conduct an ablation study on our prompting templates and two simplified prompts are developed as follows.

- Basic Prompt: as shown in Table VIII, we remove auxiliary information (*e.g.*, date information) and only keep the core information (*i.e.*, the historical observation values $x^m_{t_1:t_{obs}}$) in the input prompts. The output prompts remain the same.
- Minimum Prompt: This is the simplest and the most straightforward version of prompts. We use commas to convert the sequential numerical values $x^m_{t_1:t_{obs}}$ into comma-delimited strings as the input prompts (*e.g.*, ``78, 81, 83, 84, 84, 82, 83, 78, 77, 77, 74, 77, 78, 73, 76''). For the output prompts, the prediction targets $x^m_{t_{obs+1}}$ are directly used as single-word strings (*e.g.*, ``78'').

The prediction results of using these two types of prompts on our PISA dataset are presented in Table IX. We can notice that both two simplified prompts lead to worse performance compared to the default template reported in the main paper (Table I). While the minimum prompts can produce acceptable outcomes, the basic prompts have been found to have superior performance compared to the minimum prompts with the same language model. These comparison results demonstrate that: (1) including proper contexts (even if these contexts introduce no extra data, *e.g.*, the basic prompt) in the prompt is beneficial; and (2) the auxiliary information is a significant component in the prompt for better prediction performance.

H. Multi-step Forecasting

The proposed language foundation models-based Prompt-Cast forecasting paradigm is also suitable for multi-step forecasting scenarios. For the multi-step forecasting (*i.e.*, larger prediction horizon n), the question part of the input prompt template can be updated to "how many people will visit POI in the next Y days?" and the output prompt is also needed to be revised to reflect the multi-step future data values such as "For day X to day Y, there will be A, B, C, … visitors." Here is an example of predicting the next 7-time steps ($n = 7$) with the SG sub-set: Input Prompt "*From April 24, 2021, Saturday to May 08, 2021, Saturday, there were 6, 8, 6, 15, 12, 4, 13, 7, 8, 12, 16, 9, 11, 18, 10 people visiting POI 1 on each day. How many people will visit POI 1 in the next 7 days?*" and Output Prompt "*From May 09, 2021, Sunday to May 15, 2021, Saturday, there will be 11, 11, 8, 13, 10, 18, 19 visitors.*" Building on these modified prompt, we further examine the performance of three

		Template
CT	Input	The average temperature of was $\{x^m_{t_1:t_{obs}}\}$ degree on each day. What is the temperature going to be on tomorrow?
	Output	The temperature will be $\{x^m_{t_{obs+1}}\}$ degree.
ECL	Input	The client consumed $\{x^m_{t_1:t_{obs}}\}$ kWh of electricity on each day. What is the consumption going to be on tomorrow?
	Output	This client will consume $\{x^m_{t_{obs+1}}\}$ kWh of electricity.
SG	Input	There were $\{x^m_{t_1:t_{obs}}\}$ people visiting the POI on each day. How many people will visit the POI on tomorrow?
	Output	There will be $\{x^m_{t_{obs+1}}\}$ visitors.

language models (Bart, Bigbird, Pegasus) when using varying observation lengths (input lengths) and prediction horizons (output lengths) under the proposed PromptCast forecasting paradigm. Specifically, we test the models using observation length settings of 7 days and 15 days, and prediction horizons of 1 day, 4 days, and 7 days, resulting in a total of six length configuration combinations. The performance of these three language models on these multi-step forecasting settings are listed in Table X.

From the table, we can see that PromptCast can be easily adapted to multi-step forecasting while using the exact same forecasting model architectures (*i.e.*, the language foundation models) for different settings. Generally, compared to the single-step forecasting performance, increasing the prediction horizon leads to larger missing rates. Specifically, Bart shows a worse performance as evidenced by the large missing rates on different sub-sets. However, Bigbird and Pegasus can still yield reasonably good predictions with extreme small missing rates, which demonstrates language models are still powerful in predicting multiple steps ahead. The results further confirm that our proposed PromptCast is suitable for different length settings. We can observed that compared to the single-step forecasting performance, increasing the prediction horizon leads to larger missing rates. However, Bigbird and Pegasus can still yield reasonably good predictions with extreme small missing rates, which demonstrates language models are still powerful in predicting multiple steps ahead. Additionally, we notice that using a 15-day observation period often leads to better performance when predicting the same future length. This is likely because a 15-day period encompasses weekly patterns in the observation, leading to more accurate forecasting.

To provide further insight into the performance of the multi-step forecasting, additional examples of predictions for the next 7 days setting are available in our repository[12]. Our explo-

ration of PromptCast on different length settings illustrates that the model is robust to dynamic observation/prediction lengths, and presents a promising research direction for time series forecasting.

