

Causal Text-to-Text Transformers for Water Pollution Forecasting

Kevin Roitero^{1,*}, Cristina Gattazzo², Andrea Zancola², Vincenzo Della Mea¹ and Stefano Mizzaro¹

¹University of Udine, Italy

²AcegasApsAmga SpA, Hera Group, Italy

Abstract

We propose a novel approach based on large language causal models to perform the task of time-series forecasting, and we use the proposed approach to effectively forecast the concentration of polluting substances in a water treatment plant; we address both short- and mid-term forecasting. As opposed to the classical state-of-the-art approaches for time-series forecasting, that handle numerical and categorical features following a standard deep learning approach, we transform the input features into a textual form and we then feed them to a standard causal model pre-trained on natural language tasks. Our empirical results provide evidence that large language models are more effective than state-of-the-art forecasting systems, and that they can be practically used in time-series forecasting tasks. We also show promising results on zero-shot learning. The results of this study open up to a wide range of works aimed at predicting future temporal values by leveraging natural language paradigms and models.

Keywords

Deep learning, Time-series forecast, Language models

1. Introduction

Water treatment plants, and in particular drinking water systems make use of different water treatment methods in order to serve safe drinking water to the population. Such systems use a series of treatments steps that transform the source water that enters the systems from river, lakes, etc. to tap water. To ensure that the water that leaves the system is drinkable and safe for the population, water treatment plants constantly monitor the concentration of polluting substances into the water, making use of specific instruments and techniques, such as the ion chromatography, an analytical separation technique based on ionic interactions. Such a technique separates ions and polar molecules based on their affinity and is able to carry out both qualitative and quantitative determinations. The field of application of ion chromatography is very broad, and the most common analyses with this technique concern water related analysis such as drinking water, sea water, waste water, rain water, determination of traces in electronics and power plants, quality control and analysis of impurities, etc.


AIABI'22: 2nd Italian Workshop on Artificial Intelligence and Applications for Business and Industries, November 28 – December 2, 2022, University of Udine, Udine, Italy

*Corresponding author.

✉ kevin.roitero@uniud.it (K. Roitero); cgattazzo@acegasapsamga.it (C. Gattazzo); azancola@acegasapsamga.it (A. Zancola); vincenzo.dellamea@uniud.it (V.D. Mea); stefano.mizzaro@uniud.it (S. Mizzaro)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

In this paper we deal with the analysis carried out by a ion chromatograph instrument located in the water treatment plant of Randaccio, which serves the city of Trieste. The instrument we deal with is managed by the Laboratory of AcegasApsAmga which makes the data available through the company data transmission network. At the laboratory the data are: downloaded, validated, uploaded to the internal system, used to create a report, evaluated. The created reports are then made available.

The instrument analyzes different substances; in this paper we focus on three of them which are important for the water treatment system: chloride, nitrate, and sulfate. The instrument monitors the concentration values of such substance approximately every 1h 30min, and collects a total of approximately 14 samples per day. Multiple samples are then joined together to form a time-series. The trend of the measured values in the time-series is constantly monitored and, if predefined patterns emerge (e.g., the value of a polluting substance increases), practical countermeasures are applied to the water plant, as for example the decision to exclude an intake point from the system and switch to another one where pollution levels are lower. It must be noted that such practical counter measures require a certain amount of time to be implemented. For this reason, the domain experts are interested in predicting in advance future values and trends for the observed substances.

In this paper we propose an effective practical methodology to reliably forecast the concentration of the polluting substances monitored by the ion chromatograph in the water treatment plant; our approach is based on transforming the input features from the time-series into a textual form and we then feed them to a standard causal model pre-trained on natural language tasks and asking the model to forecast the concentration of the substances for subsequent time steps. We validate our approach on real data coming from the treatment plant, providing also promising results on domain adaptation via zero-shot learning. Empirical evidence shows that our approach is more effective than state-of-the-art approaches for both short- and mid-term forecast.

2. Dataset

In the following we detail the dataset considered for the experimental part, used to validate the proposed approach. We consider the three substances (i.e., chloride, nitrate, and sulfate) monitored by the ion chromatography system which are modeled in the form of a time-series. It should be noted that the instrument monitors more than 3 substances, but those can not be interpreted as time-series, since their values assume the value of 0 for more than 95% of the observations. Our dataset is composed by observations made over a one year period, specifically between May, 2021 and May, 2022. A sample of the time-series for the three substances used in this work is shown in Figure 1 (first row). By inspecting the time-series behavior for those substances, we notice some interesting patterns.

