Collaborative Data Relabeling for Robust and Diverse Voice Apps Recommendation in Intelligent Personal Assistants

Qian Hu, Thahir Mohamed, Wei Xiao, Zheng Gao,
Xibin Gao, Radhika Arava, Xiyao Ma, Mohamed AbdelHady
Alexa AI
{huqia,thahirm,weixiaow,zhenggao,gxibin,aravar,maxiya,mbdeamz}@amazon.com

Abstract

Intelligent personal assistants (IPAs) such as Amazon Alexa, Google Assistant and Apple Siri extend their built-in capabilities by supporting voice apps developed by third-party developers. Sometimes the smart assistant is not able to successfully respond to user voice commands (aka utterances). There are many reasons including automatic speech recognition (ASR) error, natural language understanding (NLU) error, routing utterances to an irrelevant voice app or simply that the user is asking for a capability that is not supported yet. The failure to handle a voice command leads to customer frustration. In this paper, we introduce a fallback skill recommendation system to suggest a voice app to a customer for an unhandled voice command. One of the prominent challenges of developing a skill recommender system for IPAs is partial observation. To solve the partial observation problem, we propose collaborative data relabeling (CDR) method. In addition, CDR also improves the diversity of the recommended skills. We evaluate the proposed method both offline and online. The offline evaluation results show that the proposed system outperforms the baselines. The online A/B testing results show significant gain of customer experience metrics.

1 Introduction

Intelligent personal assistants such as Alexa, Siri, and Google Assistant have been becoming more and more popular and making people's daily lives convenient. IPAs can fulfill users' requests by answering questions ranging from weather to stock price. To enrich user experience, a large amount of third-party (3P) voice apps (aka skills) have been developed. These voice apps extend IPAs built-in capabilities to better serve customers. They can perform operations like ordering food, playing a game, or helping a user sleep by playing soothing sounds. The supported 3P skills can number up to hundreds of thousands.

IPAs understand user's request using spoken language understanding (SLU) system. The request goes through a series of components to get response, as illustrated in Figure 1. The first component is automatic speech recognition (ASR), which converts speech to its transcription also called utterance. At the second stage, the utterance is interpreted by NLU system. NLU as the critical component of SLU interprets the meaning of an utterance by using several NLP technologies including domain classifier (DC), intent classifier (IC), and named entity recognition (NER). After NLU, the arbiter is responsible to select the most relevant voice app (skill) for a given NLU interpretation. Sometimes the arbiter may fail to find a relevant skill that can handle the user request. It could be a system error such as automatic speech recognition (ASR) error, natural language understanding (NLU) error. Another reason could be that the feature requested by the user is not supported yet by the dialog system or the requested content is not found such as music, video, book, recipe, etc. To reduce customer friction and recover the conversation, we propose a skill recommender system which proactively suggests 3P skills to users for unhandled requests, even if the users are not aware of the skills.

The proposed skill recommender system is composed of two components: a shortlister, and a reranker. Figure 2 shows the system architecture. Given an utterance, the shortlister, also known as the candidate generator retrieves k most relevant skills out of the skill catalog. The retrieved skills are passed to the reranker that ranks the skill candidates by using skill specific information and the utterance. Finally, the top-1 skill is presented to the user. This system is not meant to replace the original NLU or arbiter components. It is specifically designed to serve as a fallback for utterances that are not handled by the existing system (i.e., unclaimed utterances) using the increasing catalog

Proceedings of the Third Workshop on Natural Language Processing for Conversational AI, pages 113–119
November 10, 2021. ©2021 Association for Computational Linguistics