I. Multivariate Time Series Forecasting

As the first attempt to leverage language models for time series forecasting, we take the basic forecasting setting to demonstrate the concept of PromptCast. However, we would like to emphasize that the proposed new paradigm is flexible and suitable for other forecasting settings such as the multivariate time series forecasting setting. We only need to update the prompt templates accordingly when PromptCast is adapted to multivariate time series forecasting setting. Under the new setting, although input/output prompts might be updated, the core prediction model (*i.e.*, language foundation models) in PromptCastcan remain the exact same (prompt-agnostic) and same pre-trained weights can also be used. For example, in the experiments under Section IV-G, different prompts are applied. But we can also directly use the same models (without any changes to the model structures) to yield predictions. For conventional numerical forecasting methods, however, necessary updates on the deep learning model (*e.g.*, update the encoder part to fit multivariate features) must be introduced when the forecasting setting is changed. Additionally, processes such as extra hyperparameter search are often required when the model structure is modified. This could also reflect the "code less" benefits and robustness of the proposed PromptCast paradigm.

To investigate the capability of the proposed PromptCast on multivariate forecasting, we further conduct a pilot study of multivariate PromptCast on Beijing Air Quality Data[13] in addition to the three sub-sets in the main PISA dataset. The data collection period of this Air Quality (AQ) set is from 2013/03/01 to 2017/02/28. Similar to the main PISA, we split this period into training set (2013/03/01-2015/12/18), validation set (2015/12/19-2016/05/12), and test set (2016/05/13-2017/02/28). After filtering missing values, we focus on three types of air pollutants: PM2.5, PM10, and SO2, which means the feature dimension is 3.

To apply PromptCast for multivariate forecasting, we update the prompt template to include multiple features as Table XI. In total, we design and examine three different prompts. For Prompt A and B, the input prompt consists of the description of all 3 features and the main difference of these two prompts is in the output prompt part. Output Prompt A jointly describes the three features, whereas the three feature values are given separately in three short sentences in Output Prompt B. Prompt C can be seen as an extended version of the prompt template (*e.g.*, Table I) for the univariate forecasting. It decomposes the multivariate features (*i.e.*, three pollutants in this AQ dataset) into several univariate features.

The results of PromptCast with language foundation models using different prompts are reported in Table XII. Additionally, we list the performance of three naive forecasting methods as baselines. From the table, we can observe that: (1) All

TABLE IX
RESULTS OF LANGUAGE MODELS WITH DIFFERENT PROMPT TYPES.

Prompt	Model	CT				ECL				SG			
		RMSE		MAE		RMSE		MAE		RMSE		MAE	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Basic	Bart	6.512	0.016	4.790	0.012	603.571	2.461	380.583	1.695	8.677	0.055	5.912	0.013
	Pegasus	6.496	0.012	4.767	0.007	**595.942**	2.246	**366.677**	1.373	**8.530**	0.032	**5.879**	0.010
	Bigbird	**6.478**	0.033	**4.752**	0.019	609.315	3.071	378.100	2.112	8.576	0.035	5.922	0.009
Minimum	Bart	6.564	0.082	4.830	0.083	612.487	3.660	382.931	2.509	8.686	0.117	5.937	0.021
	Pegasus	6.560	0.049	4.818	0.027	**597.459**	3.763	**368.174**	1.178	**8.632**	0.046	**5.915**	0.016
	Bigbird	**6.496**	0.042	**4.766**	0.026	618.907	4.422	384.327	0.861	8.774	0.065	5.950	0.015

TABLE X
RESULTS OF LANGUAGE MODELS WITH DIFFERENT LENGTH SETTINGS.

t_{obs}	n		CT RMSE	MAE	Missing Rate
7 Days	1 Day	Bart	6.762±0.090	5.017±0.083	0
		BigBird	6.425±0.024	4.716±0.013	0
		Pegasus	6.820±0.044	5.088±0.031	0
7 Days	4 Days	Bart	10.087±0.064	7.478±0.058	1.622%±0.721%
		BigBird	9.213±0.117	6.880±0.073	0
		Pegasus	9.422±0.080	7.003±0.047	0
7 Days	7 Days	Bart	11.681±0.183	8.620±0.058	22.745%±1.926%
		BigBird	10.200±0.120	7.601±0.083	0.036%±0.018%
		Pegasus	10.144±0.188	7.526±0.131	0.022%±0.014%
15 Days	1 Day	Bart	6.432±0.040	4.759±0.027	0
		Bigbird	6.351±0.016	4.707±0.019	0
		Pegasus	6.379±0.023	4.727±0.014	0
15 Days	4 Days	Bart	10.135±0.043	7.631±0.033	1.799%±1.651%
		BigBird	9.432±0.136	7.075±0.096	0
		Pegasus	9.582±0.118	7.166±0.085	0
15 Days	7 Days	Bart	12.033±0.605	8.773±0.109	27.107%±1.159%
		Bigbird	10.329±0.162	7.745±0.120	0.023%±0.031%
		Pegasus	10.522±0.187	7.899±0.142	0.179%±0.265%

