PROMPTCAST: A NEW PROMPT-BASED LEARNING PARADIGM FOR TIME SERIES FORECASTING

Hao Xue
School of Computer Science and Engineering
University of New South Wales, Australia
hao.xue1@unsw.edu.au

Flora D. Salim
School of Computer Science and Engineering
University of New South Wales, Australia
flora.salim@unsw.edu.au

ABSTRACT

This paper studies the time series forecasting problem from a whole new perspective. In the existing SOTA time-series representation learning methods, the forecasting models take a sequence of numerical values as input and yield numerical values as output. The existing SOTA models are largely based on Transformer architecture, modified with multiple encoding mechanisms to incorporate the context and semantics around the historical data. In this paper, we approach representation learning of time-series from the paradigm of prompt-based natural language modeling. 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. We frame the forecasting task in a sentence-to-sentence manner which makes 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 such as Bart. The benchmark results with single- and multi-step forecasting settings demonstrate that the proposed prompt-based time series forecasting with language generation models is a promising research direction. In addition, in comparison to conventional numerical-based forecasting, PromptCast shows a much better generalization ability under the zero-shot setting. We believe that the proposed PromptCast task as well as our PISA dataset could provide novel insights and further lead to new research directions in the domain of time-series representation learning and forecasting.

1 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 (Hochreiter & Schmidhuber, 1997), Temporal Convolutional Network (TCN) (Lea et al., 2017), and Transformer (Vaswani et al., 2017). More recently, we are witnessing a fast growth of large-scale pre-trained models in Natural Language Processing (NLP) field. These models, also known as foundation models (Bommasani et al., 2021), are often pre-trained with an extremely large amount of data and have demonstrated good performance across various downstream tasks. For example, BERT (Devlin et al., 2019) can be adapted for multiple NLP tasks, CLIP (Radford et al., 2021) and GLIP (Li et al., 2021) 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 the advantages 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 including the state-of-the-art Transformer-based forecasting models (Zhou et al., 2021; Xu et al., 2021; Zhou et al., 2022; Drouin et al., 2022) 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

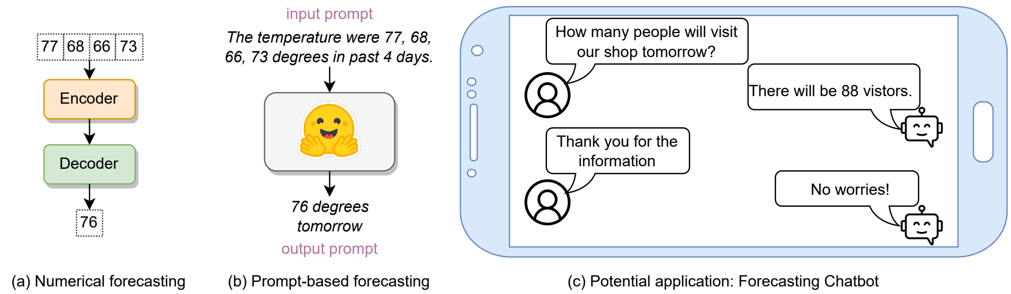

Figure 1: Conceptual illustrations of (a) existing numerical-based forecasting; (b) the framework of the proposed PromptCast; (c) a potential forecasting chatbot application based on PromptCast.

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 (Reed et al., 2022), 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 representation learning of general time-series using language-based paradigm, resulting also in the first large-scale dataset, PISA 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 the related research in the time series analysis domain. We further develop a benchmark in which we report the forecasting performance of 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 that 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 . It evaluates the state-of-the-art numerical-based forecasting methods, and popular language generation models under the PromptCast setting.

2 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 (Xue et al., 2022) or cities in weather forecasting. Under the general numerical time series forecasting task setting, the input is a history records of interested numerical data points collected on n 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 $x^m_{t_{\text{obs}+1}}$ of the next time step $t_{\text{obs}+1}$.

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 the numerical time series forecasting, the numerical values need to be transferred and described as natural language sentences. This data-to-text transformation is referred 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 value $x^m_{t_{\text{obs}+1}}$ is transformed as the output prompt. Consequently, the 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.