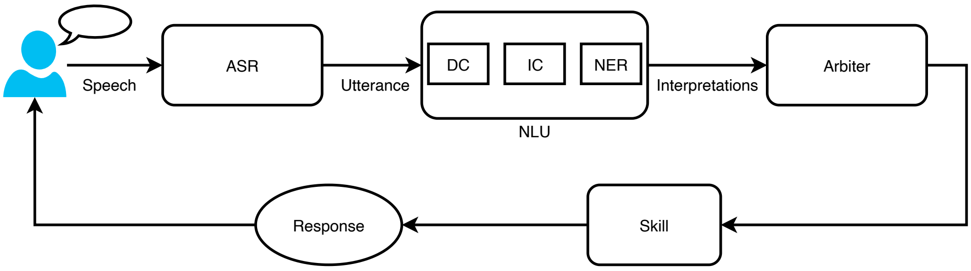


Figure 1: A high-level overview of an IPA.

of 3P skills.

Traditional recommender systems such as video recommendation recommend a ranked list of items to a user. The user scans the list and select the one he/she likes the most (Covington et al., 2016). The feedback from the user is treated as label (accept/reject) for learning a model. However, due to the limitation of voice user interface (VUI), we can only present the top-1 skill to users, as listening to the playback of a long list is tedious and can significantly degrade user experience (Cohen et al., 2004). This limitation results in partial observation problem. Namely, users cannot observe the full recommendation list and make a decision, which imposes difficulties in learning a ranking model. To solve partial observation problem, we propose a novel method called collaborative data relabeling (CDR). CDR mitigates the partial observation problem by trying to answer a counterfactual question, "what if we present another skill to the user?". CDR answers this question by matching a similar request and using feedback from that request to relabel the original ranked list. Recommender systems usually focus on optimizing the accuracy of predictions while ignoring the diversity of the recommended items, which can degrade user experience if similar items get recommended over and over again (Ekstrand et al., 2014; Castagnos et al., 2013; Knijnenburg et al., 2012; Ziegler et al., 2005; Willemsen et al., 2016). CDR improves the diversity of recommended skills by relabeling different skill candidates that serve the same intent. The relabeled skills force the model to learn to diversify their prediction distribution among multiple skill candidates.

At the beginning, we do not have data for training the model. To collect training data, we build a rule-based system. Similar to the proposed system, the rule-based system also has a two-stage architecture. We use the data collected from this system to train and evaluate our proposed model offline. The proposed model is put into production for online A/B testing after it has achieved satisfying offline results. Online experimental results show significant gains of user experience metrics such as higher volume of acceptances, lower friction rates, etc.

Overall, the contributions of this work are summarized as following:

- We propose a skill recommender system for IPAs to handle unclaimed utterances by exploiting the ever-increasing 3P voice apps.
- To mitigate partial observation issue, we propose collaborative data relabeling inspired by causal inference. Collaborative data relabeling also has the advantage of improving recommendation diversity and thus improving user satisfaction. Suggesting diverse skills to users can help them explore and discover more skills, which is also beneficial to third-party skill developers.
- We conduct offline and online experiments. Online experimental results show significant gains of user experience metrics.

2 Skill Recommender System

Our skill recommender system consists of two components, shortlister and reranker, as shown in Figure 2.

Given the input utterance text, shortlister selects top-k relevant skills from the skill catalog. We implement shortlister as a keyword-based search engine. To build the skill search engine, we index skill metadata including skill name, skill descriptions, example phrases, and invocation phrases. At retrieval time, the relevancy score between an utterance and a skill is computed as the sum of TF-IDF score (Rajaraman and Ullman, 2011) of every word in the utterance. The skills with top k relevancy scores are returned.