t_{obs}	n		ECL RMSE	MAE	Missing Rate
7 Days	1 Day	Bart	527.772±7.282	360.066±3.688	0
		BigBird	530.929±4.792	365.122±2.184	0
		Pegasus	527.735±5.921	360.702±2.352	0
7 Days	4 Days	Bart	1075.721±218.227	542.006±16.379	6.791%±2.949%
		BigBird	750.708±28.990	493.466±9.704	0.047%±0.015%
		Pegasus	741.931±9.613	482.426±4.749	0.120%±0.116%
7 Days	7 Days	Bart	1332.287±266.259	612.469±32.104	29.310%±20.472%
		BigBird	946.462±57.607	551.675±6.601	0.537%±0.314%
		Pegasus	851.768±8.422	545.536±2.627	3.669%±1.086%
15 Days	1 Day	Bart	527.350±10.608	355.390±2.751	0
		Bigbird	519.665±3.440	350.699±1.953	0
		Pegasus	537.186±11.296	361.135±4.728	0
15 Days	4 Days	Bart	855.906±60.640	528.250±15.276	25.701%±3.393%
		BigBird	700.280±14.048	481.145±7.026	0.041%±0.032%
		Pegasus	769.008±17.279	476.013±3.494	0.096%±0.112%
15 Days	7 Days	Bart	1190.564±111.116	611.117±55.916	50.259%±28.139%
		Bigbird	1063.650±222.601	541.538±7.397	0.329%±0.190%
		Pegasus	835.386±23.031	525.537±6.067	1.422%±0.465%

t_{obs}	n		SG RMSE	MAE	Missing Rate
7 Days	1 Day	Bart	8.427±0.016	5.909±0.023	0
		BigBird	8.526±0.052	5.962±0.026	0
		Pegasus	8.406±0.063	5.957±0.028	0
7 Days	4 Days	Bart	10.156±0.164	6.982±0.101	2.655%±1.285%
		BigBird	9.619±0.824	6.654±0.495	0
		Pegasus	8.813±0.039	6.142±0.005	0
7 Days	7 Days	Bart	12.453±0.894	8.375±0.169	32.716%±5.301%
		BigBird	10.851±0.612	7.433±0.410	0.034%±0.026%
		Pegasus	8.926±0.017	6.247±0.011	0.013%±0.005%
15 Days	1 Day	Bart	8.279±0.053	5.785±0.023	0
		Bigbird	8.326±0.048	5.841±0.031	0
		Pegasus	8.289±0.016	5.817±0.013	0
15 Days	4 Days	Bart	9.476±0.079	6.559±0.029	1.212%±0.325%
		BigBird	9.392±0.571	6.436±0.313	0
		Pegasus	8.743±0.062	6.082±0.026	0
15 Days	7 Days	Bart	11.048±0.325	7.653±0.129	38.328%±5.362%
		Bigbird	9.731±0.427	6.658±0.200	0.007%±0.011%
		Pegasus	8.802±0.031	6.139±0.016	0.011%±0.006%

three language models can generate plausible predicted future descriptions as evidenced by almost all zero missing rates. (2) The design of prompts could definitely affect the prediction performance. Specifically, Prompt C leads to better performance than Prompt A and B on this multivariate forecasting dataset. (3) The naive historical average forecasting outperforms PromptCast by a small margin. However, using Prompt C still shows a comparable performance. We think the potential reason of why PromptCast is not the best performer in this comparison is because that the current prompts cannot fully characterize the relation of different features. This also explains why Prompt C outperforms the other two prompts.

Open Question and Future Work. The results show that the model architectures used for univariate time series can also be applied to multivariate time series, this is an indication that the PromptCast task is a promising direction for multivariate time series forecasting. However, further research is needed to fully understand the behavior of multivariate time series forecasting with PromptCast and to develop more robust models that can handle this type of data. The above performance analysis points out an open question for PromptCast, that is, how to design prompts for the more challenging multivariate time series. One potential working direction is to learn the internal correlations between different temporal features and then develop prompts to represent the learned correlations. In the future, we will focus on this direction and also explore techniques such as learnable prompts for the multivariate PromptCast. We hope the proposed PISA dataset and the corresponding benchmark could encourage other researchers as well to investigate this interesting PromptCast topic.

V. CASE STUDIES

Why Language Model Works for Forecasting. Through the above experiments, we have shown that the proposed PromptCast task represents a new paradigm for time-series prediction. In this section, we aim to further investigate the reasons behind why language models can be used for forecasting purposes. First, it is worth noting that the model architectures of current deep learning-based forecasting models have been heavily influenced by the development of NLP models, such as RNN-based, LSTM with attention mechanisms, and the recent Transformer-based sequence modeling. Furthermore, when viewed from a different perspective, time series prediction can be seen as similar to the problem of predicting the next sentence in NLP, which has been effectively tackled using pre-trained foundation models. This suggests that the model architectures of language foundation models should also be adaptable for time series sequences. Second, using auxiliary information such as *time-of-day*, *day-of-week* [9] and semantic information [28] has been shown to be beneficial for improving forecasting performance. However, incorporating such semantic contexts in conventional numerical-based forecasting

TABLE XI
EXAMPLES OF TWO TYPES OF PROMPTS UNDER THE MULTIVARIATE TIME SERIES SETTING.

