Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 581–589
Marseille, 11–16 May 2020

Getting To Know You: User Attribute Extraction from Dialogues

Chien-Sheng Wu†, Andrea Madotto, Zhaojiang Lin, Peng Xu, Pascale Fung
Center for Artificial Intelligence Research (CAiRE)
The Hong Kong University of Science and Technology
† Salesforce Research
wu.jason@salesforce.com, [amadotto,zlinao,pxuab]@connect.ust.hk, pascale@ece.ust.hk

Abstract

User attributes provide rich and useful information for user understanding, yet structured and easy-to-use attributes are often sparsely populated. In this paper, we leverage dialogues with conversational agents, which contain strong suggestions of user information, to automatically extract user attributes. Since no existing dataset is available for this purpose, we apply distant supervision to train our proposed two-stage attribute extractor, which surpasses several retrieval and generation baselines on human evaluation. Meanwhile, we discuss potential applications (e.g., personalized recommendation and dialogue systems) of such extracted user attributes, and point out current limitations to cast light on future work.

Keywords: Dialogue Systems, Personalization, Information Extraction, Natural Language Processing

1. Introduction

User attributes are explicit representations of a person's identity and characteristics in a structured format. They provide a rich repository of personal information for better user understanding in many applications. High-quality user attributes are, however, hard to obtain since the information in social networks such as Facebook and Twitter is often sparsely populated (Li et al., 2014). Therefore, exploiting unstructured data sources to obtain structured user attributes is a challenging research direction.

Meanwhile, there is an increasing reliance on dialogue agents to assist, inform, and entertain humans, for example, keeping the elderly company and providing customer service. Conversational data between users and systems is informative and abundant, and most of the existing deep learning approaches are trained on these large crowd-sourced corpora or scraped conversations. These models, given the current dialogue context (e.g., few previous turns), are focused on either generating good responses (Serban et al., 2015), or incorporating "system attributes" to generate consistent responses (Zhang et al., 2018; Mazare et al., 2018). However, the whole dialogue history of the same person is ignored, implying that these systems are not gradually getting to know their users by extracting user information through conversations.

In this paper, we demonstrate that it is feasible to automatically extract user attributes from dialogues. Given a user utterance, our goal is to predict user information that can be represented as a *(Subject, Predicate, Object)* triplet format, which is available for any downstream application. For example, in Table 1, *(I, live_in, Florida)* is extracted from the second user utterance. Meanwhile, not every utterance has useful information, and some have multiple attributes. For instance, "How are you doing today?" does not have any user-specific information, but from the fourth user utterance in Table 1, we can conclude that the user has a son, likes to go to church, and has a Ford car. Additionally, unlike standard information extraction tasks, where the extracted information is tagged within the input, some user attributes must be inferred indirectly. For example, "My son is afraid of talking to others" implies that the user's son is a shy person.

	Conversations	User Attributes
Usr	Hello, how are you doing today?	none
Sys	I am fine! Where do you live?	
Usr	I am originally from California but now I live in Florida for long.	(I, live_in, Florida)
Sys	Florida! You must have a good work-life balance.	
Usr	Oh, I no longer work at banks but for exercise I walk often.	(I, previous_profession, banker) (I, has_hobby, walking)
Sys	Good to hear that! Do you live with your family?	
Usr	My son. I bring him to church every Sunday with my Ford.	(I, has_children, son) (I, like_goto, church) (I, have_vehicle, ford)
Sys	Wow sounds good! You can meet many people.	
Usr	Sure, but my son is afraid of talking to others.	(My son, misc_attribute, shy)

Table 1: The conversation column is a daily dialogue between a user and a system. The user attributes column is the potential extracted user information.

Since no conversational dataset is available for our purpose, we leverage the state-of-the-art natural language inference (NLI) model to train our model via distant supervision. Using the existing Persona-Chat dataset (Zhang et al., 2018), comprising dialogues collected given artificial speaker information called personas, we hypothesize that if an utterance is entailed by a persona sentence, then such a persona sentence can be viewed as a valid user attribute. For example, if the persona sentence "I was a banker" is entailed by the user utterance "I no longer work at banks," then we can extract the *(I, previous_profession, banker)* attribute for the utterance. Although NLI mapping may include some noise, these annotations are cheap and can at least provide a weak source of supervision.