First, we see that there are non negligible missing observations. The law requires minimum quality and safety levels, which are verified both internally by the company and externally by the health authority. The chromatograph used for collecting the dataset is not used for the production of required data, but it is part of an experimental setup aimed at verifying its usefulness in addition to formal measurements. As such, it is not always working, and this

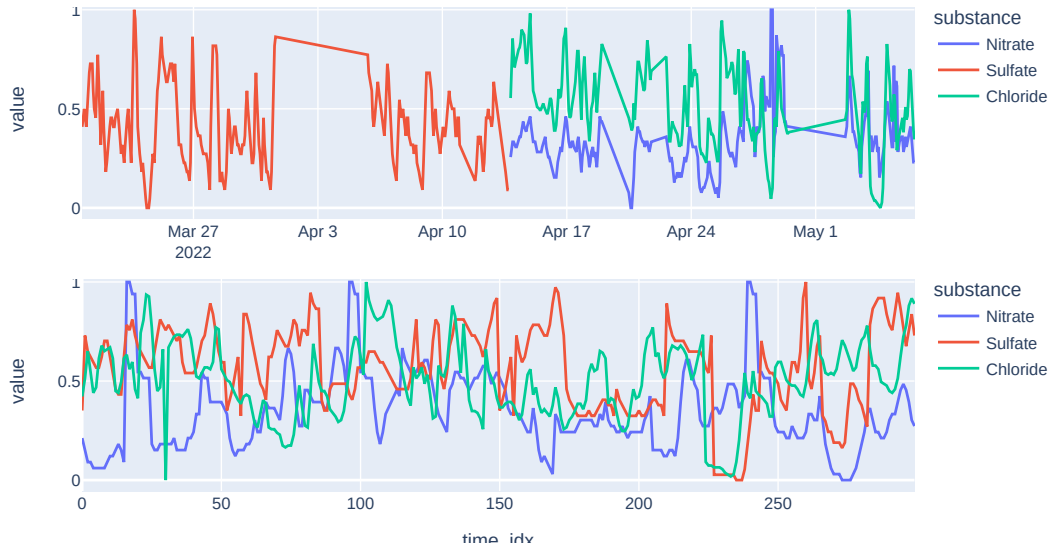


Figure 1: Time-series for the three substances before sampling (first row), and after the sampling process (second row). X-axis has been cut, and values are scaled in 0–1.

justifies missing samples. Then, we also notice that the monitoring period is not the same for all three substances, and in some periods the overlap is minimal or not-existent. In other words, when an observation is made for a substance, there is not guarantee that an observation will be available for one or both of the other substances for the corresponding time.

To overcome these issues, and transform the input time-series into a set of new ones without gaps, in a first pre-processing step we simply remove the missing observations, ending up with a smaller dataset having about 2,800 observations for each substance, on average 14 per day. Then, we check for seasonality effects by running both the seasonal decomposition using moving averages and Season-Trend decomposition using LOESS¹ [1] analyses. We found no evidence of seasonality or significant trend effects. This is also confirmed by the domain experts, which also confirmed that there is no interaction or dependence between the three substances (e.g., the pattern of chloride is not influenced by the temporal pattern of nitrate and sulfate, and the same holds for the other substances); thus, it does not make sense to use one time-series as feature to predict the others. In other words, we can frame the context as being a univariate time-series.

Then, to remove the bias introduced by the removal of missing values, we transform the dataset as follows. First, we compute for each substance the set of dates for which we have observations. Then, we random sample with replacement from the set of days and we concatenate the result. Let us make it clear by providing an example; if we suppose to have 10 days (i.e., d_1, \dots, d_{10}) and having missing values for days 2, 6, 7, and 9, the initial dataset can be represented as: $d_1, d_3, d_4, d_5, d_8, d_{10}$, while the resulting dataset can be represented as: $d_1, d_3, d_4, d_3, d_1, d_8, \dots, d_4$. Then, we form a training, validation, and test sets, by paying attention that if a day is present in the training set it can not be included in the test set. The final

¹see <https://www.statsmodels.org/dev/tsa.html>.

dataset is obtained by sampling approximately observations from 8600 days, and is composed as follows: 93, 183 observations in the training set, 4, 905 in the validation set, and 24, 522 in the test set. It should be noted that the sampling process performed is used only as a data augmentation technique to train the considered algorithms, and it does not affect the practical application of the proposed approach. A sample of the resulting dataset is shown in Figure 1 (second row).

3. Related Work

3.1. Time Series Forecast

The forecast of substances concentration that we deal with in the paper is related to general time-series forecasting research. State-of-the-art deep learning approaches designed for time-series forecasting are based on Recurrent Neural Networks (RNN) and their variations such as Long Short Term Memory (LSTM) networks [2] and Gated Recurrent Units (GRU) [3]. RNNs are a particular neural network architecture where the output of previous steps is fed as input to the current step. Such architecture is well suited to model scenarios where the prediction of the current value (e.g., the next word in a sentence or the next value of a time-series), is dependent on previous observations. More recently, architectures based on transformers as addition to classical architectures [4, 5] have been proposed [6].

While some successful attempt of adopting vanilla transformer architectures standalone [7] or in conjunction with other architectures [8] has been made in the setting of human mobility forecast where many contextual features are available, plain transformers and in particular causal models are quite new to the task of time-series forecasting, especially in the univariate setting and/or when there is a lack of context features, such as in the case investigated in this paper. This is primarily due to two main reasons [7], the absence of large-scale training data needed to develop pre-trained models, and the requirement for unique designs needed to capture domain-specific time-series features, such as seasonality effects.