3 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. 3.1) and (2) to transform the numerical data to natural language input/output formats with prompts (Sec. 3.2). We also describe the features and statistics (Sec. 3.3).

3.1 Data Sources and Processing

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

- **City Temperature (CT)**[1]: This data source provides 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 dataset[2] includes the electricity consumption values (in Kwh) of 321 users. We filtered users with missing values and randomly selected 50 users with full records of the entire data collection period. In addition, the hourly consumption values of each selected user are aggregated into daily consumption 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 Xue et al. (2022) to almost 15 months and then randomly selected 324 POIs with full records.

The exact data collection periods are reported in Table 1. Following the standard protocol (Xu et al., 2021; Xue et al., 2022), each sub-set is divided into train/val/test at the ratio of 7:1:2 by the chronological order (Table 1). 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 (Xue et al., 2022), 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 (*i.e.*, the city name in CT and the POI id in SG). To remove this, we randomly assigned object-of-interest index U_m starting from 1 to M. Since the original data source of each sub-set is an

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

[2] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[3] https://docs.safegraph.com/docs/weekly-patterns#section-weekly-patterns-schema

Table 1: PISA dataset overview and key statistics.

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

aggregate statistics with no personal identifiable information, no private information can be decoded from the generated PISA dataset.

3.2 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 PromptCast is to describe and transform the numerical sequential data (*i.e.*, PISA-numerical) to natural language sentences. As demonstrated in Xue et al. (2022), 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 2 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 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).

3.3 Statistics Overview

To highlight the diversity of PISA, we analyze its key statistics as given in Table 1. PISA dataset contains 311,932 instances in total obtained from three different forecasting application domains. Each sub-set has its own statistical characteristics. The last row of Table 1 lists the value range distributions and the average value of each sub-set. It is noticeable that CT involves negative values in the dataset and ECL includes large numbers with a large range. This diverse data ensures the representativeness of our PISA dataset.

Table 2: 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.

4 Benchmark

In this section, we present the benchmarking study and analysis for the proposed PromptCast task. Through the experiments on the established PISA dataset, we aim to answer 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?

4.1 Evaluation Metrics

Although the proposed PromptCast task is a language generation task aiming at generating the target output prompts, we are particularly interested in the time series forecasting performance. To this end, the first step of the evaluation protocol is to decode the numerical predicted 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 cannot guarantee that the numerical value can be decoded from the generated output for every testing instance. To reflect this in the evaluation, we explicitly introduce 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 smaller Missing Rate means a better performance.

After decoding the numerical predicted value, the evaluation of PromptCast task will be the same as the evaluation of traditional numerical-based forecasting methods. As a result, two widely used metrics are also selected to evaluate the prediction performance in our benchmark: the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE). For each deep learning method, we

Table 3: 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

report the average performance and the standard deviation of 5 runnings with different random seeds. Note that the Missing Rate is naturally ignored when evaluating the numerical-based methods.

4.2 Baselines

In order to provide a useful benchmark of the proposed PromptCast task for other researchers, we select 10 popular natural language generation models and test their performance on our PISA dataset (*i.e.*, PISA-prompt). These language models are T5 (Raffel et al., 2020), Bart (Lewis et al., 2020), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), Electra (Clark et al., 2020), Bigbird (Zaheer et al., 2020), ProphetNet (Qi et al., 2020), LED (Beltagy et al., 2020), Blenderbot (Roller et al., 2021), and Pegasus (Zhang et al., 2020). 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). Other three basic numerical forecasting methods are included: AutoARIMA, LSTM, temporal convolutional network (TCN). Transformer-based time-series forecasting methods including the vanilla Transformer (Vaswani et al., 2017), the state-of-the-art Informer (Zhou et al., 2021), Autoformer (Xu et al., 2021), and FEDformer (Zhou et al., 2022) are also considered in this part. Overall, 20 different methods (10 numerical methods and 10 language models) are considered in the current benchmark and we will also expand the benchmark continuously in the future.