We view user attribute extraction as a pipeline of two tasks: the predicate prediction task and entity generation task. The predicate prediction task first determines whether there is a predicate triggered by a user utterance. This is considered as a multi-label classification problem because there could be zero or multiple attributes. If there is a triggered predicate, then the entity generation task further generates

Persona A	Persona B
I just bought a brand new house.	I love to meet new people.
I like to dance at the club.	I have a turtle named Timothy.
I run a dog obedience school.	My favorite sport is the ultimate frisbee.
I have a big sweet tooth.	My parents are living in Bora.
I like taking and posting selkies.	Autumn is my favorite season.
Conversation	
[A] Hi, I just got back from the club.	
[B] Cool, this is my favorite time of the year season wise.	
[A] I would rather eat chocolate cake during this season.	
[B] What club did you go to? Me and Timothy watched TV.	
[A] I went to club Chino. What show are you watching?	
[B] We watched a show about animals like him.	
[A] I love those shows. I am really craving cake.	
[B] Why does that matter any? I went outdoors to play frisbee	
[A] It matters because I have a sweet tooth.	

Table 2: A conversation from the Persona-Chat dataset. Two different personas are provided before they have the conversation below.

the subject and object phrases to complete the whole user attribute. The subject phrase indicates the "who" information, and the object phrase contains the "what" information. We empirically show that our strategy outperforms several retrieval and generation baselines on human evaluation. Our contributions are summarized as follows: [1]

- We are the first to extract user attributes from chit-chat dialogues, which contain strong evidences to suggest users information.
- We propose a two-stage attribute extractor that surpasses baselines on human evaluation. We train our model via distant supervision, leveraging an NLI model to obtain cheap and effective training samples.
- We discuss potential applications of the extracted user attributes and point out current limitations to cast light on future research directions.

2. Distant Supervision Data

There are no existing dialogue datasets with the labels required for the attribute extraction task. Hence, we leverage two datasets, Persona-Chat (Zhang et al., 2018) and Dialogue NLI (Sean et al., 2018), to generate distant supervision data. We briefly introduce these datasets and discuss some of their limitations.

Persona-Chat This is a multi-turn chit-chat corpus with annotation of the participants' personal profiles (e.g., preferences about food, movies). It is collected by asking two crowd-workers to talk to each other freely but conditioned on their artificial personas, which are established by four to six persona sentences. An example from the dataset is provided in Table 2. In total there are 1155 personas with over 5,000 persona sentences, and 162,064 utterances over 10,907 dialogues. Most of the related works using this dataset (Weston et al., 2018; Semih Yavuz, 2018; Wolf et al., 2019; Dinan et al., 2019) focus on adapting systems to a given persona, i.e., learning to generate responses that are consistent with the persona.

[1]The code is released at `https://github.com/jasonwu0731/GettingToKnowYou`

Although the dataset contains pre-defined personas and the corresponding conversations, it cannot be applied directly to the attribute extraction task for the following two reasons: 1) The mapping between utterances and the persona is missing. Which persona sentence is related to which utterance remains unknown. 2) All the personas are written in natural language instead of in a structured format. Natural language description is not easy-to-use for downstream tasks.

Dialogue NLI This is a new dataset built upon Persona-Chat (Zhang et al., 2018), which provides a corpus for NLI task in dialogues. The authors demonstrate that consistency of dialogue agents can be improved by re-ranking responses using an NLI model. Dialogue NLI consists of sentence pairs labeled as entailment, neutral, or contradiction. For example, in Table 2, the persona sentence "I like to dance at the club" for persona A is entailed with the utterance "I just got back from the club."

The authors first require human annotation of all the persona sentences in Persona-Chat, mapping into the triplet *(e_1, r, e_2)*, where e_1 and e_2 are entities and r is the relation types. They pre-define around 60 different relation types such as *live_in_general, like_food*, and *dislike*. Subsection 2.1. shows all the relation types considered in this paper. For example, the persona sentence "I just bought a brand new house" is labeled to the triplet *(I, own, house)*. Then they group different persona sentences with the same triplet together. Thus sentences in the same group are considered as entailment, and others as neutral and contradiction.

A drawback is that the dataset does not have a human-annotated triplet for each utterance. The authors assign a triplet to an utterance by the following criteria: 1) if its object (e_2) is a sub-string of the utterance or 2) if word embedding similarity between the utterance and the persona sentence is suitably large. In this way, they can retrieve a small portion of the utterances that are potentially entailed, but noise is introduced to the dataset and many utterances remain unlabeled.

