2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020230

POSGen: Personalized Opening Sentence Generation for Online Insurance Sales

Yu Li*, Yi Zhang †, Weijia Wu †, Zimu Zhou ‡ and Qiang Li *
*College of Computer Science and Technology, Jilin University †Wesure Inc.
‡School of Information Systems, Singapore Management University
Email: {yuli19,li_qiang}@jlu.edu.cn, {jamesyzhang, wymanwu}@wesure.cn, zimuzhou@smu.edu.sg

***Abstract*—The insurance industry is shifting their sales mode from offline to online, in expectation to reach massive potential customers in the digitization era. Due to the complexity and the nature of insurance products, a cost-effective online sales solution is to exploit chatbot AI to raise customers' attention and pass those with interests to human agents for further sales. For high response and conversion rates of customers, it is crucial for the chatbot to initiate a conversation with personalized opening sentences, which are generated with user-specific topic selection and ordering. Such personalized opening sentence generation is challenging because *(i)* there are limited historical samples for conversation topic recommendation in online insurance sales and *(ii)* existing text generation schemes often fail to support customized topic ordering based on user preferences. We design POSGen, a personalized opening sentence generation scheme dedicated for online insurance sales. It transfers user embeddings learned from auxiliary online user behaviours to enhance conversation topic recommendation, and exploits a context management unit to arrange the recommended topics in user-specific ordering for opening sentence generation. POSGen is deployed on a real-world online insurance platform. It achieves 2.33x total insurance premium improvement through a two-month global test.**

***Index Terms*—Online Insurance Recommendation; Transfer Learning; Data-to-text Generation**

I. Introduction

The insurance industry is shifting digital for cost-effective product marketing and sales [1]. For example, mini-programs are deployed on social media to approach potential customers [2]. To promote insurance sales via such digital channels, a chatbot can be exploited to initiate conversations with millions of users, while those interested are then seamlessly handed over to insurance agents for complex queries.

Although chatbot AI techniques been applied in question answering [3], chit-chat [4], product recommendation [5] etc., they are unfit for promoting online insurance sales due to the distinctive chatbot-user interactions therein. Compared with other daily goods, insurance products are more complex and less familiar to the public. Customers are reluctant to inquire on insurance products [6], and a tedious FAQ even scare potential buyers away [7]. For insurance products, the chatbot should actively initiate the conversation with users via one or multiple *opening sentences*, rather than passively waiting for their queries. Effective opening sentences are crucial to raise user interests, and eventually increase sales [8].

To understand the characteristics of effective opening sentences, we resort to the historical opening sentences used by human agents and their sales records from a large online insurance platform. We divide the agents into two groups based on their seniority, *i.e.*, junior and senior, and introduce a controlled group of randomly selected senior agents for online tests, *i.e.*, senior agents (controlled), whose opening sentences are selected from junior agents by the system. Fig. 1 plots the response rates (RRs, *i.e.*, the number of responded users divided by the number of contacted users) and the conversion rates (CVRs, *i.e.*, the number of converted users divided by the number of contacted users) of the opening sentences used by three agent groups. The senior group notably outperforms the junior group in both RR and CVR, *i.e.*, the opening sentences used by the senior group are more effective. Furthermore, the performances difference between senior agents and senior agents (controlled) demonstrate the opening sentences notably affect conversions, even if agents have similar sales skills. Fig. 2 shows a few example opening sentences. We observe that the opening sentences from the junior agents are generic and indifferent to customers. In contrast, the effective opening sentences used by the senior group are highly *personalized*: they exhibit user-specific topic selection and ordering.

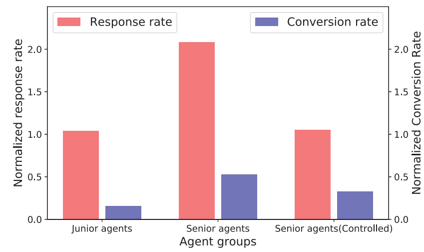

Fig. 1: Effectiveness of opening sentences used by junior (within one year of experience), senior (over three years of experience) human agents and controlled senior agents (opening sentences are selected from junior agents). The customers are randomly distributed to the three agent groups. From the results we can see: *(i)* Opening sentences from the senior group outperform the junior group in both response rate and conversion rate (junior agents vs senior agents). *(ii)* Opening sentences is important in promoting final conversions (senior agents vs senior agents (controlled)).

However, personalized opening sentence generation for online insurance sales faces two technical challenges.