In this work we propose an approach based on causal language models, and compare the proposed approach to state-of-the-art time-series forecasting models.

3.2. Large Language Models

In recent years, rapid advancements in the self-supervised learning paradigm joint with the success of the transformer-based architectures [9] contributed to the spread of general pre-trained and domain-specific fine-tuned models that demonstrated their effectiveness on a large variety of natural language processing (NLP) tasks; famous examples include BERT [10], a large masked language model pre-trained on English and Multi-language corpora which can be fine-tuned to a huge variety of tasks due to the learned language understanding ability. Masked language models are trained by randomly masking a percentage (e.g., 15%) of the input tokens and training the model to predict the masked tokens. The model loss is computed by considering the cross entropy loss between the logits of the model and the vocabulary tokens.

Opposed to masked language models, another popular set of transformer based models are causal models, as for example T5 [11]. Masked language models are trained to predict the

masked tokens in a sentence, and by doing so they leverage a bidirectional representation schema, because the representation of the masked tokens is learned based on the tokens that occur to the left and to the right of the masked part; the analogy for this representation schema is a “fill-in-the-blanks” problem statement. On the contrary, causal models predict the masked token in a given sentence but, unlike masked models, a causal model is allowed to just consider tokens that occur to the left of the masked set of tokens, thus leveraging a unidirectional representation schema. As result, such models are used in the case of generative tasks, where they are trained to predict the next token (or set of tokens) in a sentence based on the previous observed ones. As well as masked language models, the causal loss is computed by considering the cross entropy loss between the predicted token against the tokens in the vocabulary.

In this paper, due to the their intrinsic nature of being trained to predict the next value in a sequence based on the occurrence of past values, i.e., being that exactly the classical way of representing and modeling a time-series, in the following we base our solution on causal models, and specifically on the T5 model.

4. Methodology

4.1. Problem Formulation

We are interested, given a set of past observations of the substance concentration as measured by the ion chromatography, to predict the value for the substance for the subsequent timestamps. More in detail, we feed the models with 56 past timestamps, corresponding approximately to the measures obtained in the past 4 days, and we forecast two different future time steps: the next value in the time-series ($t+1$) which corresponds to a short-term prediction, as well as a mid-term prediction that allows domain experts to take practical countermeasures and apply them to the clean water plant, $t+14$ (i.e., one day forecast).

4.2. Metrics

To evaluate the effectiveness of the proposed approach, we rely on the following metrics used to evaluate the effectiveness of time-series forecasting methods: Mean Absolute Error (MAE), defined as the sum of absolute errors divided by the sample size, Max Error (ME), computed by considering the maximum of all absolute differences between the target and the prediction, and Root Mean Squared Error (RMSE), computed by considering the standard deviation of the residuals (i.e., prediction errors).

4.3. Deep Learning Methods

We consider the following state-of-the-art deep learning based methods: Long Short-Term Memory network (LSTM) [12], a sequence to sequence model which employs an architecture that allows the network to remember values over arbitrary intervals, thus showing a relative insensitivity to gap length between observations. Gated Recurrent Unit network [13] (GRU), a LSTM variation designed to solve the vanishing gradient problem, which makes use of the update gate and the reset gate to decide which part of information should be passed trough