Since their goal is only to create an NLI dataset, with the strategy mentioned above, the authors are able to collect a large number of training samples. On the other hand, our goal is to extract structured attributes from the utterances, and we need as many training samples as possible to learn the mapping. Therefore, we need a method to help us find the mapping of the unlabeled utterances.

2.1. Relation Types

We show all the relation types used in the original dataset and our setting: place origin, live in city state country, live in general, nationality, employed by company, employed by general, has profession, previous profession, job status, teach, school status, has degree, attend school, like general, like food, like drink, like animal, like movie, like music, like read, like sports, like watching, like activity, like goto, dislike, has hobby, has ability, member of, want do, want job, want, favorite food, favorite color, favorite book, favorite movie, favorite music, favorite music artist, favorite activity, favorite drink, favorite show, favorite place, favorite hobby, favorite season, favorite animal, favorite

sport, favorite, own, have, have pet, have sibling, have children, have family, have vehicle, physical attribute, misc attribute, has age, marital status, gender, other.

2.2. Combination Strategy

Our strategy is to combine Persona-Chat and Dialogue NLI. We hypothesize that by combining these two datasets, if a user utterance and a persona sentence are positively entailed, then the persona triplet of that persona sentence can be represented as one of the possible user attributes. For example, if the utterance "I prefer basketball; team sports are fun" and the persona sentence "I like playing basketball" has an entailment relationship, then we assign the triplet of the persona sentence labeled by Dialogue NLI, which is *(I, like_sports, basketball)*, to be one of the user attributes.
We train an NLI model using the Dialogue NLI corpus, and the trained model can be used as a scorer to predict the entailment score. We fine-tune BERT (Devlin et al., 2018), [2] a recently proposed pre-trained deep bidirectional Transformer (Vaswani et al., 2017), to predict entailment given two sentences as input. This scorer achieves 88.43% test set accuracy on Dialogue NLI, which is aligned (slightly better) with the best-reported model, ESIM (Chen et al., 2017), with 88.2% accuracy.

3. Methodology

Let us define N utterances in a dialogue as $U = \{u_1, \ldots, u_N\}$, where odd and even turns are represented as user utterances and system responses. M natural language persona sentences $P = \{p_1, \ldots, p_M\}$ in the dataset have their corresponding triplets $T = \{t_1, \ldots, t_M\}$. Besides persona sentences, each of the utterances may have zero, one or multiple triplets selected from T. We design a two-stage attribute extractor to obtain *(subject, predicate, object)* triplets from dialogues using a context encoder, a predicate classifier, and an entity generator.

3.1. Two-stage Attribute Extractor

To predict the user attributes, we use a context encoder to capture utterance semantics. Then instead of directly generating triplets, we predict all the triggered predicates first. Next, an entity generator decodes multiple times for every triggered predicate to obtain their corresponding subject and object phrases. For example, in Figure 1, three predicates *(have_vehicle, like_goto, has_children)* are triggered by the predicate classifier. Given *have_vehicle* as input to the entity generator, the subject "I" and the object "Ford" will be generated.

Context Encoder The context encoder takes a sequence of word embeddings as input and obtains a set of fixed-length vectors $H = (h_1^{enc}, \ldots, h_l^{enc}) \in \mathbb{R}^{l \times d_{hdd}}$ by bi-directional gated recurrent units (GRUs), where l is the number of words in the utterance and d_{hdd} is the hidden size of the GRU. The last hidden state h_l^{enc} is represented as the final encoded vector, which will be used to query the predicate classifier and initialize the entity generator.