- *How to recommend user-specific topics for opening sentence generation?* Although it is viable to learn individuals' preferences on topics from the opening sentences used by senior agents and select the top k for sentence generation, there are insufficient such historical conversation data. Given the vast user attributes and conversation topics, it is challenging to extract effective user representations for ranking the preference of topics given limited historical opening sentence samples.
- *How to generate opening sentences with user-specific*

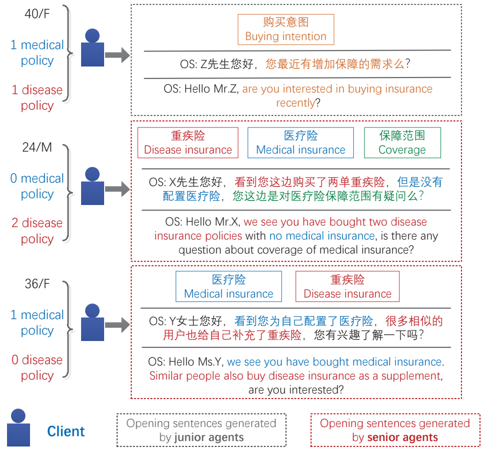

Fig. 2: Example opening sentences by junior and senior agents. The left shows a few client features that agents may use in the opening sentences. The right shows the opening sentences and the corresponding topics. The senior agents tend to adjust the opening sentences according to the clients' features.

topic ordering? This question belongs to the research on data-to-text generation [9]–[13]. However, prior studies either require fixed sentence length [12], [14] or can only generate sentence from one topic or style [9], [10]. A few [11], [13], [15] generate sentences on multiple topics, yet they arrange topics in a generic order *e.g.*, for fluency, without considering user-specific features.

We design POSGen, a personalized opening sentence generation scheme for promoting online insurance sales. To handle the data sparsity in topic recommendation, we exploit users' auxiliary behaviors (*e.g.*, clicks, conversions, and browsing history) about page items in the online insurance mini-program to enhance topic recommendation. These auxiliary behaviors on the same online insurance platform share similar user representations and are thus transferable to topic recommendation [16], [17]. Accordingly, we design a selective attentive transfer learning model (SATL), which adaptively learns user representations from these auxiliary behaviors. To generate sentences with user-specific topic ordering, we propose a context-aware sentence generation model (CSG). In addition to the topics themselves, we also feed the user's embedding into a context manage unit to determine the next topic for sentence generation. Thus the generated sentences not only account for fluency, but also conform to individual preferences.

Contributions and Main Results. The contributions of this paper are summarized as follows.

- To the best of our knowledge, POSGen is the first proposal on opening sentence generation dedicated to promote online insurance sales.
- We design a personalized opening sentence generation scheme with user-specific topic selection and ordering. We also tackle practical challenges such as data sparsity by learning transferable user representations from auxiliary user behaviors related to online insurance.
- We evaluate the performance of POSGen on a large online insurance platform. Through a two-month global test, the results show that POSGen increased 51% number of average talk rounds, which were finally converted into 2.33× improvement on the total insurance premium without decreasing the quality of users' experiences (QoE).

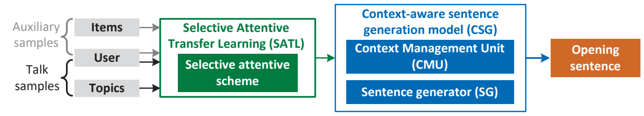

Fig. 3: Overview of POSGen, which consists of selective attention transfer learning (SATL) and context-aware sentence generation model (CSG). SATL extracts user's embedding and recommends interested topics, which are then passed to CSG to generate personalized opening sentences.

II. Related Work

Transfer Learning in Recommendation Systems. Transfer learning has been applied in recommendation systems to improve their performances with limited data or supervision [18]. Zhao *et al.* [19] propose an entity-correspondence mapping method for collaborative filtering to improve recommendation quality. Chen *et al.* [17] devise an adaptive transfer network, with several attention gates to combine low-level feature representations from source domains to improve recommendation in the target domain. However, they do not apply to our scenario because they either require strong constraint on data in source and target domains, *e.g.*, the users are required to have features in source domain [17], or fail to design efficient structure for transferring knowledge from source to target [20].

The SATL model in our POSGen also exploits transfer learning to improve topic recommendation. Unlike previous work, we relax the constraints on samples by designing a shared embedding layer to capture the co-occurrence of user features in the source and target domains. Based on the shared embedding layer, we design a dual attentive mechanism in SATL to adaptively learn knowledge from source domains.