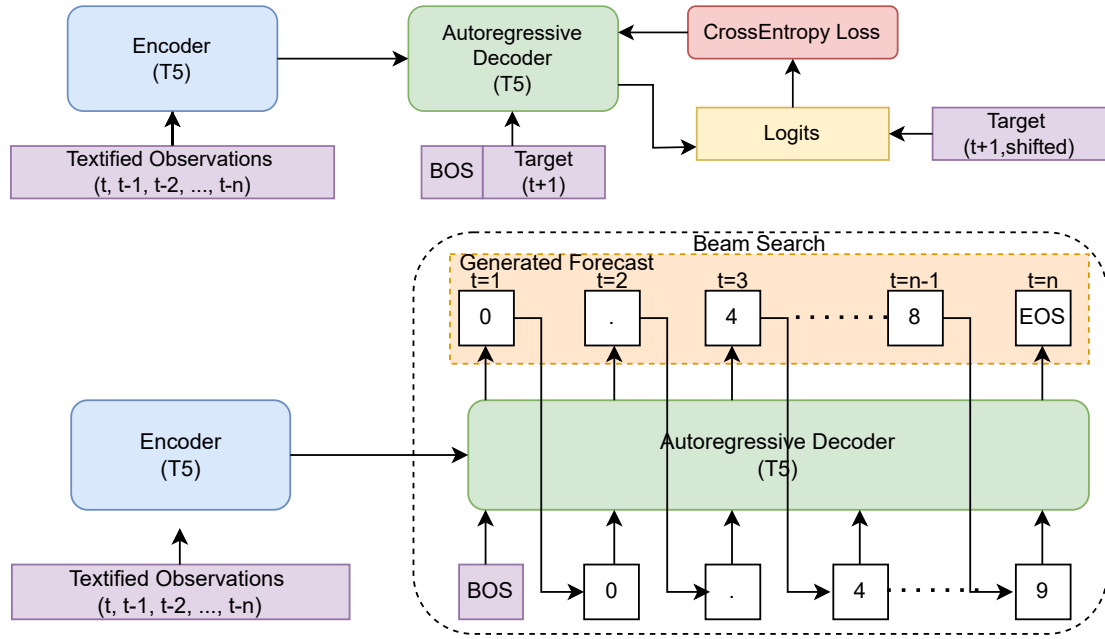


Figure 2: Training and inference phases for the transformer based model.

the network to compute the output. Neural Basis Expansion Analysis For Interpretable Time Series Forecasting [4] (NBeats), a deep neural architecture which is based on a set of backward and forward residual link and a deep stack of fully connected layers arranged in a doubly-residual stacking manner, and bases the predictions on a lookback and forecast period. Deep Autoregressive model [5] (DeepAR), an algorithm based on recurrent neural networks (RNN) which learns successive approximations of the target time-series. Temporal Fusion Transformer [6] (TFT), an attention-based neural network which leverages the recently developed transformer architecture [9] to identify important long-range patterns in the time-series and prioritizes the most relevant patterns.

4.4. Text-to-Text Transformer Model

To be able to train our model based on natural language processing, we first need to describe the input features i.e., the past observations of the time-series in a natural language form. To this aim, we leverage a process denoted as “textification” or “prompting” of the input features and that has been proven to be effective in the context of diagnostic texts [14, 15, 16] as well as in forecasting of human mobility [8]. Such approach takes in input the past observations of the time-series (i.e., the input features) and translate them into a string, which is then used as input to the NLP-based model. In this case we only rely on the array of floating point values corresponding to the past values of each time-series (called lags). We can denote our prompting schema as follows:

```
contextual information: {contextual features}.  
previous observations: {time-series features}
```

More in detail, if we consider a set of k previous values (i.e., lags), the prompt is as follows:

```
contextual information: {contextual features}.  
previous observations: {value} at time t-1, . . . , {value} at time t-k.
```

A real example of the prompt applied to the dataset is reported in the following, considering $k = 56$.

```
contextual information: the month is 4, the day is 9 (5 day of the week), 14 week of the  
year. the time is 08:14.  
previous observations are: 9.8 at time t-1, 9.8 at time t-2, 9.8 at time t-3, 9.8 at time t-4, 9.6  
at time t-5, 9.8 at time t-6, . . . [features from time t-7 to time t-54] . . . , 8.7 at time t-55, 9.2  
at time t-56.
```

We develop and train our model using the PyTorch² and HuggingFace³ frameworks. We rely on the T5-base model⁴, which was trained on a mixture of unsupervised and supervised tasks [11, Appendix Section]. The considered model is composed of an encoder decoder stack including 12 blocks, each comprising self-attention, optional encoder-decoder attention, and a feed-forward network. The attention is of dimension 64, while embeddings have 768 dimensions. The final model has about 220 million parameters.

We initialized the model with the pre-trained weights. We feed the textual input to the model by using custom prefixes “predict:”, “input:”, and “target:”. The experiments have been carried out on a Linux server equipped with 16x Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz, 70GB of RAM, and 2x Nvidia Geforce RTX 3090 GPUs for 3 epochs. As loss we use the conventional multi-class cross entropy loss, where the number of classes is equal to the size of the vocabulary, defined as $\mathcal{L} = -\frac{1}{B} \sum_{b=1}^B \sum_{k=1}^{|V|} y_k^b \log(\hat{y}_k^b)$ where the superscript b represents the current batch and B is the batch size, $|V|$ is the size of the vocabulary, y represents the true token, and \hat{y}_k is the output probability distribution over the vocabulary for each time-step.

To perform inference we generate text using beam search, thus generating the output sequence token-by-token by leveraging the cross-attention layers while passing the input to the decoder, and we generate auto-regressively the output of the decoder. We implement early stopping by setting the corresponding parameter to true. We found that our fine-tuned model generates floating point numbers for each beam, so we had no need to leverage constrained search strategies. The training and inference phases for our model are summarized in Figure 2.

Table 1

Metrics for chloride, nitrate, and sulfate test sets. We consider a lag of 4 days (14 observations per day x 4 days = 56), and we forecast the next value in the series (t+1), the subsequent day (t+14). We highlight in bold the most effective method for each section.