Input Prompt A&B	From May 13, 2016, Friday to May 27, 2016, Friday, the PM2.5 was 32, 23, 20, 53, 82, 113, 133, 94, 64, 83, 20, 4, 18, 48, 57; PM10 was 32, 40, 57, 87, 90, 113, 133, 94, 64, 83, 20, 20, 18, 48, 103; and SO2 was 2, 3, 4, 11, 29, 27, 26, 33, 13, 14, 6, 2, 2, 6, 16 on each day. What are the pollutant values going to be on May 28, 2016, Saturday?
Output Prompt A	The pollutant values will be 3, 23, 2.
Output Prompt B	The PM2.5 will be 3. The PM10 will be 23. The SO2 will be 2.
Input Prompt C	From May 13, 2016, Friday to May 27, 2016, Friday, the PM2.5 was 32, 23, 20, 53, 82, 113, 133, 94, 64, 83, 20, 4, 18, 48, 57 on each day. What is the PM2.5 value going to be on May 28, 2016, Saturday? From May 13, 2016, Friday to May 27, 2016, Friday, the PM10 was 32, 40, 57, 87, 90, 113, 133, 94, 64, 83, 20, 20, 18, 48, 103 on each day. What is the PM10 value going to be on May 28, 2016, Saturday? From May 13, 2016, Friday to May 27, 2016, Friday, the SO2 was 2, 3, 4, 11, 29, 27, 26, 33, 13, 14, 6, 2, 2, 6, 16 on each day. What is the SO2 value going to be on May 28, 2016, Saturday?
Output Prompt C	The PM2.5 will be 3. The PM10 will be 23. The SO2 will be 2.

TABLE XII
RESULTS OF LANGUAGE MODELS ON MULTIVARIATE TIME SERIES FORECASTING.

Prompt			RMSE	MAE	MissingRate
N/A	CY		73.058	43.274	N/A
	HA		63.289	40.772	N/A
	CLW		87.250	52.796	N/A
Prompt A	Bart		79.218±4.948	46.553±2.838	0
	BigBird		75.783±1.988	45.590±2.105	2.681%±3.471%
	Pegasus		82.859±0.588	48.168±0.444	0
Prompt B	Bart		75.268±2.150	41.869±0.915	0
	BigBird		81.510±2.524	48.913±0.639	0
	Pegasus		75.706±1.736	45.206±1.156	0
Prompt C	Bart		68.282±2.345	40.690±1.131	0
	BigBird		67.729±1.880	41.547±1.071	0
	Pegasus		67.834±0.859	39.893±0.691	0

methods often requires additional efforts to explicitly design special layers or modules in a heuristic way for the auxiliary inputs. Using natural language prompts, on the other hand, is a more direct way to take semantic information into account in the forecasting process, which presents a new perspective for the forecasting task. Even though some approaches (such as the temporal embedding methods discussed above) have been proposed for numerical-based methods to incorporate auxiliary information, it remains challenging for a forecasting model to seamlessly model the contexts and the main temporal information. In contrast, with PromptCast, both auxiliary and core temporal inputs are considered together as tokens after prompting. The intra-relation of numerical value tokens at different time steps and the inter-relation between numerical value tokens and contextual information tokens (*e.g.*, date information) could be better learned simultaneously by language foundation models (*e.g.*, through the self-attentions in Transformers). Modeling these relations would then result in good forecasting performance.

To validate the above hypothesis, we conduct two case studies and visualize the attentions (between the input sentence

and the output sentence) learned by the language models (using the Bart model as an example in these case studies).

- *Case Study 1*
  - **Input**: From September 06, 2019, Friday to September 20, 2019, Friday, the average temperature of region 1 was 65, 66, 70, 71, 71, 77, 78, 65, 70, 76, 74, 70, 64, 61, 64 degree on each day. What is the temperature going to be on September 21, 2019, Saturday?
  - **Predicted**: The temperature will be 68 degree.
  - **Ground Truth**: The temperature will be 71 degree.
- *Case Study 2*
  - **Input**: From June 08, 2021, Tuesday to June 22, 2021, Tuesday, there were 13, 16, 9, 11, 21, 9, 13, 15, 8, 24, 9, 10, 11, 10, 14 people visiting POI 1 on each day. How many people will visit POI 1 on June 23, 2021, Wednesday?
  - **Predicted**: There will be 11 visitors.
  - **Ground Truth**: There will be 12 visitors.

Specifically, in Figure 3, we show the attention weights between the numerical predicted values in the generated outputs and all the tokens in the input prompts. For each heatmap plot, with hotter regions representing larger attention values, the horizontal axis represents the input prompt (in token format) and the vertical axis represents different attention heads.

For Case 1 (Figure 3 (a)), we can clearly observe that head 6 pays more attention to the numerical sequential data in the input prompt while the other heads have higher attention to other semantic auxiliary information such as `September` and `Friday`. Moreover, head 7 shows attentions to both the input numerical historical observations (*e.g.*, `76`, `74`, and `64`) and semantic tokens simultaneously. For Case 2, the attention visualization of two layers are displayed in Figure 3 (b) and Figure 3 (c). It can be noticed that the learned attentions between the predicted value and the numerical historical values in the input prompt are mainly in layer 5 (Figure 3 (c)) of the Bart model. Compared to layer 5, layer 4 (Figure 3 (b)) focuses on not only the numerical tokens but also contextual

(a) Case 1: CT sub-set.

(b) Case 2 (layer 4): SG sub-set.