Text Generation. Personalized opening sentence generation relates to data-to-text generation [9], [12], [14], [15], [21], [22]. Bowman *et al.* [22] propose a variation autoencoder (VAE) based method to generate sentences. By introducing a latent variable, the model can generate sentences with diverse patterns. A follow-up [21] introduces a context prior to generate sentences with given contexts, *e.g.*, topics. PHVM [14] adapts the model in [21] for generating product descriptions. Other text generation applications include controllable poetry generation [12], data-to-summary generation [13], etc.

Our personalized opening sentence generation is more challenging because we demand user-specific sentence structuring, which no prior studies can fulfill. Hence we propose a context-aware sentence generation model (CSG) to adjust the topic ordering based on user features.

III. POSGen Overview

In this section, we present an overview of POSGen, a new personalized opening sentence generation scheme for promoting online insurance sales. As shown in Fig. 3, POSGen consists of two functional modules.

- **Selective attentive transfer learning model (SATL).** This module extracts user embeddings and recommend

TABLE I: Summary of auxiliary behaviors.

Behavior types	Items	Description
Click	insurance/videos/articles	Whether a user clicks an item
Conversion	insurance	Whether a user buys an insurance product
Visit history	insurance/videos/articles	Whether a user visits an item

the top k conversation topics via transferring learning from auxiliary user behaviours like clicks on presented insurance products on the same online insurance platform. SATL is a two-tier model consisting of an auxiliary network (AuxNet) for training auxiliary samples and a topic recommendation network (TopicNet) for topic recommendation. TopicNet shares feature embedding with AuxNet and uses a dual attention array to adaptively learn high-level knowledge from AuxNet. The design enables TopicNet to learn comprehensive user preferences for accurate topic recommendation.

- **Context-aware sentence generation model (CSG).** This module generates personalized sentences based on the given user embedding and recommended topics from SATL. In CSG, we design a context management unit to customize topic orders based on user's embedding. The ordered topics then guide the sentence generator and output personalized opening sentences.

IV. Selective Attentive Transfer Learning (SATL)

This section presents the design of our selective attentive transfer learning (SATL) model, which aims to predict users' topics of interest. Given the sparsity of talk samples, we design a selective attentive scheme to transfer knowledge from samples of users' auxiliary behaviours. With the assist of the transferred knowledge, SATL is able to capture rich users' representations and make accurate topic prediction. Fig. 4 shows the architecture of SATL. We explain its data input and knowledge transfer mechanism below.

A. Data Input

SATL takes two types of data inputs: auxiliary samples and talk samples. The talk samples are collected from the conversations between agents and customers. The topics in the talk samples are manually annotated. The auxiliary samples have two components: item features and user features. Items refer to the elements users interact with, *e.g.*, presented page item when user clicks. Talk samples consist of user features and topics collected from opening sentences. The auxiliary samples are obtained from three user behaviours on the same insurance platform: clicks, conversions, and visit histories, with 0/1 as the labels. Table I lists the items and descriptions of each behavior of the auxiliary samples. For each type of samples, we mark its label as 1 if the user clicks/buys/visits the given item, and 0 otherwise.

We use a multi-hot vector $\hat{q}_{ik}$ to denote the auxiliary sample of user i with item k, *i.e.*, $\hat{q}_{ik} = [\underbrace{1, 0, ..., 1}_{user\ \hat{u}_i}, \underbrace{0, 1, ..., 0}_{item\ \hat{im}_k}]$, where im_k refers to the k-th item. The dense vector q_{ik} embedded by the dictionary is as $q_{ik} = Emb(\hat{q}_{ik})$. Similarly, we use $\hat{x}_{jl} = [\hat{u}_j, \hat{tp}_l]$ for the raw talk sample of user j and topic l.

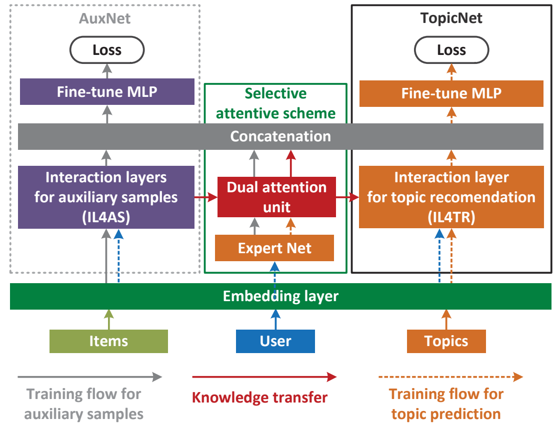

Fig. 4: SATL architecture. A dual attention array cooperates with an expert network to transfer knowledge from auxiliary samples to topic recommendation.