Model	Pred	Chloride			Nitrate			Sulfate		
		MAE	ME	RMSE	MAE	ME	RMSE	MAE	ME	RMSE
LSTM	t+1	.1572	.8278	.2003	.1086	.6468	.1419	.1893	.8857	.2381
GRU	t+1	.1577	.8109	.2007	.1090	.6484	.1424	.1888	.8869	.2375
DeepAR	t+1	.1533	.7839	.1949	.1058	.6473	.1377	.1851	.8391	.2324
NBeats	t+1	.1592	.8477	.2030	.1095	.6518	.1435	.1910	.9027	.2406
TFT	t+1	.1589	.8548	.2027	.1114	.6576	.1456	.1918	.9112	.2413
T5	t+1	.0316	.6027	.0674	.0121	.8163	.0402	.0182	.6596	.0579
LSTM	t+14	.1212	.7075	.1543	.0899	.5939	.1156	.1526	.6838	.1912
GRU	t+14	.1207	.6948	.1533	.0888	.6093	.1145	.1555	.7099	.1932
DeepAR	t+14	.1208	.6103	.1534	.0881	.6318	.1143	.1481	.6018	.1817
NBeats	t+14	.1278	.6482	.1620	.0934	.6958	.1209	.1575	.7043	.1954
TFT	t+14	.1260	.6246	.1594	.0853	.5909	.1114	.1457	.6236	.1792
T5	t+14	.1176	.6027	.1506	.0762	.6122	.1068	.1292	.6170	.1697

5. Results

Table 1 and Figure 3 show the results for the three substances for the short- and mid-term predictions. Let us start by inspecting the predictions for the subsequent timestamp. As we can see from the first section of the table, it is almost always the case that the proposed approach achieves higher effectiveness than the state-of-the-art approaches, with the only exception of the maximum error for the nitrate substance. Similarly, our model outperforms state-of-the-art models when performing predictions for the mid-term, that is predicting the substance concentration for the subsequent day, with the two only exceptions. This is an important result; in fact, having a reliable prediction for the subsequent day allows domain experts to plan and implement effective countermeasures for the drinking water plant.

Besides providing quantitative results, we also perform qualitative ones. Figure 4 shows the prediction for the sulfate substance when predicting the subsequent value in the time-series (i.e., t+1) for the best method (i.e., T5) and the second best (i.e., DeepAR) according to the effectiveness metrics as in Table 1. The results for the other two substances are identical and thus not reported. As we can see from the plot, both approaches approximate the real time-series. Nevertheless, by inspecting the two series closely we can find an important difference; the DeepAR algorithm (as we well as the other deep-learning based methodologies) tends to predict accurate values of the time-series, but they also tend to provide those forecasts with a certain time-lag; in other words, it predicts accurate values with a (mostly) fixed time delay, noticeable by inspecting the x-axis of the plot and comparing the pace of the two series, the real and the predicted one. Thus, if we select a real value in the y-axis, we see that the same value is

²<https://pytorch.org/>

³<https://huggingface.co/>

⁴<https://huggingface.co/t5-base>

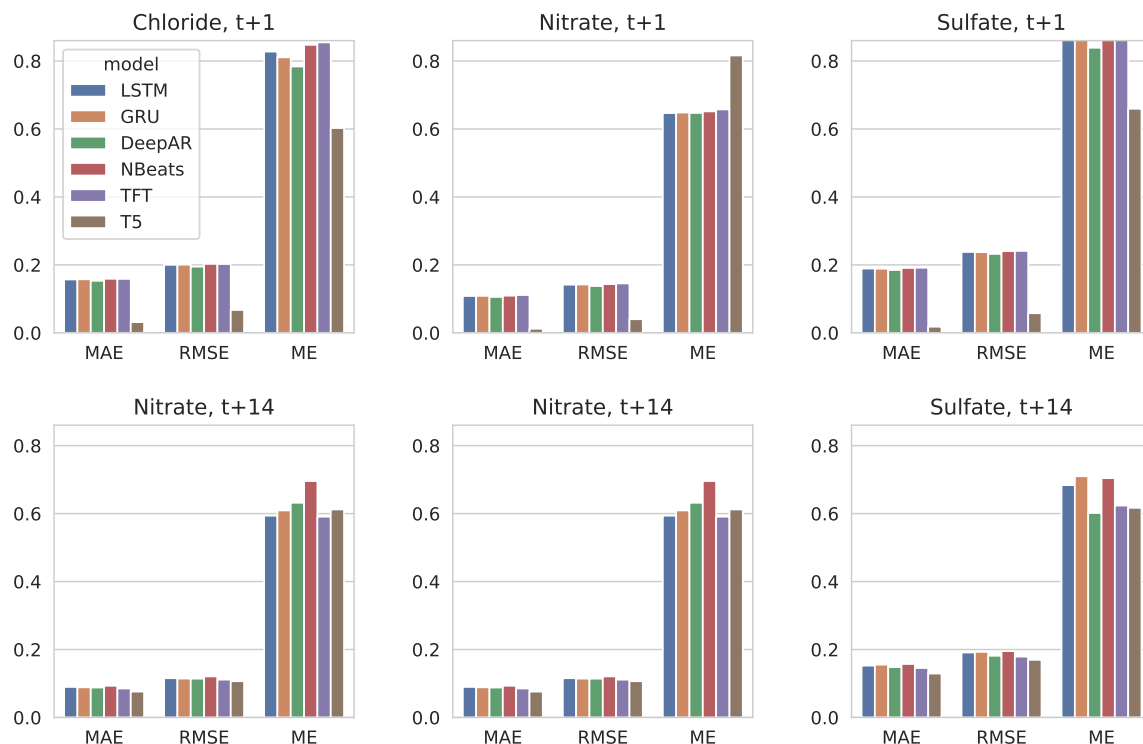


Figure 3: Metrics for chloride, nitrate, and sulfate test sets.