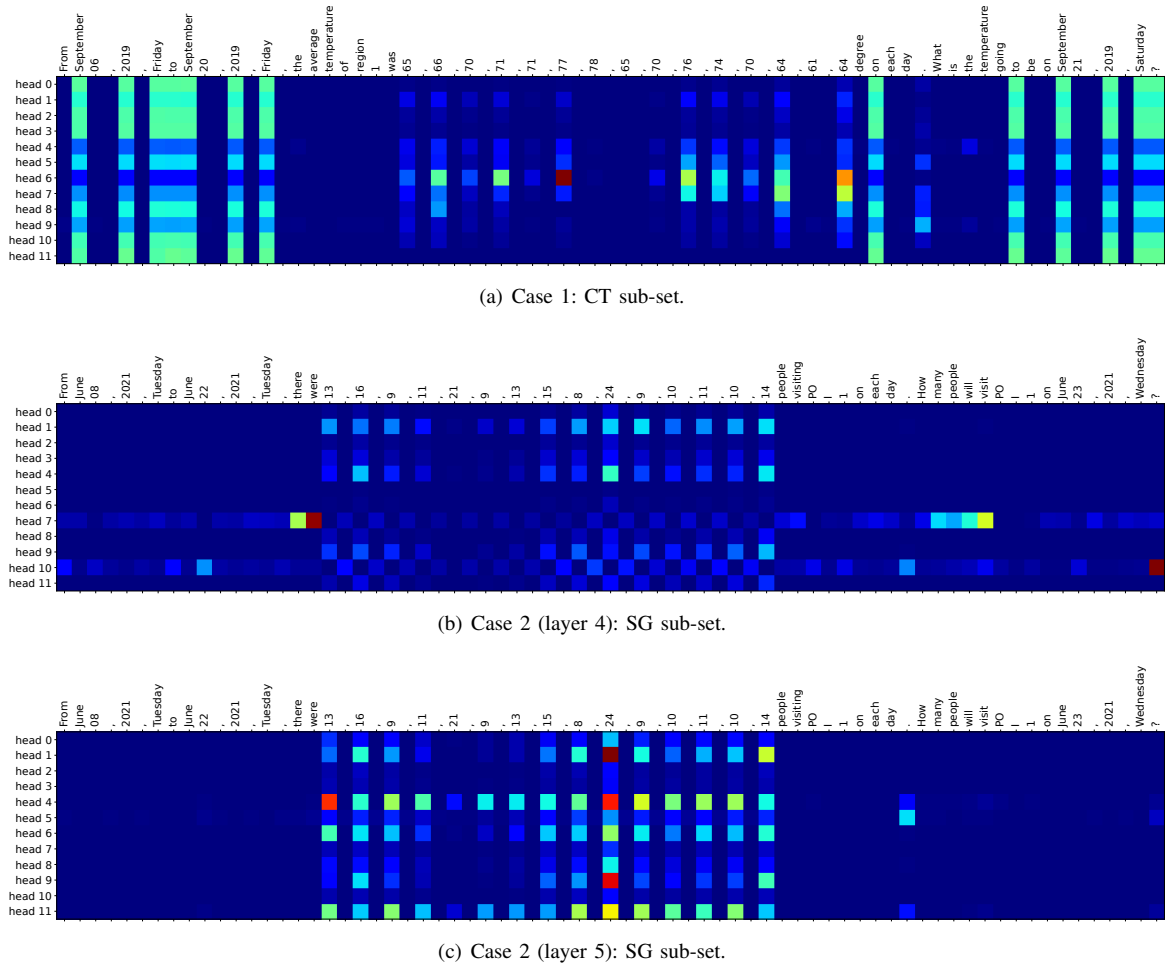(c) Case 2 (layer 5): SG sub-set.

Fig. 3. Visualizations of attentions in language models under the proposed PromptCast setting.

tokens. More specifically, head 7 demonstrates large attention weights to semantic tokens like `there were` and `How many people will visit` whereas head 10 highlights the auxiliary date token `22` as well as the punctuation token `?`. These case studies demonstrate that the attention learned by language models is able to take into account both the numerical value tokens (historical values) and auxiliary information tokens to make predictions under the PromptCast paradigm. It further justifies the rational of leveraging language models for forecasting time series.

VI. DISCUSSION AND CONCLUSION

In this paper, we introduce a new task, PromptCast, which utilizes language models to predict time series in a language generation manner. As this is the first work on the PromptCast task, and no existing datasets are suitable, we construct the first dataset, PISA, to investigate prompt-based forecasting. This large-scale dataset contains three real-world time series forecasting scenarios. To further advance the research of PromptCast, we also establish a benchmark on the released dataset and provide a set of strong baselines that include both state-of-the-art numerical forecasting methods and language generation models. The experimental results demonstrate that

using language models in the PromptCast setting results in good forecasting performance and generalization ability.