B. Knowledge Transfer Mechanism

SATL is inspired by knowledge transfer in recommendation systems [23]. As with these proposals, the embedding layer of SATL is shared for both tasks since those users are from the same online insurance platform and share similar feature representations. However, SATL improves upon [23] by allowing not only such low-level feature sharing, but also high-level feature sharing. Specifically, we design a dual attention array and an expert network in the interaction layers between the AuxNet and TopicNet. The expert network is activated by both auxiliary samples and talk samples for learning more representative user embeddings. Upon the expert network, a dual attention array is designed for extracting high-level features to improve the accuracy of topic recommendation task. For the expert network, we use multi-layer perceptron as its inner structure. In the dual attention array, we design two attention units *i.e.*, $\boldsymbol{Att} = [Att^{aux}, Att^{tp}]$. Att^{aux} controls the weight on the expert network output when trained on auxiliary samples, while Att^{tp} controls the weight on the expert network output when trained on talk samples.

To explain the details of the high-level knowledge transfer, we abstract the interaction layers of the the AuxNet and TopicNet as interaction layers for auxiliary samples (IL4AS), and interaction layers for topic prediction (IL4TR), respectively. We use H to denote the intermediate output of IL4AS and G for IL4TR. In the training process, we first train IL4AS through auxiliary samples and then, train IL4TR through talk samples and trained IL4AS. Let the output of expert network be $F(u_i)$. The attention weights $\boldsymbol{w}^{aux}$ for auxiliary sample q_{ik} is computed as:

$$\boldsymbol{a}^{aux} = H(q_{ik}) * W^{aux} * F(u_i)^T,\ \boldsymbol{w}^{aux} = softmax(\mathbf{a}^{aux}) \quad (1)$$

where W is the weight matrix in the attention unit. The output of Att^{aux} is then $Att^{aux}(q_{ik}, F, H) = \boldsymbol{w}^{aux} F(u_i)$.

The training for topic recommendation is similar. The only difference is that we use the output of the previously trained expert network through dual attention array. Following the definition above, we have:

$$Att^{tp}(x_{jl}, F, G) = \boldsymbol{w}^{tp} F(u_j) \quad (2)$$

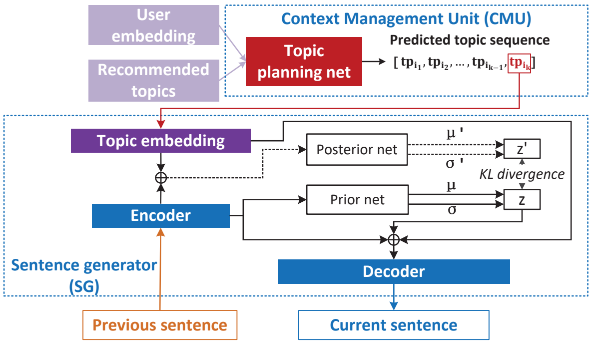

Fig. 5: CSG architecture. A context management unit (CMU) is used to customize the topic order based on user's embedding. The recommended topics are then sent to sentence generator (SG) to generate opening sentences.

The output of dual attention array is concatenated with H and G respectively and then sent to the upper fine-tune MLP to calculate the loss. We use two loss functions: $\mathcal{L}^{aux}$ for auxiliary samples and $\mathcal{L}^{tp}$ for topic recommendation. For simplification, we use logloss for the two loss functions. The loss of SATL $\mathcal{L}$ can be calculated as:

$$\mathcal{L} = \alpha\mathcal{L}^{aux} + (1-\alpha)\mathcal{L}^{tp} + \lambda\Omega(\Theta) \tag{3}$$

where α is pre-defined controlling weights, $\Omega(\Theta)$ is the regularization for network parameters and y is the sample label. For $\mathcal{L}^{aux}$ and $\mathcal{L}^{tp}$, CE denotes the cross entropy and MLP denotes the fine-tune multi-layer perceptron.

V. Context-aware Sentence Generation (CSG)

This section introduces the context-aware sentence generation (CSG) model, which focuses on generating opening sentences based on the predicted topics from SATL. We design two components in CSG: a context management unit (CMU) and a sentence generator (SG). The CMU determines the ordering of topics so that the generated sentences is logical and easy to understand. The generated topic sequences are then passed to SG to generate corresponding sentences. In SG, we design a conditional VAE [21] for sentence generation. Different from previous works, we incorporate previous sentence in the context so that the fluency and coherence of generated sentences can be guaranteed. Fig. 5 shows the architecture of CSG.