predicted by the algorithm in a time frame around $t+1$. This is a well documented effect in time-series forecasting literature and it is known to affect both machine and deep learning approaches. On the contrary, possibly due to the different modeling approach adopted by the natural language approach, we see that T5 does not suffer, or suffers in a limited form, from such effect. In fact, it tends to make different kind of errors, distributed mostly with shifts on the y-axis (i.e., prediction errors) rather than on the x-axis (i.e., delayed forecasts).

Figure 5, similarly to Figure 4, shows the prediction for the sulfate substance when predicting the value in the time-series for the next day (i.e., $t+14$) for the best method (i.e., T5) and the second best (i.e., DeepAR) according to the effectiveness metrics as in Table 1. The results for the other two substances are very similar and thus not reported. As we can see from the plot, the models make very different prediction errors, analogously to what observed in the previous result for $t+1$. In this case, while the DeepAR algorithm prediction follows a sort of moving average computed for the different time stamps, T5 successfully predicts some of the peaks present in the time-series, and makes errors distributed mostly around the y-axis.

6. Zero-Shot Capabilities

One of the documented advantages of large pre-trained natural language models is that they carry the ability of zero- and few-shot learning [17, 18] i.e., the ability of solving a task for a

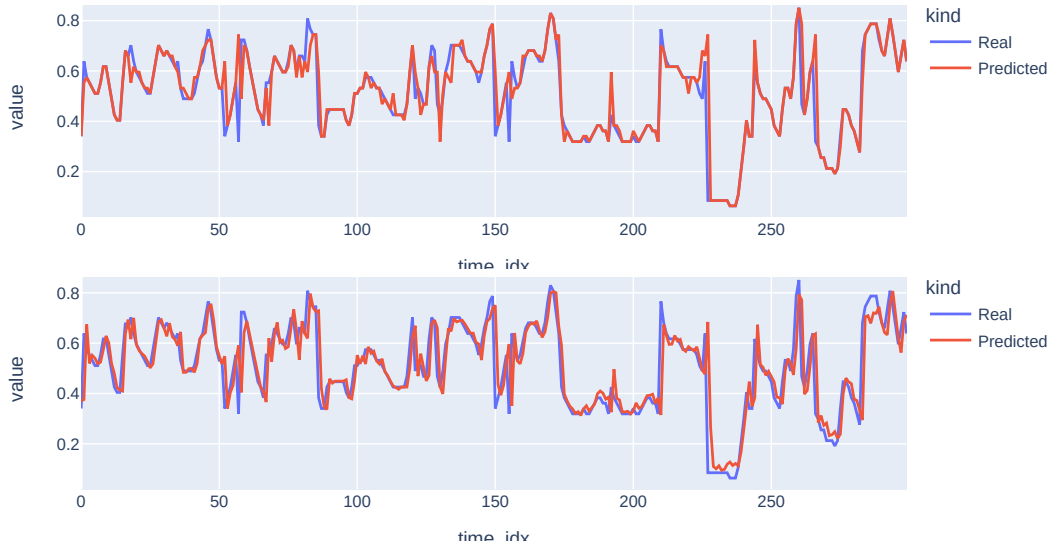


Figure 4: Prediction for the sulfate substance at $t+1$ for the T5 (best) and DeepAR (second best) method. X-axis has been cut, and values are scaled in 0–1.