4.3 Experimental Performance

4.3.1 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 position embedding becomes popular in Transformer-based time series forecasting methods. This is equivalent to providing temporal information in the input prompt context (*i.e.*, From $\{t_1\}$ to $\{t_{\text{obs}}\}$ in Table 2). Specifically, based on

Table 4: 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 5: The Missing Rate performance of language models on PISA-prompt.

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

the implementations of Informer (Zhou et al., 2021)[4] and Autoformer (Xu et al., 2021)[5], we fully investigate and benchmark three different temporal embedding approaches, namely, *timeF*, *fixed*, and *learned*.

Table 3 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. As for the comparison of different embeddings, the *fixed* embedding demonstrates an overall good performance. This embedding leads to good predictions on 5 out of 6 metrics and the *timeF* is the best performer of the rest metric (MAE on SG). The *learned* embedding has the worst performance on CT and SG, whereas it beats the *timeF* on the ECL sub-set. The above results show that the *fixed* embedding is a favorable embedding approach for incorporating the temporal cues.

4.3.2 Pre-trained Language Models

For language models investigated in the benchmark, the ready-to-use pre-trained weights provided by HuggingFace (Wolf et al., 2020) are used for initialization. It is worth noting that the pre-trained weights are trained with general English-language corpora datasets such as BookCorpus (Zhu et al., 2015), CC-News (Liu et al., 2019), and OpenWebText (Radford et al., 2019). These common language datasets are about general articles and do not include specific time series sequences orientated data. 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 4. 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 3 and Table 4, 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 3, 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 5 reports the Missing Rate metric. It is clearly noticeable that only three methods (ProphetNet, Electra, and 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 5 , 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

[4]https://github.com/zhouhaoyi/Informer2020/blob/main/models/embed.py
[5]https://github.com/thuml/Autoformer/blob/main/layers/Embed.py

Table 6: Results of numerical forecasting methods and language models under the zero-shot setting.

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
Zero-Shot PromptCast													
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

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.

Zero-shot Performance. To further explore the language models for prompt-based forecasting, we conduct an experiment under the zero-shot setting (see Table 6). 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 results are given in Table 6. Similar to the results given in Table 3 (the normal training setting), the *fixed* embedding has better performance (5 best performance out of 6 metrics) than the other two embeddings under this challenging zero-shot setting. Since the date range of three sub-sets are different (Table 1), learnable *learned* and *timeF* temporal embeddings would result in worse performance when transferred to unseen scenarios. Except for the Autoformer and FEDformer on ECL, the numerical-based methods fail to yield reasonable predictions under the zero-shot setting. Considering the different characteristics of three sub-sets, such a poor performance is as expected for numerical methods. However, for the language models, although the performance is worse than the normal setting and the train-from-scratch setting, prompt-based forecasting can still generate sensible predictions. It shows a good generalization ability when time series forecasting is addressed with prompts (RQ2). This good zero-shot ability could bring benefits in real-world forecasting applications such as fast deployment for new forecasting scenarios and cold start forecasting for scenarios without any existing historical data. In the future, prompt-based forecasting could also support the exploration of more complicated forecasting scenarios such as can we predict energy consumption based on the weather temperature or can we forecast the customer visiting flow with the temperature trend.

Train-from-Scratch. Moreover, we disable the pre-trained weights and train language models from scratch with the training set of PISA dataset. We select three language models that demonstrate good forecasting performance on all three sub-sets for investigation in this part of experiments: Bart, Pegasus, and Bigbird. The results are reported in Table 7. We observe that there is a performance reduction for each method without loading pre-trained weights. Another finding is that even training from scratch, the language generation models (especially Bart) still can yield comparable prediction results compared to numerical-based methods. The above results show that using pre-trained weights could contribute to the forecasting performance gain. This also reveals one of the advantages of prompt-based forecasting: pre-trained language models can be easily leveraged.

Table 7: Results of language models under the train-from-scratch setting.