A. Context Management Unit (CMU)

This module is designed for planning topic order based on user's preferences, *i.e.*, user's embedding from SATL. The input of CMU is user's embedding and the recommended topics from SATL. The input will be sent to a topic planning network to generate topic sequence based on user's preferences. We design three components in the topic planning network: a GRU unit, an attention unit and a fine-tune MLP. Let the topic sequence by step $n-1$ be $\text{TP}_{n-1} = \{tp_{i_1}, tp_{i_2}, ..., tp_{i_{n-1}}\}$. The process in generating TP_n is as below.

First, we use the embedding dict, *i.e.*, the embedding layer from SATL, to derive the dense vector of TP_{n-1}, *i.e.*, v_{n-1}^{TP}.

The topic sequence vector v_{n-1}^{TP} are then sent to GRU to get encoded output O_n^{TP}:

$$O_n^{TP}, h_n^{TP} = \boldsymbol{GRU}^{\text{CMU}}(v_{n-1}^{TP}, h_{n-1}^{TP}) \tag{4}$$

where h_{n-1}^{TP} is hidden state for step $n-1$. When $n=1$, we pad h_0^{TP} with zeros for initialization.

We then feed the output O_n^{TP} and user embedding u into the attention unit to calculate the tuned vector, *i.e.*, Att_n^{CMU}

$$Att_n^{\text{CMU}} = softmax(u * W^{\text{CMU}} * O_n^{TP}) * O_n^{TP} \tag{5}$$

Then the output Att_n^{CMU} is concatenated with u and sent into the MLP to calculate the probability on recommended topics:

$$p(tp_i|u, \boldsymbol{TP}_{n-1}) = \boldsymbol{MLP}^{\text{CMU}}([Att_n^{\text{CMU}}; u]) \tag{6}$$

We use beam search to generate the topic sequence as [11].

B. Sentence Generator (SG)

This module takes the recommended topic sequence TP_n as input and generates personalized opening sentences. We build SG on top of a conditional variational autoencoder (CVAE) for its capability to generate controlled sentences [21]. Let $\boldsymbol{S}_{n-1} = \{w_1, w_2, ..., w_M\}$ be the sentence generated based on TP_{n-1}. The process to generate $\boldsymbol{S}_n$ is as follows.

To generate $\boldsymbol{S}_n$ based on the topic sequence as well as guarantee its quality, *e.g.*, coherence, we use topic tp_{i_n} and previous sentence $\boldsymbol{S}_{n-1}$ as the context c to learn the latent factor for generation, *i.e.*, $c = [Emb(tp_{i_n}); enc(\boldsymbol{S}_{n-1})]$, where $enc(\boldsymbol{S}_{n-1})$ denotes the embedding vector from the encoder. Then the latent parameter z' for posterior network and z for prior network can be calculated via reparameterization, *i.e.*, $z' = \mu' + \epsilon' * \sigma'$, $\epsilon' \in \mathcal{N}(0,1)$, $z = \mu + \epsilon * \sigma$, $\epsilon \in \mathcal{N}(0,1)$. The design enables posterior net to learn a more accurate distribution of z to generate $\boldsymbol{S}_n$ given the connection between $\boldsymbol{S}_n$ and $\boldsymbol{S}_{n-1}$. The concatenation also allows the latent factor z to capture the smooth transition between $\boldsymbol{S}_{n-1}$ and $\boldsymbol{S}_n$ from training.

The latent variable z is then sent to the decoder for sentence generation. In our design, we choose bidirectional GRU as the inner structure of the decoder. We concatenate the topic embedding $Emb(tp_n)$, encoded previous sentence vector $enc(\boldsymbol{S}_{n-1})$ and the latent vector z as the initial hidden state of the decoder. Denoting dec as the output of decoder, we then greedily generate words, *i.e.*,

$$\begin{aligned} w_i^{\boldsymbol{S}_n} &= \arg\max_i P(w_i^{\boldsymbol{S}_n} | dec(\{w_{<i}^{\boldsymbol{S}_n}\}_{(k)}, h_n^{dec})) \\ h_n^{dec} &= [emb(tp_{i_n});\ enc(\boldsymbol{S}_{n-1});\ z_n] \end{aligned} \tag{7}$$

where $\{w_{<i}^{\boldsymbol{S}_n}\}$ denotes the previous word sequence before step i. When $n=0$, we pad $enc(\boldsymbol{S}_{n-1})$ as zeros in h_n^{dec}.

C. Loss Function of CSG

Based on the structure of CSG, the topic sequence generation guides the sentence generation. Accordingly, we can split the training process of CSG into two phases and use two loss functions for CMU and SG.