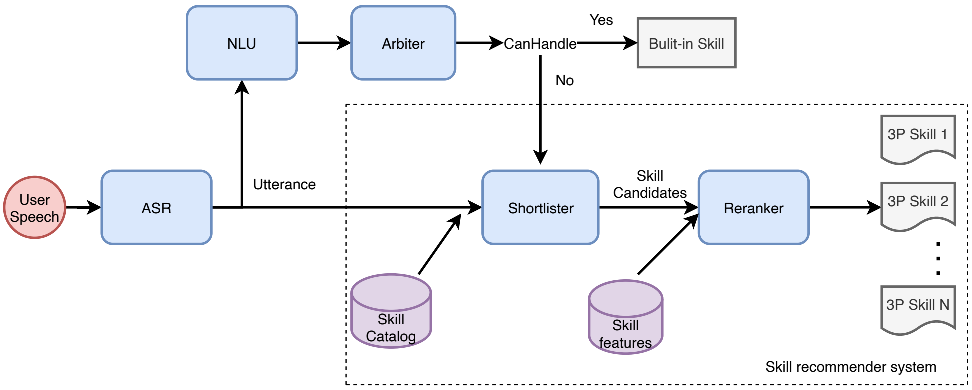


Figure 2: A overview of skill recommender system.

The reranker model takes in the skill candidates generated by shortlister and returns a ranked list of skills based on the utterance and skill specific information. The reranker is a deep learning model with a listwise ranking loss function. Figure 3 shows the reranker model architecture. The utterance is encoded by a BERT encoder (Devlin et al., 2018). The features of skills include skill id, skill name, and skill score returned by shortlister. Skill id is represented using an embedding vector; skill name is encoded into an embedding vector by BERT. The skill score feature is converted into a bin and encoded as an embedding vector. The skill feature embedding vectors are concatenated to form a single embedding vector. The utterance embedding vector is concatenated with every skill embedding vector to form a sequence of utterance-skill embedding vectors. As the skill candidates returned by shortlister is ordered by relevance score, to capture such sequential information, these sequence of embedding vectors are put into a Bi-LSTM layer (Hochreiter and Schmidhuber, 1997). The outputs from the Bi-LSTM layer is converted to probability scores by using softmax function. Each skill has a corresponding probability score. The skills are reranked according to the predicted probability scores. For a list of skill candidates, the user feedback for the skill candidates is $\mathbf{y} = \{y_1, ..., y_k\}, y_i \in \{0, 1\}$ indicating whether the user accepts or rejects the skills and the predicted probabilities by the reranker model is $\mathbf{s} = \{s_1, ..., s_k\}$. We use listwise ranking objective function (Cao et al., 2007). The objective function is formulated as

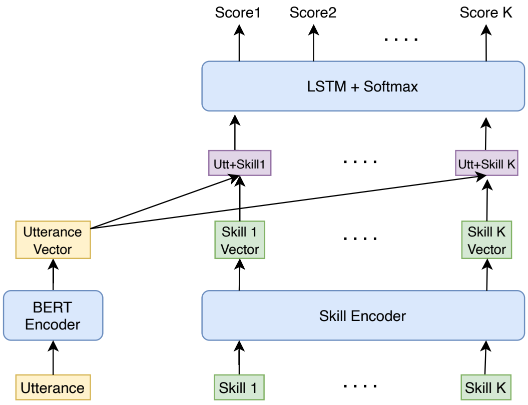


Figure 3: Model architecture of reranker.

$$L(\mathbf{y}, \mathbf{s}) = \frac{1}{k} \sum_{i=1}^{k} [-y_i \log(s_i) - (1 - y_i) \log(1 - s_i)]. \quad (1)$$

Due to the partial observation issue, the labels of the unobserved skills are treated as negative. However, this assumption is not realistic and can bias the model, because the missing values are not necessarily negative. In the next section, we introduce collaborative data relabeling to mitigate this issue.

3 Collaborative Data Relabeling

Compared to traditional recommender system such as video recommendation where users view the full recommended list and select the best one they like, the skill recommender system has its unique challenge. Limited by VUI, we can only present the top-1 ranked skill to user, which results in a partial observation problem. With partial observation, users have no chance to view and compare other skills in the list. We do not know if the user would like the other skills more than the top-1. Without comparing the top-1 skill with the other skills, it is hard to learn a ranking model, as ranking in essence is about comparing. To solve partial observation problem, we propose collaborative data relabeling (CDR) approach.