[2] PyTorch version in `github.com/huggingface/pytorch-pretrained-BERT`

Predicate Classifier We use a multi-hop ($K = 3$ hops) end-to-end memory network (MN) (Sukhbaatar et al., 2015) as our predicate classifier because we believe its reasoning ability can benefit predicates prediction, as shown in question answering and dialogue tasks (Bordes et al., 2016; Wu et al., 2018; Madotto et al., 2018; Wu et al., 2019b). We assign the memory in the MN as all the predicate words $R = \{r_1, \ldots, r_J\}$, where J is the total number of possible predicates. The predicate classifier is queried by the encoded vector h_l^{enc}, and the memory attention at each hop k is computed as

$$\alpha^k = Softmax(C^k(P)q^k) \in \mathbb{R}^J, \quad (1)$$

where C^k and q^k are the embedding matrix and query vector at hop k, respectively. Here, α^k is a soft memory selector that decides the memory relevance with respect to the query vector q^k. The model reads out the memory o^k as

$$o^k = \sum_i \alpha_i^k C^{k+1}(r_i) \in \mathbb{R}^{d_{hdd}}. \quad (2)$$

Then the query vector is updated for the next hop using

$$q^{k+1} = q^k + o^k \in \mathbb{R}^{d_{hdd}}. \quad (3)$$

In order to perform multi-label classification, instead of taking the *Softmax* function, as in the original MN, to obtain the probability distribution, we replace the *Softmax* layer with a *Sigmoid* layer in Eq.1 at the last hop. In this way, each of the predicates is triggered separately, and we can predict whether multiple predicate will be triggered, or none of them will be triggered.

Entity Generator If a predicate is triggered, our entity generator will generate the corresponding subject and object phrases to complete the final user attribute. Note that both the subject and object can have more than one word, and we manually concatenate them into one sequence separated by a semicolon. For example, we train our model to generate a sequence "my son; shy" if the triplet is *(my son, misc_attribute, shy)*.
Motivated by the multilingual neural machine translation work (Johnson et al., 2017) that uses a single model for all languages but with different start-of-sentence tokens, we also use a single entity generator for all the predicates. If there are multiple predicates triggered, we decode multiple times using the same parameters for the entity generator with different predicates as input. In this way, we expect our model to transfer knowledge between different predicate generations.
The first input token of the entity generator is one of the triggered predicates. At decoding time step t, the generator GRU takes a word embedding w_t as the input and returns a hidden state h_t^{dec}. The output word distribution P_t^{final} is the weighted-sum of two distributions,

$$P_t^{final} = P_{gen} P_t^{vocab} + (1 - P_{gen}) P_t^{source}, \quad (4)$$

where $P_t^{vocab} = Softmax(W_1 h_t^{dec})$ is the mapping from the generator hidden states to the vocabulary space using trainable matrix W_1, and $P^{source} = Softmax(H h_t^{dec})$