We use cumulated softmax loss as the loss function for CMU. The loss function $\mathcal{L}^{\text{CMU}}$ for topic sequence $\boldsymbol{TP}_i$ is

$$\mathcal{L}^{\text{CMU}}(\boldsymbol{TP}_i) = -\frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N^{tp}} y_{i,k}(m)\log(tp_{i,k}(m)) \tag{8}$$

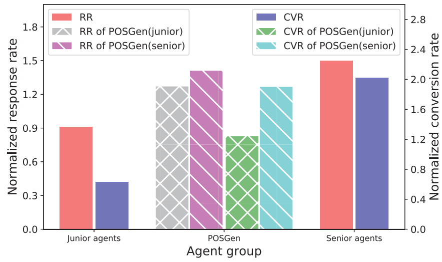

Fig. 6: Validation test results. POSGen improves the performance of junior agents by 1.4x in RR and 1.73x in CVR and achieves competitive results to senior agents.

where K is the topic length of this sample, N^{tp} is the number of topic candidates, and $y_{i,k}$ is the k-th topic label, *i.e.*, $y_{i,k}(m) = 1$ if the m-th topic in N^{tp} is at the k-th place.

For SG, we follow the design in [21] and use empirical lower bound (ELBO) as the loss: $\mathcal{L}^{\text{SG}}(\boldsymbol{S}) = \beta\mathcal{L}_{KL} + \mathcal{L}_{pred}$. $\mathcal{L}_{KL} = -KL(q_\theta(\boldsymbol{z}|\boldsymbol{enc}(\boldsymbol{S}), \boldsymbol{c})||p_\phi(\boldsymbol{z}|\boldsymbol{c}))$ is the loss from KL divergence where β is a dynamic weight on $\mathcal{L}_{KL}$ for mitigating KL vanishing problem [21]. $\mathcal{L}_{pred} = \frac{1}{|\boldsymbol{S}|}\sum_{i=1}^{|\boldsymbol{S}|}\sum_{j=1}^{|\boldsymbol{S}_i|}\log p_\theta(w_j^{\boldsymbol{S}_i}|w_{<j}^{\boldsymbol{S}_i}, \boldsymbol{S}_{i-1}, z_i, tp_i)$ is the loss of reconstructing $\boldsymbol{S}_i$ based on the context c.

VI. Online Evaluation

This section reports the online evaluation of POSGen on a large online insurance platform. We first conduct validation test on a small group of junior agents and senior agents. The test examines the impacts of POSGen on those two groups separately. Then we deploy POSGen to all agents as global test to validate the effectiveness of POSGen in increasing total premium. Following the global test, we conduct a case study on reached users in global test to further understand how POSGen stimulates their interests of insurance purchase.

A. Validation Test

Metrics. In the validation test, we randomly replace a group of human agents by POSGen to reach out for customers, and those who respond to the opening sentences generated by POSGen are handed over to human agents. We use response rate and conversion rate as metrics for the validation test.

Results. Fig. 6 shows the results of validation test. For better illustration, the test group is split into junior and senior. We observe notably improvement by POSGen for the junior group. Specifically, POSGen improves junior's RR by 1.4x and CVR by 1.73x. From the response data, we find the greater improvement in CVR is due to the higher ratio of affirmative responses brought by POSGen. Note that POSGen also achieves similar performance as the senior group. Compared with senior agents, the gap is around 10%. These results demonstrate that POSGen can greatly improve the efficiency of agents in approaching potential customers while guaranteeing the conversion rate. It implies that POSGen may also increase the overall premium brought by the agents, which leads to the global test below.

B. Global Test

Metrics. The global test includes all agents and we evaluate the improvement in total answered users, converted users and premium. The test lasts two months.

Overall performance. Fig. 7 show the results. Among three evaluated metrics, the number of weekly answered users immediately improved after deployment. The average improvement from w5 to w8 is 1.52x, where 1.46x is for junior agents and 1.58x for senior. This is because POSGen is effective in stimulating user responses and can reach more potential customers than human agents. For weekly converted users and premium, the improvement is lagged for around one week. This is because a large portion of users need time to make buying decisions. The gain on weekly converted users and premium is also significant. For converted users, the growth is greatly accelerated, *i.e.*, the improvement of average weekly growth is 11.1x for junior and 2.1x for senior. Based on the growth before POSGen is deployed, the estimated average improvement in converted users is 1.91x. The improvement on weekly premium shows the same pattern, *i.e.*, 2.33x improvement in average. The results demonstrate the effectiveness of POSGen in promoting online insurance sales.