Broader Impact. We believe that the findings of this study will provide forward-thinking concepts and fresh insights for researchers working in the field of time series forecasting. We also think that the proposed PromptCast task, as well as the PISA dataset, could open new related research directions and provide visionary ideas about downstream applications that could be enabled by this work. Some potential directions for future research are discussed below. (1) *Automatic Prompting*: In this paper, the transformation of numerical data to text is achieved through the use of templates. Although template-based prompting is efficient, it can be difficult to produce diverse prompts and fixed templates may introduce biases (biases towards certain templates). To address these limitations, a potential research direction is the development of automatic time series prompting or time series captioning (similar to image captioning [29]), which utilizes generative models to describe time series data. (2) *Explainable PromptCast*: Another area that has yet to be fully investigated is the question of why models designed for language modeling tasks are able to predict time series. Recent research has begun to explore the use of language models for non-language downstream tasks [30]. Although we have initially discussed our thoughts on why

language models can be effective for time series forecasting under the PromptCast task through case studies, further studies on the interpretability and explainability of PromptCast models would be an interesting and valuable research direction to better understand the underlying mechanisms. This can help us to design better models and understand the limitations of current models. By understanding the interpretability, we can also have a better understanding of how to apply these models to real-world problems. (3) *PromptCast QA and Chatbots*: The research of PromptCast task would trigger and promote the development of time series forecasting question answering tasks and building chatbot applications with forecasting capabilities. Note that the PromptCast QA task differs from recent tasks such as TimeQA [31], which is proposed to answer general time-related questions based on Wikipedia text, and ForecastQA [32], which is also based on text articles. The core of the PromptCast QA task is the question-answering ability for forecasting based on given sequential numerical value contexts. This task can be applied to digital assistants such as Siri.
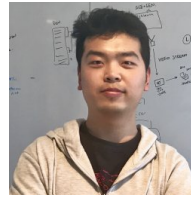
Limitations and Future Work. As the first attempt to create a dataset for the novel PromptCast task, in this dataset release, we have mainly focused on the univariate time series forecasting setting. As discussed in the last part of Section IV-I, one potential direction for our future work is to extend the dataset to include multivariate time series, which would support the research of more complex time series forecasting scenarios. Another future work is to develop an auto-regressive method to further enhance the PromptCast performance on multi-step forecasting setting. For example, the language models could be fine-tuned to predict the value at the next time step. The generated sentence will then be recurrently appended to the input prompts to predict longer horizons. Additionally, more challenging heterogeneous datasets that incorporate multiple data sources will also be considered for the PromptCast task in the future. The prompt-based paradigm can also be extended to other tasks such as time series classification and anomaly detection. This would enable the proposed paradigm to be applied to a wider range of real-world scenarios.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[4] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[6] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 139, 2021, pp. 8748–8763.

[7] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang *et al.*, "Grounded language-image pre-training," *arXiv preprint arXiv:2112.03857*, 2021.

[8] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in neural information processing systems*, vol. 32, 2019.

[9] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of AAAI*, 2021.

[10] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *International Conference on Learning Representations*, 2021.

[11] J. Xu, J. Wang, M. Long *et al.*, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[12] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 27 268–27 286.

[13] A. Drouin, É. Marcotte, and N. Chapados, "Tactis: Transformer-attentional copulas for time series," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 5447–5493.

[14] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[15] H. Xue, F. D. Salim, Y. Ren, and C. L. Clarke, "Translating human mobility forecasting through natural language generation," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1224–1233.

[16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2020.

[17] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 2020, pp. 7871–7880.

[18] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[19] K. Clark, M. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: pre-training text encoders as discriminators rather than generators," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[20] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang *et al.*, "Big bird: Transformers for longer sequences," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 283–17 297, 2020.

[21] W. Qi, Y. Yan, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang, and M. Zhou, "Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training," in *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, ser. Findings of ACL, vol. EMNLP 2020. Association for Computational Linguistics, 2020, pp. 2401–2410.

[22] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[23] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, E. M. Smith, Y. Boureau, and J. Weston, "Recipes for building an open-domain chatbot," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*. Association for Computational Linguistics, 2021, pp. 300–325.

[24] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 328–11 339.

[25] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Oct. 2020, pp. 38–45.

[26] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27.

[27] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[28] H. Xue, F. Salim, Y. Ren, and N. Oliver, "Mobtcast: Leveraging auxiliary trajectory forecasting for human mobility prediction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30 380–30 391, 2021.

[29] S. Herdade, A. Kappeler, K. Boakye, and J. Soares, "Image captioning: Transforming objects into words," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[30] T. Dinh, Y. Zeng, R. Zhang, Z. Lin, S. Rajput, M. Gira, J.-y. Sohn, D. Papailiopoulos, and K. Lee, "Lift: Language-interfaced fine-tuning for non-language machine learning tasks," *arXiv preprint arXiv:2206.06565*, 2022.

[31] W. Chen, X. Wang, W. Y. Wang, and W. Y. Wang, "A dataset for answering time-sensitive questions," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, vol. 1, 2021.

[32] W. Jin, R. Khanna, S. Kim, D. Lee, F. Morstatter, A. Galstyan, and X. Ren, "Forecastqa: A question answering challenge for event forecasting with temporal text data," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 2021, pp. 4636–4650.

Hao Xue is a Research Fellow who currently holds a position at the School of Computer Science and Engineering at UNSW Sydney. He earned his PhD from The University of Western Australia in 2020. After completing his doctorate, he worked as a Research Fellow at the School of Computing Technologies at RMIT University from 2020 to 2022. He was awarded the DAAD AInet Fellowship in 2022. His research interests include spatiotemporal data modelling, time series forecasting, language generation based forecasting, and data-efficient time series representation learning.

Professor Flora Salim is the CISCO Chair of Digital Transport, School of Computer Science and Engineering, UNSW Sydney.Her research, on behaviour modelling, AI and machine learning on time-series and spatio-temporal sensor data, has been funded by the ARC,Humboldt Foundation,Bayer Foundation,Microsoft Research,Qatar National Research Fund,and many local and international industry partners. She won the Women in AI Awards 2022 ANZ Defence and Intelligence category.