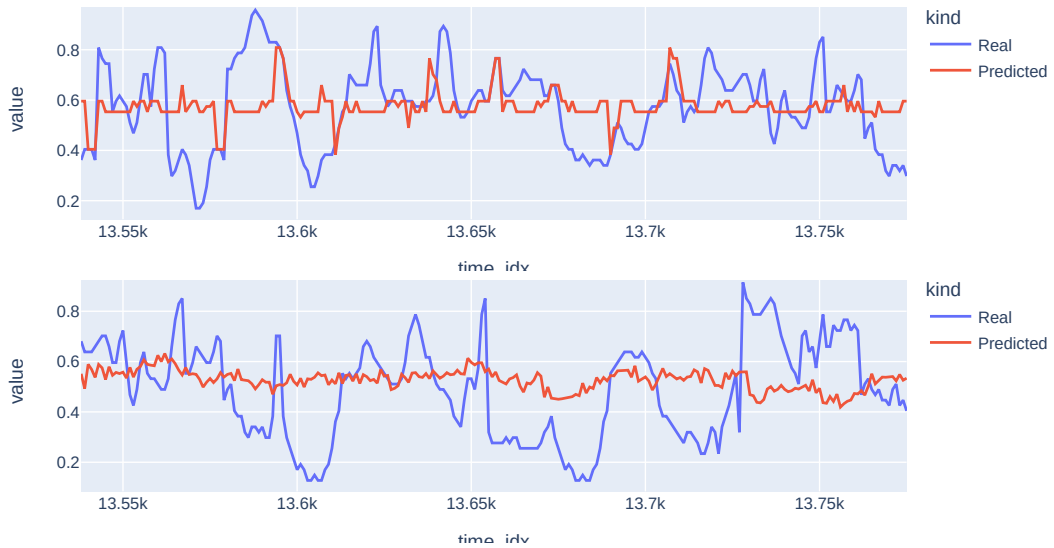


Figure 5: Prediction for the sulfate substance at $t+14$ for the T5 (best) and DeepAR (second best) method. X-axis has been cut, and values are scaled in 0–1.

domain without receiving any, or just few, examples of that task or for that domain at training phase. To further investigate the effectiveness of the T5 model to forecast the concentration of polluting substances in a water treatment plant, we conduct an experiment under the zero-shot paradigm. More in detail, we train each model on a substance and we test the trained model on the set of other substances which are different from the training one (i.e., we use the model trained on chloride to forecast the sulfate substance).

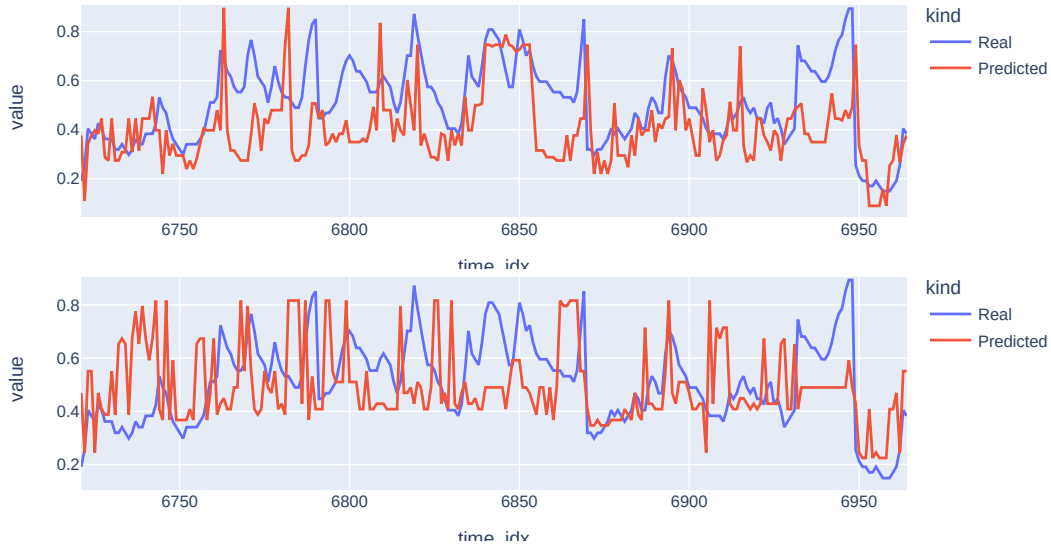


Figure 6: Prediction for the sulfate substance at $t+1$ performed using the T5 model trained on the chloride substance (above), and the nitrate substance (below). X-axis has been cut, and values are scaled in 0–1.

Figure 6 shows the qualitative prediction for the sulfate substance at $t+1$ performed using the T5 model trained on either the chloride or the nitrate substance. As we can see from the plots, while the model predictions are far from the ones computed with the corresponding model and test set (i.e., T5 trained on sulfate), they are not random either, and we can see that the predictions tend to follow the real time-series and correctly approximate some of the series peaks.

We also computed the effectiveness metrics for the zero-shot scenario: the model trained on chloride and nitrate achieves on sulfate respectively a MAE of 0.1717 and 0.1808 (T5 had 0.0182 and DeepAR 0.1377), a ME of 0.7368 and 0.8298 (T5 had 0.6596 and DeepAR 0.8391), and a RMSE of 0.2095 and 0.2182 (T5 had 0.0579 and DeepAR 0.2324). By looking at the metrics, we found that while the zero-shot model effectiveness is far the one obtained with the T5 model trained on domain specific data, the zero-shot models are almost as effective as, and for RMSE even more effective than, state-of-the-art deep learning approaches.

Although using the T5 model does not demonstrate optimal performances for the zero-shot task, this experiment show that causal models have promising generalization abilities for time-series forecast. Thus, we believe that further research in this direction, with the help of domain specific pre-trained models would improve the effectiveness and generalization abilities of those models.

7. Discussion and Conclusion

We studied the capabilities of causal language models (especially T5) for the task of forecasting the concentration of polluting substances in a water treatment plant, addressing both short- and

mid-term forecasting. To this end, we applied transformation to the input features to translate them into a textual form and feed them to the natural language model. The results show that our approach could improve state-of-the-art algorithms for forecasting on both the short and mid-term.