Pre-train	Model	CT				ECL				SG			
		RMSE		MAE		RMSE		MAE		RMSE		MAE	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Scratch	Bart	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

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

5 Discussion and Conclusion

In this paper, we present a novel task, PromptCast, which predicts time series using language models in a language generation manner. Given that this is the first work towards PromptCast task and no prior datasets are suitable, we build the first dataset aimed at investigating prompt-based forecasting. This large-scale dataset (PISA) contains three real-world time series forecasting scenarios. To advance the research of PromptCast, we also develop a benchmark on the released dataset and provides a set of solid baselines including both state-of-the-art numerical forecasting methods and language generation models. The experimental results show that using language models under the PromptCast setting has good forecasting performance and generalization ability.

Why Language Model Works for Forecasting. Using auxiliary information such as *time-of-day*, *day-of-week* (Zhou et al., 2021) and semantic information (**?**) has been demonstrated beneficial for boosting the forecasting performance. It often requires extra efforts to explicitly design special layers or modules in a heuristic way for the auxiliary inputs Although some approaches are proposed to incorporate auxiliary information, it remains challenging for a forecasting model to seamlessly model the contexts and the main temporal information. While for PromptCast, after prompting, both auxiliary and the core temporal inputs are considered together as tokens. 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-attention mechanisms in Transformers). The modeling of such relations would then result in good forecasting performance.

Broader Impact. We believe that this study's findings would offer forward-thinking concepts and fresh insights for other researchers. The research of time series forecasting could contribute in solving problems such as climate change and resources allocation for social good. 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 empowered by this work. Some potential directions are discussed below. (1) *Automatic Prompting*: In this paper, the transformation for numerical data to text is achieved by templates. Although template-based prompting is efficient, it still is difficult to produce diverse prompts. The fixed templates might also cause biases towards templates. To this end, one research direction is the automatic time series prompting or time series captioning (similar to image captioning (Herdade et al., 2019)) that aims at using generative models to describe the time series data. (2) *Explainable PromptCast*: Another research question yet to be fully investigated is why models designed for language modeling tasks can predict time series? A newly research has started to explore whether language models can be used for non-language downstream tasks (Dinh et al., 2022). However, future studies in the interpretability and explainability of PromptCast models would be an interesting and valuable research direction. (3) *PromptCast QA and Chatbots*: The research of PromptCast would trigger and promote time series forecasting question answering tasks and building chatbots applications with forecasting ability. Note that the PromptCast QA task differs from the recent TimeQA (Chen et al., 2021) that is proposed to answer general time related questions based on Wikipedia text and ForecastQA (Jin et al., 2021) which is also based on text articles. The core of PromptCast QA task is about the question-answering ability about forecasting upon the given sequential numerical value contexts.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Wenhu Chen, Xinyi Wang, William Yang Wang, and William Yang Wang. A dataset for answering time-sensitive questions. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Shashank Rajput, Michael Gira, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *arXiv preprint arXiv:2206.06565*, 2022.

Alexandre Drouin, Étienne Marcotte, and Nicolas Chapados. Tactis: Transformer-attentional copulas for time series. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5447–5493. PMLR, 2022.

Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. *Advances in Neural Information Processing Systems*, 32, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Woojeong Jin, Rahul Khanna, Suji Kim, Dong-Ho Lee, Fred Morstatter, Aram Galstyan, and Xiang 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*, pp. 4636–4650, 2021.

Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 156–165, 2017.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke 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*, pp. 7871–7880. Association for Computational Linguistics, 2020.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. *arXiv preprint arXiv:2112.03857*, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming 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*, volume EMNLP 2020 of *Findings of ACL*, pp. 2401–2410. Association for Computational Linguistics, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason 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*, pp. 300–325. Association for Computational Linguistics, 2021.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander 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*, pp. 38–45, October 2020.

Jiehui Xu, Jianmin Wang, Mingsheng Long, et al. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34, 2021.

Hao Xue, Flora D Salim, Yongli Ren, and Charles LA Clarke. Translating human mobility forecasting through natural language generation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 1224–1233, 2022.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pp. 11328–11339. PMLR, 2020.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong 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*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja 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*, pp. 19–27, 2015.