ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Language Modeling in Logistics: Customer Calling Prediction

Xi Chen[1], Giacomo Anerdi [2], Daniel Stanley Tan [3], Stefano Bromuri [4]

1,3,4 - Open University of the Netherlands - Department of Computer Science
Valkenburgerweg 177, 6419 AT Heerlen - Netherlands

2- Maastricht University - Department of Advanced Computing Sciences
Minderbroedersberg 4-6, 6211 LK Maastricht - Netherlands

Abstract. Customer centers in logistics companies deal with many customer calls and requests daily. One of the most common calls is related to requesting an update on the shipment status. Proactively sending message updates to customers can reduce the number of calls. However, naively sending updates to everyone can cause unnecessary anxiety to people who do not want it, thus leading to customer dissatisfaction or even more calls. If a machine learning model could predict shipments leading to a customer call based on its journey, it could be possible to proactively send message updates only to customers likely to make a call. Therefore, reducing the workload in the customer center while increasing customer satisfaction. In large logistic companies where the volume of calls can reach a million calls per month, even 10% of the reduction of calls could already significantly reduce the additional expenses and workload associated with tracing a shipment. In this paper, we formulate the shipment journey as a variant of a language model. Specifically, we treat checkpoints (station, facility, time, event code) as tokens and predict the next checkpoint(station, facility, time delta, event code). Our core insight is that shipment checkpoints follow a set of rules that dictate the possible sequence of checkpoints. This is similar to how grammar rules dictate which words can follow another. Despite remaining a difficult problem, our experiments show that features learned by modeling shipment checkpoints as a language model can improve customer calling prediction.

1 Introduction

Customer centers of logistics companies are responsible for managing a diverse range of requests from a multitude of customers on a daily basis. The most frequent request is the demand for shipment progress updates. The customer center receives over a million of those calls each month globally, representing a significant workload. Logistic companies could regularly send shipment updates. However, not everyone is keen on receiving them. In fact, unwanted notifications have been shown to have adverse effects such as increased anxiety [1], which leads to reduced customer satisfaction, and in some cases even more calls. By leveraging machine learning algorithms to predict shipments that are likely to prompt customer inquiries, we can proactively send message updates only to customers that are likely to call, thereby reducing the number of calls received.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

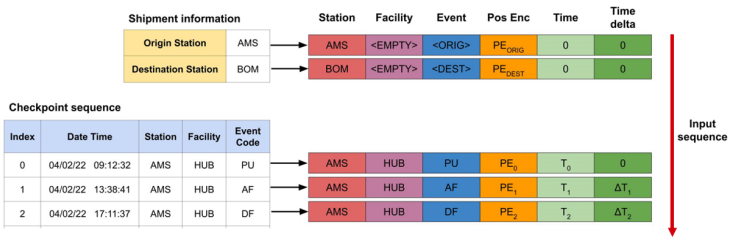

Fig. 1: Data structure

Every percentage of reduction in call volume can result in substantial savings in workload and enhanced customer satisfaction.

From a machine-learning perspective, predicting customer calling based on a shipment's journey can be seen as a time-series classification task with a binary target [2]. The shipment journey consists of a sequence of checkpoints as Figure 1 shows. Each checkpoint contains a timestamp, location, and event code that indicates what happened to a shipment.

Every logistics company has its own process rules that determine the sequence of checkpoints that a shipment makes during its journey. This shipment journey can be treated as a directional graph if the origin and destination are known. The way checkpoints are generated during this journey also follows a certain order. For example, a shipment should always have an arrival event before a departure event. Though, the sequence of the checkpoints is not always fixed; it also depends on the schedule and incidents such as missing flights, transportation delays, etc.

With these observations, we hypothesize that shipment journeys can be formulated as a language modeling problem [3], where checkpoint events taking place concerning the shipment are represented following a set of "grammar" rules dictated by the logistic process.

In this paper, we show how to formulate the prediction of shipment journeys by means of a variant of a language model. We demonstrate that customer calling prediction from shipment journeys can benefit from pre-training on a large unlabeled collection of data where the only supervision comes from predicting the next checkpoint, similar to how large language models [4, 5] have demonstrated that pre-training models can significantly improve downstream tasks [6, 7]. Our findings suggest that modeling shipment journeys as if they were sentences in a language model has the potential to help with various tasks in the logistics domain and that treating checkpoints and journeys as words and sentences is appropriate from a modeling perspective, thus opening many possibilities for future research.