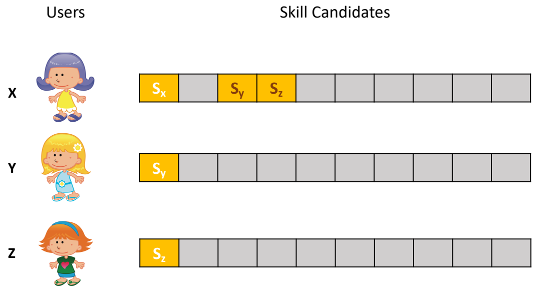


Figure 4: Illustration of collaborative data relabeling. In skill recommendation, only the top-1 skill is presented to users. Given a user x who invoked voice assistant with some utterance, to know her responses to skills s_y and s_z that were not presented to her, we found two users y and z who spoke similar utterances and were suggested with skills s_y and s_z, respectively and use their responses to relabel user x's feedback to skills s_y and s_z.

The intuition of CDR is to answer a counterfactual question, namely, "what if we had presented another skill to user?". To answer this question, we find k nearest neighbors of a user request (utterance) and use their feedback to relabel the original ranked list of skill candidates, which is inspired by matching method (Stuart, 2010) in causal inference. In causal inference, matching is an approach to estimate the treatment effect by comparing the treated units to non-treated units with similar characteristics. CDR has a similar working mechanism. Given a user utterance, to know the user's response to an unpresented skill, we find a similar utterance whose invoker has interacted with that skill and use his/her response to relabel it as either positive or negative, as illustrated in Figure 4. Usually, there are more than one neighbors that have interacted with the skill, in which case we use majority vote to decide the final label. As the final labels are decided by multiple neighbors, they are more reliable than those of just one user, which results in learning a more robust model.

Recommender systems are confronted with an over-fitting problem that only a small portion of items are recommended to users (Kunaver and Požrl, 2017), which can hurt user satisfaction as they can quickly get bored by always being suggested with similar types of items. This problem is especially relevant for skills that serve the same intent with different content. For example, when users ask to play a soothing sound to help them sleep, always suggesting the same sleep sound can get users bored, while there exist many types of sleep sounds in the skill store such as frog, ocean, rain, waterfall sleeping sound, etc. Suggesting diverse skills can improve user satisfaction (Castagnos et al., 2013). The proposed CDR method improves diversity by relabeling different skill candidates as positive, which forces the model's predictive distribution to be dispersed among more skills. Diversified suggestions can lead to drop in accuracy (McNee et al., 2006; Ziegler et al., 2005), which imposes difficulties in faithfully evaluating the real user satisfaction metrics. To evaluate how diversity can influence user satisfaction, we use manual annotation. The detail of manual annotation schema will be explained in Section 5.1.2.

To capture semantic meaning of utterances for similarity measurements, we use fine-tuned BERT encoder to encode utterances into embedding vectors. The BERT model is fine-tuned using data with a multi-task objective function, specifically, intent classification and named entity recognition. We also experiment with pre-trained BERT encoder and find that it does not work well for capturing semantics of an utterance, which has also been discovered by several works such as (Reimers and Gurevych, 2019) and (Li et al., 2020). We use the average pooling of the contextual embedding vectors in the last layer as the utterance embedding vector. We measure the similarity score between two utterances using cosine similarity between their embedding vectors.