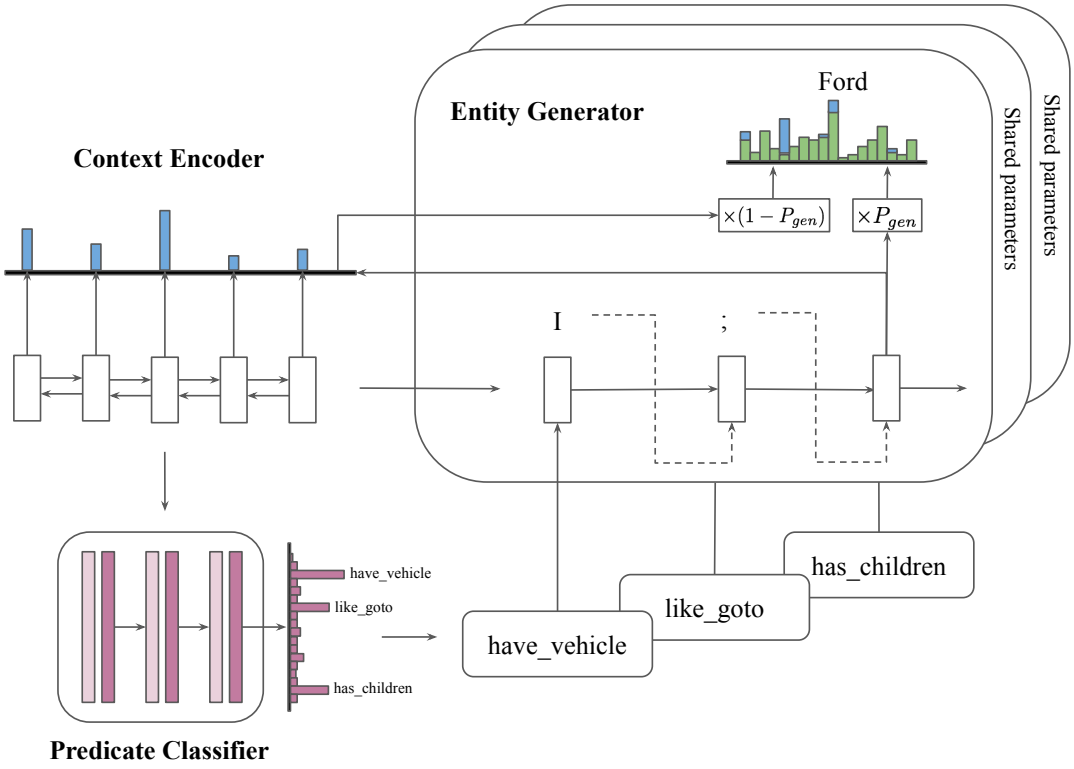

Figure 1: The proposed attribute extractor, which has a context encoder, a predicate classifier, and an entity generator. The generator will decode multiple times for every triggered predicate.

is the attention weights of the input. The scalar P_{gen} is learned to combine the two distributions,

$$P_{gen} = Sigmoid(W_2[h_t^{dec}; w_t; v_c]), \quad (5)$$

where W_2 is a learned matrix and $v_c = \sum P_i^{source} * h_i^{enc}$ is the context vector.

3.2. Objective Function

We use the user attributes obtained from the NLI model as the distant supervision labels. During training, we optimize the weighted-sum of two loss functions end-to-end, one for the predicate classifier and the other for the entity generator. The former computes a binary cross-entropy loss L_p between the predicate attention (α^K) and the expected ones (R^{label}) as

$$L_p = -\sum_i [R_i^{label} \times \log \alpha_i^K + (1 - R_i^{label}) \times \log (1 - \alpha_i^K)]. \quad (6)$$

The latter computes standard cross-entropy loss L_v between the generated sequence (P^{final}) and the true subject and object values (defined as Y^{label}) as

$$L_v = -\sum_t \log(P_t^{final}(Y_t^{label})). \quad (7)$$

Lastly, we optimize the whole model using the weighted-sum of two losses by a hyper-parameter λ. The final objective function is

$$Loss = \lambda L_p + (1 - \lambda) L_v. \quad (8)$$

4. Experimental Setup

4.1. Training Details

The attribute extractor is trained using the Adam optimizer (Kingma and Ba, 2014) with batch size of 32. The learning rate annealing starts from 0.001 to 0.0001, and a 0.6 dropout ratio is used. All the embeddings are initialized by concatenating Glove embeddings (300) (Pennington et al., 2014) and character embeddings (100) (Hashimoto et al., 2016). The λ to weight two losses is set to be 0.5. A greedy search decoding strategy is used for our entity generator since the generated phrases are usually short. In addition, to increase model generalization and simulate an out-of-vocabulary setting, a word dropout is applied to the input by randomly masking a small number of input source tokens into unknown tokens.

4.2. Baselines

We compare our model with the following implemented baselines: the sequence-to-sequence (Seq2Seq) model (Sutskever et al., 2014), the pointer-generator (PG) model (See et al., 2017), and the key-value memory networks (KVMN) (Miller et al., 2016). Meanwhile, existing OpenIE models, which parse sentences and tag parts of them as output, could be an alternative. We compare our model with two state-of-the-art open information extraction (OpenIE) pre-trained models, S-OpenIE (Stanovsky et al., 2018) and LLS-OpenIE (Angeli et al., 2015).

Seq2Seq, PG, and KVMN are used for internal comparison, where all the models are trained from scratch using

the distant supervision data. S-OpenIE and LLS-OpenIE, on the other hand, are used for external comparison, where these two models are trained on several OpenIE datasets and evaluated on the attribute extraction task. We briefly introduce the baselines:

- **Seq2Seq** is the most common baseline for sequence generation. We use GRUs as a base model to encode a sequence of words and decode a sequence that concatenates *(subject, predicate, object)* by semicolons.
- **PG** is one of the best generation models that can copy words from the source text via a pointing mechanism. It computes two distributions (input distribution and vocabulary distribution) and combines them automatically.
- **KVMN** is one of the best neural retrieval models that use memory networks to perform key hashing and value reading. It stores all the pre-defined user attributes in the memory and performs multiple hops before final prediction.
- **S-OpenIE** enables a supervised learning approach to the OpenIE task. It formulates OpenIE as a sequence tagging problem. A bi-LSTM transducer and semantic role labeling models are used to extract OpenIE tuples.
- **LLS-OpenIE** first learns a linguistically-motivated classifier to split a sentence into shorter utterances, and produce coherent clauses which are logically entailed by the original sentence.