The rest of this paper is structured as follows: Section 2 explains the method used in this paper; Section 3 shows the details of the experiment; The results and discussion are covered in Section 4 ; Section 5 Section 5 contains the conclusion and the future work.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

2 Method

2.1 Shipment journey as a language model

Similarly to the case of a language model predicting the next word, we train our model to predict the details of the next checkpoint in a shipment journey, as shown in Figure 2. This language model based pre-training allows our model to implicitly learn the logistic process and its rules.

We adopt a decoder-only architecture [5, 8] consisting of six decoder layers. In our experiments, checkpoints are represented by their station information, facility, event code, and time stamp, as shown in Figure 1. Station information, facility, and event code are categorical variables that we encode using an embedding layer. For the timestamp, we encode month, day, year, day of week, hours, and minutes using cyclical feature encoding.

Since the time information has a big variance, it is not clear whether it would be helpful to make it a prediction target. Therefore, we experimented on two different pre-training targets. First variant of our model only predicts the station, event code, and the facility. This will help the model understand the sequence of events. The second variant includes the transition time, which is the time difference (or time delta) between the previous checkpoint and the next checkpoint.

2.2 Customer calling prediction as downstream task

After the language model based pre-training, we fine-tune the model on the target of predicting customer calls, given a shipment journey consisting of a sequence of checkpoints. This is done by replacing the last output layer to output a binary prediction target optimizing a binary cross-entropy loss.

Time duration of a particular checkpoint is crucial information for customer calling predictions. As a matter of facts, if a shipment is stuck at the same checkpoint for a long time, then it is highly likely that the customer will complain. However, simply using the first k checkpoints as input does not indicate how long the shipment has been on the k-th checkpoint. Therefore, we append an end checkpoint at the $k+1$ position with a synthetic time stamp to indicate the duration of the k-th checkpoint and a special 'end' token for facility, station, and event codes.

During training, we augment the negative samples by appending end checkpoints in between two checkpoints where customers did not call. The idea is that if the customer does not call between checkpoint k and checkpoint $k+1$, then an artificial checkpoint $k+1$ with an earlier time stamp will also not trigger a call from the customer. For the positive samples, we generate end checkpoints with the same time stamp as the beginning of the call from the customer.

Due to the nature of the calls, the number of customers who call is significantly less than the number of customers who do not call. In our case, the ratio between positive and negative samples is approximately $1:19$. Thus, we sample the negative ones in such a way that the number of positive and negative

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9.
Available from http://www.i6doc.com/en/.

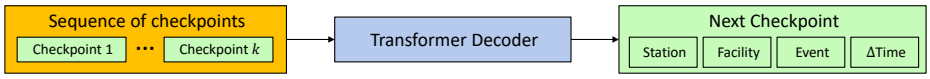

Fig. 2: Pre-train target.

samples is balanced. Specifically, we under-sample the negative samples in each epoch training.

3 Experiment

3.1 Data

The data that we used in this experiment comprises six months of shipments toward one country. For each shipment, we have corresponding checkpoint sequences as shown in Fig. 1.

Overall the data set contains a sample of 2.49 million shipments, where in 5.2% of the cases the customer called to obtain more information. In order to give some insights into the difficulty of this problem, Table 1 below shows the proportion of customer calls on a particular shipment event. The data is not only highly imbalanced, but also contains various types of noise and uncertainties. For example, some customers call at random times purely out of concern. Moreover, even for the shipments that present the same status, some customers call while others do not because of external factors, such as their personal situations or urgency of receiving the shipment.

We use three-month data for training, and half a month for validation and testing respectively. In order to evaluate statistical significance, we apply a five-fold rolling cross-validation with a window size of half a month in this experiment.

3.2 Implementation details

As for the transformer decoder, we use an Adam optimizer with a learning rate of $1e-5$. The number of heads is 6, and the dimension of the model is 512, with 6 layers of the decoder.