Given that the application of language models for the task of time-series forecasting might appear counter-intuitive at a first sight, let us make some remarks on why such approach works in practice. As we have seen, recent research showed that transformer based models are suitable and effective on a variety of tasks which are not related to the NLP paradigm, from images [19, 20] to videos [21] and even reinforcement learning [22] and graphs [23]. All the transformers based models rely on the attention mechanism which, joint with the training procedure that always consist in reconstructing a masked or perturbed part of the input, allow them to learn latent relationship in input sequences and between the input and output ones. For textual tasks they learn to reconstruct missing tokens, for visual ones they learn to reconstruct missing or altered frames, but they also showed the ability to learn and reconstruct complex structures such as (sub) graphs. For the same reason, we believe that the textual description of the time-series allows the model to form an accurate latent representation of it, which is then leveraged, jointly with the causal training modality (i.e., predict the next item in a sequence), to make accurate forecasting predictions. We plan to provide further insights on this by leveraging interpretability frameworks [24].

The results of this paper opens for a wide range of applications of language models to time-series forecasting problems. Future work aims at validating predictions with domain experts to understand to what extent the predicted values allow for practical and effective countermeasures to be applied in the treatment plant. Furthermore, we plan to improve zero-shot effectiveness by deepening the study on domain-invariant features.

Acknowledgments

This work was partially supported by the REACT-EU project “Data-Driven Multiutility Grid: Supporto alle Decisioni per Garantire la Sostenibilità dal Real Time al Lungo Termine” with “PON 2014-2020 AZIONE IV.6 GREEN”.

References

- [1] R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning, Stl: A seasonal-trend decomposition, *Journal of Official Statistics* 6 (1990) 3–73.
- [2] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural computation* 9 (1997) 1735–1780.
- [3] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, *arXiv preprint arXiv:1412.3555* (2014).
- [4] B. N. Oreshkin, D. Carпов, N. Chapados, Y. Bengio, N-beats: Neural basis expansion analysis for interpretable time series forecasting, *arXiv preprint arXiv:1905.10437* (2019).
- [5] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, Deepar: Probabilistic forecasting

- with autoregressive recurrent networks, *International Journal of Forecasting* 36 (2020) 1181–1191.
- [6] B. Lim, S. Ö. Arık, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting, *International Journal of Forecasting* 37 (2021) 1748–1764.
 - [7] H. Xue, B. P. Voutharaj, F. D. Salim, Leveraging language foundation models for human mobility forecasting, *arXiv preprint arXiv:2209.05479* (2022).
 - [8] H. Xue, F. D. Salim, Y. Ren, 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.
 - [9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, *Advances in neural information processing systems* 30 (2017).
 - [10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
 - [11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al., Exploring the limits of transfer learning with a unified text-to-text transformer., *J. Mach. Learn. Res.* 21 (2020) 1–67.
 - [12] Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: Lstm cells and network architectures, *Neural computation* 31 (2019) 1235–1270.
 - [13] R. Dey, F. M. Salem, Gate-variants of gated recurrent unit (gru) neural networks, in: *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS), IEEE, 2017*, pp. 1597–1600.
 - [14] M. H. Popescu, K. Roitero, S. Travasci, V. Della Mea, Automatic assignment of ICD-10 codes to diagnostic texts using transformers based techniques, in: *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI), IEEE, 2021*, pp. 188–192.
 - [15] K. Roitero, B. Portelli, M. H. Popescu, V. Della Mea, DiLBERT: Cheap embeddings for disease related medical NLP, *IEEE Access* 9 (2021) 159714–159723.
 - [16] V. Della Mea, M. H. Popescu, K. Roitero, Underlying cause of death identification from death certificates using reverse coding to text and a nlp based deep learning approach, *Informatics in Medicine Unlocked* 21 (2020) 100456.
 - [17] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, *Advances in neural information processing systems* 33 (2020) 1877–1901.
 - [18] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners, *arXiv preprint arXiv:2205.11916* (2022).
 - [19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, *arXiv preprint arXiv:2010.11929* (2020).
 - [20] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, Cvt: Introducing convolutions to vision transformers, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021*, pp. 22–31.
 - [21] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, H. Xia, End-to-end video instance segmentation with transformers, in: *Proceedings of the IEEE/CVF Conference on Computer*

Vision and Pattern Recognition, 2021, pp. 8741–8750.

- [22] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, I. Mor-datch, Decision transformer: Reinforcement learning via sequence modeling, *Advances in neural information processing systems* 34 (2021) 15084–15097.
- [23] V. P. Dwivedi, X. Bresson, A generalization of transformer networks to graphs, *arXiv preprint arXiv:2012.09699* (2020).
- [24] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, et al., Captum: A unified and generic model interpretability library for pytorch, *arXiv preprint arXiv:2009.07896* (2020).