4 Experiments

4.1 Data Collection

At the beginning, we do not have data to train and evaluate our system. To collect data, we build a rule-based system which has similar architecture as our proposed one. The rule-based system uses the same shortlister but a rule-based reranker. The rule-based reranker ranks the skill candidates by using their historical acceptance rates. The skill with the highest acceptance rate is selected. To ensure high quality of recommendation, we only suggest the top skill to customer if its acceptance rate is higher than 0.5. We collect two-month data from a commercial voice assistant traffic for model training and evaluation. The data of the last week is used for testing. The data of the second last week is used for validation. The remaining is used for training. The proportions of training, validation and testing data are around $80\%, 10\%, 10\%$, respectively. Each data sample is composed of an utterance u_t, forty skill candidates $s_{t,1}, ..., s_{t,40}$ generated by shortlister, and ground truth label y_t, denoted as $(u_t, (s_{t,1}, ..., s_{t,40}), y_t)$, where $y_t \in \{s_{t,1}, ..., s_{t,40}\} \cup \{N\}$, N is null which means all the skill candidates are rejected by the user. Note that for the sake of customer privacy, the data is de-identified and we are not able to know the identify of the user from the data.

4.2 Collaborative Data Relabeling

For CDR, the k-nearest neighbors of an utterance is found from the training data. And only the training data is relabeled. We keep the labels of the validation and testing data as it is. When relabeling the skill candidates of an utterance, we select up to 200 neighbors and keep those whose similarity score is above a certain threshold s. To avoid bringing noisy labels from neighbors, a skill candidate is relabeled if the number of its supportive neighbors are higher than n. The supportive neighbors of a skill are the neighboring utterances which relabel it as positive. The intuition is that if there are multiple neighbors confirming a skill candidate, the relabeled skill is reliable. We treat s and n as hyperparameter and choose the best value by using validation dataset.

4.3 Evaluation Metrics

To simulate the real application, we evaluate the model by selecting the top-1 skill with predicted probability higher than 0.5 and compare it with the ground truth label. The evaluation metrics are F1 scores, namely, F1_1 and F1_2 which are harmonic means of Recall and Precision_1, Precision_2, respectively. Precision_1 calculates the number of correct predictions over all the predictions, while Precision_2 means the number of correct predictions over all the non-empty predictions. Recall calculates the number of correct predictions over the number non-empty ground truth labels. Due to company policy, in this paper, we report relative performance numbers.

5 Experimental Results

5.1 Collaborative Data Relabeling

5.1.1 Comparative experimental results

Figure 5 shows the relative performance improvements of collaborative data relabeling method with the change of hyperparameters, the number of support n and similarity threshold s. The baseline model is listwise reranker model trained with the original training data. From the figures, we can see that with the increaseing of the number of support and similarity score, the F1 scores are becoming higher. With the increasing of s, the relabels we obtain are from closer neighbors which tend to bring cleaner labels. When the similarity score is lower, the two utterances are less similar, which even leads to wrong labels. With the increasing of n, we require more neighbors to confirm the relabeling of a skill candidate, which leads to higher quality of labels. Overall, the performance of the models trained on relabeled data is higher than that of the baseline model.

Table 1: Manual annotation results.

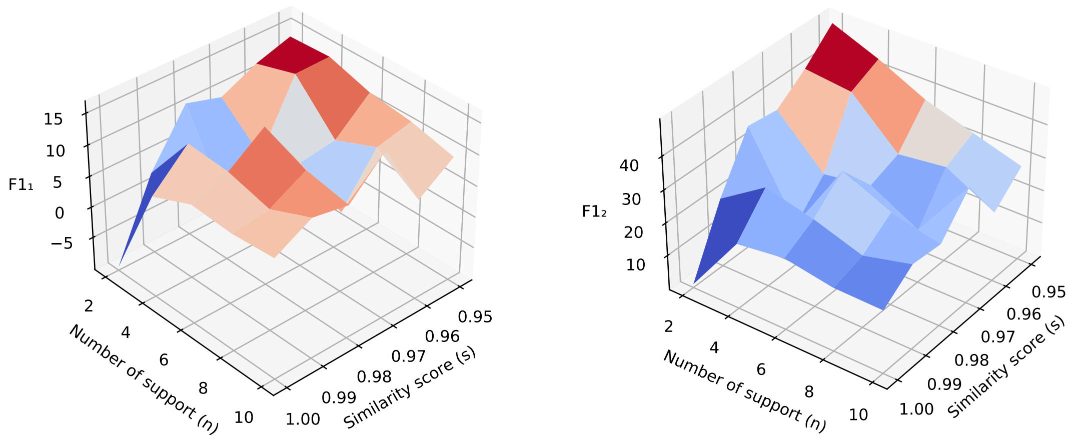
Model	Accuracy	Score	#suggestions
CDR-based model	+19.21%	+62.73%	+112.13%