Appendix

This Appendix document is organized as follows:

- Section A introduces the hosting and licensing of our PISA dataset.
- Section B presents the *Datasheet for Datasets* for our PISA dataset.

A. PISA Hosting and Licensing

Source Data.

- CT: The source data for the CT sub-set can be accessed through the Average Daily Temperature Archive[14]. Based on its description, this source data is available for research and non-commercial purposes only.
- ECL: The original data is available at UCI Machine Learning Repository[15]. We used a processed version of the original data provided by Informer repository which is licensed under Apache License 2.0[16].
- SG: We access the raw SafeGraph Weekly Patterns data through SafeGraph Data for Academics[17]. "*SafeGraph*[18], *a data company that aggregates anonymized location data from numerous applications in order to provide insights about physical places, via the SafeGraph Community. To enhance privacy, SafeGraph excludes census block group information if fewer than five devices visited an establishment in a month from a given census block group.*" According to the policy, it is against SafeGraph's terms of service to directly re-share raw SafeGraph data. However, it is acceptable under SafeGraph's terms of service to share aggregated and derived data, and to include data in chart and visual forms. We strongly recommend users to register SafeGraph Data for Academics before accessing our PISA dataset.

PISA. The PISA dataset and codes for models reported in the benchmark are available at https://github.com/HaoUNSW/PISA. In this repository, we also provide some generated examples of language models for the PromptCast task. It's worth noting that only the validation sets are provided as PISA examples in the above-mentioned repository during the submission period. After the acceptance decision notification, the full PISA dataset (including train/val/testing sets) will be uploaded to the same repository and made publicly available. The full dataset will also be available through HuggingFace Dataset[19], which will make it easier to use our dataset with HuggingFace models.

The PISA dataset will be distributed under CC BY-NC-SA 4.0[20].

B. PISA Datasheet

Motivation

For what purpose was the dataset created? (Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.)

The PISA dataset is created to support the research of novel PromptCast task proposed in this paper.

Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

The authors (institution: School of Computer Science and Engineering, University of New South Wales, Sydney, Australia) of this paper created this PISA dataset.

Who funded the creation of the dataset? (If there is an associated grant, please provide the name of the grant or and the grant name and number.)

The creator of the dataset was supported by Australian Research Council (ARC) Discovery Project *DP190101485*.

Any other comments?

None.

Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? (Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.)

For the CT sub-set, the instances represent the daily temperature of a city. For the ECL sub-set, the instances represent the daily electricity consumption of a user. For the SG sub-set, the instances represent the daily visitor counts of a POI.

How many instances are there in total (of each type, if appropriate)?

The proposed PISA dataset includes 311,932 instances in total. The details of the partitions are presented in Table II.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? (If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).)

For the CT sub-set, 110 international cities are randomly selected to form the dataset. For the ECL sub-set, We filtered users with missing values and randomly selected 50 users (from 321 users) with full records of the entire data collection period. For the SG sub-set, we randomly selected 324 POIs with full records. We list the data value range of each sub-set in Table II and show the data distributions in Figure 2.

What data does each instance consist of? ("Raw" data (e.g., unprocessed text or images)or features? In either case, please provide a description.)

Each instance consists of the input prompt and the output prompt. The input prompt is the input of a model and the output prompt is the desired output of the model. Our PISA provides the input prompts and the corresponding output prompts in separate text files (*e.g.*, val_x_prompt.txt and val_y_prompt.txt).

[14]https://academic.udayton.edu/kissock/http/Weather/default.htm
[15]UCI Machine Learning Repository
[16]https://github.com/zhouhaoyi/Informer2020/blob/main/LICENSE
[17]https://www.safegraph.com/academics
[18]https://www.safegraph.com/
[19]https://huggingface.co/datasets
[20]https://creativecommons.org/licenses/by-nc-sa/4.0/

Is there a label or target associated with each instance? If so, please provide a description.

Yes. As described in the above response, the output prompt is considered as the label.

Is any information missing from individual instances? (If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.)

There is no missing data (beyond what was intentionally omitted, *e.g.*, the POI geo-location of SG sub-set).

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? (If so, please describe how these relationships are made explicit.)

Since the sliding window approach is applied (see Section III-A), the observation data of different instances might belong to the same object-of-interest. However, aach instance should be considered and treated as an independent instance with no relationship to other instances in PISA.

Are there recommended data splits (e.g., training, development / validation, testing)? (If so, please provide a description of these splits, explaining the rationale behind them.)

Yes. Each sub-set in our PISA is divided into train/val/test at the ratio of 7:1:2 by the chronological order. Please refer to Section III-A and Table II for more details about the data splits.

Are there any errors, sources of noise, or redundancies in the dataset? (If so, please provide a description.)

No.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? (If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.)

The PISA dataset is self-contained.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals' non-public communications)? (If so, please provide a description.)

No.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? (If so, please describe why.)

No.

Does the dataset relate to people? (If not, you may skip the remaining questions in this section.)

No.

Does the dataset identify any subpopulations (e.g., by age, gender)? (If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.)