4 Results & Discussion

In this experiment, we evaluated three different models. The first model is a transformer model that is directly trained on the target. This acts as the standard classification baseline wherein we do not perform any language model based pre-training. The second and third models are the fine-tuned models with language model based pre-training. Specifically, the second model is pre-trained without a time delta, while the third model includes a time delta as a pre-training target.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

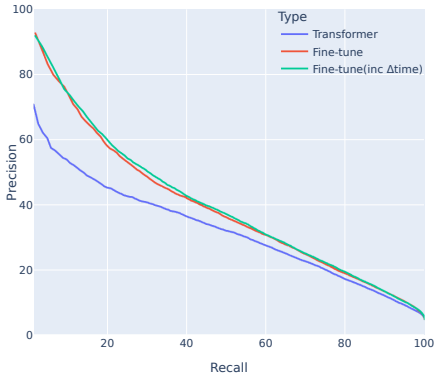

Fig. 3: The precision-recall plot.

Table 1: The proportions of the last event before the customer called with the non-call shipments for the same event. We show only the top 4 events related to customer calls.

Event code	A	B	C	D
Call	8.1%	5.2%	4.3%	3.2%
Non-call	91.9%	94.8%	95.7%	96.8%

The results are shown in the Table 2. We report the average recall from the five-fold cross-validation at different precision values. A complete precision-recall curve is visualized in Figure 3. Based on the results, we can observe that pre-training the transformer model on the shipment journey can improve the performance of customer call prediction. The difference is statistically significant when compared to the transformer without pre-training. We can also observe that the performance of the fine-tuning with time delta is marginally better than without. Time delta is important information when it comes to understanding the shipment journey. Predicting the time delta can help the model to understand the intrinsic shipping logic and the average duration of each checkpoint, thus leading to better performance when fine-tuning the downstream tasks.

The recall values are not particularly high on all models, which is reflective of how difficult the task is. The highly imbalanced data coupled with the randomness of customer call behavior makes it challenging to predict customer calls accurately. However, we would like to note that the level of performance of the models is already useful for industry usage. Depending on the business needs, we can always trade off precision & recall by moving the threshold (Figure 3). From a business perspective, based on 1 million calls per month, if we use a precision threshold at 50% and send the customer an update message, even if only half of the customers can be prevented from contacting the logistic company, this can reduce 15% of the calls, which accounts for 150k calls per month.

5 Conclusion & Future work

In this paper, we showed that it is possible to formulate the shipment journey as a variant of the language model. This opens the possibility of implementing a large language model that can be used in the logistic domain. The downstream task customer calling prediction can also be a benefit for the logistic company.

Future work concerning customer contact prediction could imply looking into

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Precision	Random Guess	Transformer	Fine-tune	Fine-time (w/ time delta)
40	n.a	32.95	43.50 (2.9e-3)	**45.55** (3.6e-3)
50	5.2	14.05	29.29 (6.0e-3)	**30.86** (9.3e-4)
60	n.a	4.80	19.39 (1.7e-3)	**20.16** (2.5e-3)
70	n.a	1.58	12.04 (5.2e-3)	**13.14** (8.3e-3)
80	n.a	0.76	6.60 (7.6e-3)	**7.65** (9.6e-3)

Table 2: Recall under the different precision threshold. The number in the bracket is the P-value from paired student T-test compared with Transformer.

data cleansing and uncertainty measurement approaches [9, 10] as there is certain randomness involved in the data. For the language model itself, further analysis could be performed on fine-tuning it towards various downstream tasks to validate its usage in other logistic case studies.

References

[1] Jon D Elhai, Dmitri Rozgonjuk, Ahmad M Alghraibeh, and Haibo Yang. Disrupted daily activities from interruptive smartphone notifications: Relations with depression and anxiety severity and the mediating role of boredom proneness. *Social Science Computer Review*, 39(1):20–37, 2021.

[2] Kalliopi Tsolaki, Thanasis Vafeiadis, Alexandros Nizamis, Dimosthenis Ioannidis, and Dimitrios Tzovaras. Utilizing machine learning on freight transportation and logistics applications: A review. *ICT Express*, 2022.

[3] Chenguang Wang, Xiao Liu, and Dawn Song. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*, 2020.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019.

[7] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

[8] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[9] Öncü Hazır and Gündüz Ulusoy. A classification and review of approaches and methods for modeling uncertainty in projects. *International Journal of Production Economics*, 223:107522, 2020.

[10] Reihaneh H Hariri, Erik M Fredericks, and Kate M Bowers. Uncertainty in big data analytics: survey, opportunities, and challenges. *Journal of Big Data*, 6(1):1–16, 2019.