Performance on QoE. We also review the impact of POSGen on the quality of users' experiences (QoE). Consist with our commercial practice, we utilize average talk rounds and the degree of satisfaction as two quantifying metrics. The average talk rounds are automatically derived from our data warehouse. For the degree of satisfaction, it is calculated based on users' feedback and human online check. As shown in Fig. 8, the average talk rounds are increased by 51% after the deployment. The results show that finding users' most concerned topics is more effective in raising their interests to continue the talk than solely focusing on selling products. In addition, the degree of satisfaction slightly increases by 2%. In summary, POSGen increases the insurance sales of agents without decreasing the QoE.

C. Case Study on Users in Global Test

Finally, we conduct a case study on the reached users in the global test. The case study provides details to understand how POSGen improves the final insurance sales.

TABLE II: Users' response of POSGen and agents. Normalized CVR improvement is also listed for each group.

Group	Normalized Proportion Impr (POSGen vs Agents)	Normalized CVR Impr (POSGen vs Agents)
Affirmative	370%	5%
Questionable	210%	21%
Negative	-12%	-0.4%
No response	-11%	0.1%

Metrics. We randomly select 10000 users touched by POSGen (after week 4) and agents (before week 4) for evaluation. The evaluation consists of automatic evaluation, *e.g.*, pattern mining, and human evaluation. The responses are classified into four groups: affirmative, questionable, negative and no response. Affirmative and negative groups refer to whether a user wants to continue the conversation. Questionable group refers to the responses containing specific questions. Note that we mark a user as *no response* if he/she did not respond in three days.

Results. Table II lists the results. The table contains two metrics, *i.e.*, normalized proportion improvement and normalized CVR improvement. Compared with the responded users touched by agents, the proportions of affirmative group

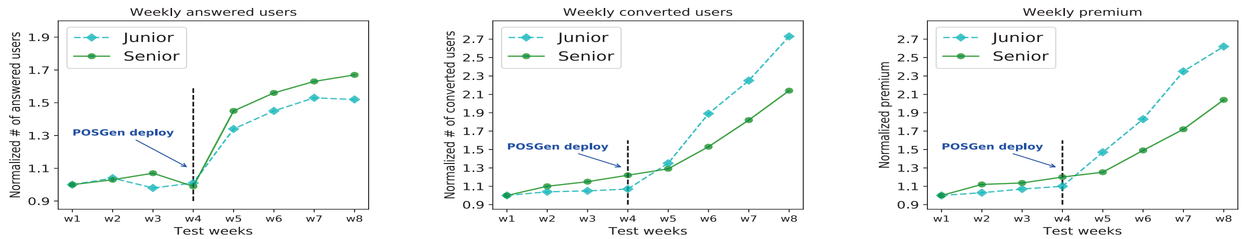

Fig. 7: Global test results. For better comparison, we evaluate junior and senior agents separately after deployment.

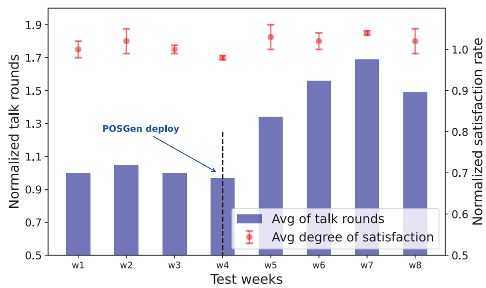

Fig. 8: Quality of users' experiences (QoE) analysis in global test. After deployment, the average talk rounds increased 51%. The average degree of satisfaction is slightly increased by 2%.

and questionable group ares greatly increased, *i.e.*, 370% for affirmative group and 210% for questionable group. The results validate the effectiveness of POSGen in raising users' interests. In addition, we also see that the responded users in affirmative group and questionable group of POSGen have similar Normalized CVR to those of agents. This observation implies that POSGen is able to find more potential users with purchase intention, which are missed by generic opening sentences. The decrease of negative and "no response" users also validate the results.

VII. Conclusion

In this paper, we design POSGen, a personalized opening sentence generation scheme for promoting online insurance sales. POSGen consists of two components: selective attentive transfer learning (SATL) model and context-aware sentence generator (CSG). SATL learns users' favorite topics with limited historical conversation samples by transferring knowledge from auxiliary samples of users' behaviors. The learned topics and user embeddings are then passed to CSG to generate opening sentences with user-specific topic ordering. Extensive offline and online experiments show that POSGen is effective in generating personalized opening sentences and promoting online insurance sales.