5.1.2 Manual evaluation

To evaluate how CDR impacts the model performance, we manually compare the relabeled model against the baseline model. We randomly sampled 2500 samples and asked human annotators to check the suggested skills by relabeled and baseline models. We use two types of evaluation metrics. The first one is accuracy which is the number of correct predictions divided by the total number of predictions. As the model only makes a suggestion if the predicted probability is higher than 0.5, the model can reject to make a suggestion if it is not confident enough. To compare no suggestion with suggestion, we use score. A model gets a score by the following rules:

- If the model's prediction is correct, it gets a score.
- If the model doesn't make a suggestion: a) if the other model makes a wrong prediction, the current model gets a score; b) if the other model makes a correct prediction, the current model does not get a score. The intuition is that no suggestion is better than a wrong suggestion and a correct suggestion is better than no suggestion.

Table 1 shows the evaluation results based on manual annotation. The baseline model is the model trained on original data. From the results, we can see that the relabeled model has higher accuracy, which indicates higher user satisfaction. The relabeled model also gets higher score and make more suggestions.



(a) F1$_1$

(b) F1$_2$

Figure 5: Experimental results of collaborative data relabeling method by varying the number of support and similarity threshold. The unit of vertical axis is percentage.

5.2 Online Experiments

After seeing performance gains in offline experiments, we put our model into online A/B testing. We compare it with the rule-based heuristic model. The online experiments show that the proposed model reduced friction rate by 0.35%. Friction means the circumstances where the voice assistant does not understand the user and cannot act on the user's request. The number of accepted skills increased by 5.86%. The average number of new skills enabled per customer increased by 0.98%. A skill has to be enabled before it can be used by the customer. In addition, the number of unique suggested and accepted skills increased by 233% and 98.75%, respectively, which indicates that the new model makes more diverse suggestions than the legacy system.

6 Conclusions and Future Work

In this work, we proposed a skill recommender system to suggest skills for unhandled voice commands in intelligent personal assistants that aims to reduce the user friction and recover the conversation. Compared to traditional recommender systems, the skill recommender system faces the challenge of partial observation. To resolve this challenge, we proposed collaborative data relabeling. In addition, it improves the diversity of recommended skills. Collaborative data relabeling as a simple and effective approach is especially useful for industrial deployment. We evaluated the proposed system offline before put it online for A/B testing. The online experimental results showed significant gains of user experience metrics. In the future, we will try contextual bandits and let the model learn to explore the under-exploited skills.

References

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Sylvain Castagnos, Armelle Brun, and Anne Boyer. 2013. When diversity is needed... but not expected!

Michael H Cohen, Michael Harris Cohen, James P Giangola, and Jennifer Balogh. 2004. *Voice user interface design*. Addison-Wesley Professional.

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Michael D Ekstrand, F Maxwell Harper, Martijn C Willemsen, and Joseph A Konstan. 2014. User perception of differences in recommender algorithms. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 161–168.

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

Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):441–504.

Matevž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems–a survey. *Knowledge-Based Systems*, 123:154–162.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Sean M McNee, John Riedl, and Joseph A Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 1097–1101.

Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of massive datasets*. Cambridge University Press.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Elizabeth A Stuart. 2010. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1.

Martijn C Willemsen, Mark P Graus, and Bart P Knijnenburg. 2016. Understanding the role of latent feature diversification on choice difficulty and satisfaction. *User Modeling and User-Adapted Interaction*, 26(4):347–389.

Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32.