N/A.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? (If so, please describe how.)

N/A.

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? (If so, please provide a description.)

N/A.

Any other comments?

None.

Collection Process

How was the data associated with each instance acquired? (Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.)

The data associated with each instance is acquired and derived from three data sources: CT, ECL, and SG. We have examined and verified the data. These data sources have also been used in the literature forecasting tasks.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? (How were these mechanisms or procedures validated?)

N/A. Our PISA dataset is based on existing data sources. We did not collect the data by ourselves.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?

For the CT sub-set, 110 international cities are randomly selected to form the dataset. For the ECL sub-set, We filtered users with missing values and randomly selected 50 users (from 321 users) with full records of the entire data collection period. For the SG sub-set, we randomly selected 324 POIs with full records.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

All collection and annotation was done by the first author.

Over what timeframe was the data collected? (Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.)

The data collection period of the data sources are reported in Table II. The collection period of CT is 2017/01/01 - 2020/04/30. The collection period of ECL is 2012/01/01 - 2014/12/31. The collection period of SG is 2020/06/15 - 2021/09/05.

Were any ethical review processes conducted (e.g., by an institutional review board)? (If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.)

N/A. The original data sources are publicly available and have been widely used in the literature.

Does the dataset relate to people? (If not, you may skip the remaining questions in this section.)

No.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?

N/A.

Were the individuals in question notified about the data collection? (If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.)

N/A.

Did the individuals in question consent to the collection and use of their data? (If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.)

N/A.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? (If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).)

N/A.

Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted? (If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.)

N/A.

Any other comments?

None.

Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? (If so, please provide a description. If not, you may skip the remainder of the questions in this section.)

Yes. The details of the preprocessing/cleaning/labeling of the raw data sources are described in Section III-A. The prompting process of our PISA dataset is provided in Section III-B.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? (If so, please provide a link or other access point to the "raw" data.)

For the CT and ECL sub-sets, the raw data sources are publicly available. For the SG sub-set, due to SafeGraph policy, the raw data will not be provided. Please refer to Section A for more details.

Is the software used to preprocess/clean/label the instances available? (If so, please provide a link or other access point.)

Yes. In the GitHub repository (https://github.com/HaoUNSW/PISA), we provide the codes of transferring the source numerical data to prompts.

Any other comments?

None.

Uses

Has the dataset been used for any tasks already? (If so, please provide a description.)

No. This PISA dataset is designed and introduced for the novel PromptCast task proposed in the paper.

Is there a repository that links to any or all papers or systems that use the dataset? (If so, please provide a link or other access point.)

Considering that the proposed dataset PISA is associated with the novel task PromptCast formulated in this paper, there is no repository for the time being. In the future, we do have a plan to create such a repository to summarize the papers related to this dataset or this task.

What (other) tasks could the dataset be used for?

The PISA dataset could also be used for the conventional numerical-based time series forecasting task.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? (For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?)

No.

Are there tasks for which the dataset should not be used? (If so, please provide a description.)

No. We encourage other researchers to try our dataset on other related tasks.

Any other comments?

None.

Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description.

Yes. During the reviewing period, the dataset (*i.e.*, the validation sets as examples) is available https://github.com/HaoUNSW/PISA. The full dataset will be publicly available on the same GitHub page for download by all interested third parties for research purpose after the acceptance decision notification.

How will the dataset will be distributed (e.g., tarball on website, API, GitHub) Does the dataset have a digital object identifier (DOI)?

The dataset will be distributed through the GitHub page and HuggingFace dataset page. Please refer to Section A for more details.

When will the dataset be distributed?

The full dataset will be made publicly available after the acceptance decision notification.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.

Our PISA dataset will be distributed under the CreativeCommons Attribution-NonCommercial-ShareAlike license (CC-BY-NC-SA). The terms of this license may be found at: https://creativecommons.org/licenses/by-ncsa/2.0/legalcode.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.

No. There are no third party restrictions on the dataset.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.

No export controls or other regulatory restrictions apply to the dataset. However, we strongly recommend users to register SafeGraph Data for Academics before considering our PISA dataset. Please refer to Section A for more details.

Any other comments?

None.

Maintenance

Who will be supporting/hosting/maintaining the dataset?

The authors of this paper will support/host/maintain the proposed PISA dataset.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

The owner/curator/manager of the dataset can be contacted via: hao.xue1@unsw.edu.au.

Is there an erratum? If so, please provide a link or other access point.

No erratum for the current version of PISA dataset.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?

The dataset will be updated in the future by the authors of this paper. The updates will focus on adding new forecasting scenarios, introducing more prompt templates, or expanding PISA to multivarite time series setting. The updated will be communicated to users through the GitHub page and the HuggingFace page.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? If so, please describe these limits and explain how they will be enforced.

N/A.

Will older versions of the dataset continue to be supported/ hosted/ maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to users.

Yes, all the versions of the dataset will be supported/ hosted/ maintained by the authors of this paper. The versioning information will be communicated to users through the GitHub page and the HuggingFace dataset page.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.

If others want to extend/augment/build on/contribute to the dataset, the authors of this paper should be first contacted. Other researchers are encouraged to pull requests on the GitHub page. We will validate and verify the contributed data before merge the contributed data.

Any other comments?

None.