VIII. Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 62072208 and International Science and Technology Cooperation Projects of Jilin Province under Grant No 20210402082GH.

References

[1] Research and Markets, "Insurance global industry almanac 2016-2025," Research and Markets, 2021.

[2] FintechNews, "How tencent is trying to fill chinaś us$805 b insurance gap using social media," Fintech News, 2019.

[3] D. Braun, A. H. Mendez, F. Matthes, and M. Langen, "Evaluating natural language understanding services for conversational question answering systems," in *SIGDIAL*. New York, NY, USA: ACM, 2017, pp. 174–185.

[4] R. Yan and D. Zhao, "Coupled context modeling for deep chit-chat: towards conversations between human and computer," in *KDD*. New York, NY, USA: ACM, 2018, pp. 2574–2583.

[5] W. Lei, X. He, M. de Rijke, and T.-S. Chua, "Conversational recommendation: Formulation, methods, and evaluation," in *SIGIR*. New York, NY, USA: ACM, 2020, pp. 2425–2428.

[6] S. H. Lim, Y. Hur, S. Lee, and C. E. Koh, "Role of trust in adoption of online auto insurance," *Journal of Computer Information Systems*, vol. 50, no. 2, pp. 151–159, 2009.

[7] G. Charlton, "How insurance companies can improve ux," INSURANCE UX, 2016.

[8] B. De Souza *et al.*, *Getting Yes Decisions What insurance agents and financial advisors can say to clients.* Houston, TX, USA: Fortune Network Publishing Inc., 2019.

[9] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu, "Plug and play language models: A simple approach to controlled text generation," arXiv preprint arXiv:1912.02164, 2019.

[10] L. Shu, A. Papangelis, Y.-C. Wang, G. Tur, H. Xu, Z. Feizollahi, B. Liu, and P. Molino, "Controllable text generation with focused variation," arXiv preprint arXiv:2009.12046, 2020.

[11] D. Shen, A. Celikyilmaz, Y. Zhang, L. Chen, X. Wang, J. Gao, and L. Carin, "Towards generating long and coherent text with multi-level latent variable models," arXiv preprint arXiv:1902.00154, 2019.

[12] X. Yang, X. Lin, S. Suo, and M. Li, "Generating thematic chinese poetry using conditional variational autoencoders with hybrid decoders," arXiv preprint arXiv:1711.07632, 2017.

[13] R. Puduppully, L. Dong, and M. Lapata, "Data-to-text generation with content selection and planning," *AAAI*, vol. 33, no. 01, pp. 6908–6915, 2019.

[14] J. Li, Y. Song, H. Zhang, D. Chen, S. Shi, D. Zhao, and R. Yan, "Generating classical chinese poems via conditional variational autoencoder and adversarial training," in *EMNLP*. Stroudsburg, PA, USA: ACL, 2018, pp. 3890–3900.

[15] Z. Shao, M. Huang, J. Wen, W. Xu, and X. Zhu, "Long and diverse text generation with planning-based hierarchical variational model," arXiv preprint arXiv:1908.06605, 2019.

[16] Y. Li, Y. Zhang, L. Gan, G. Hong, Z. Zhou, and Q. Li, "Revman: Revenue-aware multi-task online insurance recommendation," in *AAAI*. Palo Alto, CA, USA: AAAI, 2021, pp. 303–310.

[17] C. Chen, M. Zhang, C. Wang, W. Ma, M. Li, Y. Liu, and S. Ma, "An efficient adaptive transfer neural network for social-aware recommendation," in *SIGIR*. New York, NY, USA: ACM, 2019, pp. 225–234.

[18] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *ICANN*. Berlin, Germany: Springer, 2018, pp. 270–279.

[19] L. Zhao, S. J. Pan, E. W. Xiang, E. Zhong, Z. Lu, and Q. Yang, "Active transfer learning for cross-system recommendation," in *AAAI*. Palo Alto, CA, USA: AAAI, 2013.

[20] P. Li and A. Tuzhilin, "Ddtcdr: Deep dual transfer cross domain recommendation," in *WSDM*. New York, NY, USA: ACM, 2020, pp. 331–339.

[21] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *NeurIPS*, vol. 28, pp. 3483–3491, 2015.

[22] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," arXiv preprint arXiv:1511.06349, 2015.

[23] X. Ma, L. Zhao, G. Huang, Z. Wang, Z. Hu, X. Zhu, and K. Gai, "Entire space multi-task model: An effective approach for estimating post-click conversion rate," in *SIGIR*. New York, NY, USA: ACM, 2018, pp. 1137–1140.