4.3. Evaluation Metrics

4.3.1. Human Evaluation

True attributes are not available even in the test set, therefore, we conduct a human evaluation to verify the generated attributes. Randomly selected utterances from the test set are annotated by three people on Amazon Mechanical Turk. Turkers are asked to label "1" if the attributes can be inferred from the utterance, and otherwise label "0".

We provide several examples to the turkers to let them understand better the task. We ask the turkers that "Given the sentence said by someone, can the information listed be inferred?", and provide them one sentence and the extracted attributes. We provided some examples for selecting "YES" as below:

- "Sentence": "i am really craving cake."; "Information": ['i', 'like_food', 'cake']
- "Sentence": "being an old man, i am slowing down these days"; "Information": ['i', 'gender', 'male']
- "Sentence": "i am great. i just got back from the club."; "Information": ['i', 'like_activity', 'dancing']

And some examples for selecting "No" as well:

- "Sentence": "amazon is my favorite store."; "Information": ['i', 'employed_by_company', 'amazon']
- "Sentence": "i never have juice , just water."; "Information": ['i', 'dislike', 'water']

	ACC	F1	BLEU-1	Human
Seq2Seq	7.36	21.57	41.94	31.02
PG	11.80	22.99	46.14	37.58
KVMN	25.37	27.32	40.98	52.01
Ours	**26.52**	**28.68**	**51.87**	**67.11**
*Gold**	-	-	-	*79.80*

Table 3: Results on user attribute extraction. Our model achieves the highest human evaluation score (statistically significant), outperforming other generation and retrieval models. * Note that the *Gold* row is the distant supervision data.

	ACC	F1
Predicate Classifier	41.57	44.40
Entity Generator	43.48	46.03

Table 4: Oracle results of the predicate classifier and entity generator. The entity generator is evaluated given correct predicates as input.

- "Sentence": "me too . what do you do for a living?"; "Information": ['i', 'other', 'poor']

Sometimes models will predict "Nothing can be inferred", and in this case we ask the turkers to label "Yes" if the sentence is generic and does not contain any personal information, otherwise is "No". We also give some examples for this case such as "Hello, how are you today?" or "sounds good, I sure I would love that!" to guide turkers.

4.3.2. Automatic Evaluation

For reference, we also report the accuracy, F1 score, and BLEU-1 score between the attributes of distant supervision data and the generated attributes. Accuracy and F1 score are computed by strict matching; i.e., the generated attributes are considered as true positive if and only if every token is exactly the same as the expected attributes. The BLEU-1 score (Papineni et al., 2002) is, meanwhile, more flexible since the object words do not need to be exactly the same (e.g. "dogs" and "two dogs", "dislike heights" and "fear of heights").

On the other hand, S-OpenIE and LLS-OpenIE are the models pre-trained on other information extraction datasets. We conduct a qualitative study with multiple different utterances as input to suggest the fundamental difference in ability between the OpenIE models and ours.

5. Results

5.1. Internal Comparison

As shown in Table 3, the proposed attribute extraction model achieves the highest F1 score, 28.68%, which surpasses the other two generation models (Seq2Seq and PG), and it is slightly better than the neural retrieval model (KVMN). Moreover, our model achieves the highest BLEU-1 score, 51.87, where all the generation models work better than KVMN. This is because KVMN has

the limitation that it can only retrieve triplets that are predefined in the dataset, and cannot generate new triplets.

The oracle study of the attribute extractor is shown in Table 4. The predicate classifier achieves a 44.4% F1 score on the multi-label classification with 61 possible predicates. In the oracle study, the entity generator, which is given the correct predicates in the distant supervision data as input, can obtain a 46.03% F1 score. Therefore, the performance drop from 46.03% to 28.68% is because of the incorrect predicate prediction.

We also conduct human evaluation over 100 randomly selected test samples. The results show that 67.11% of our generated user attributes can be inferred from the user utterances, which is significantly better than KVMN by 15.1%. We also evaluate the distant supervision data, the *Gold* row in Table 3, and the results suggest that around 20% of the data we use could be noisy input.

In general, the automatic evaluation scores are not that promising, which suggests that extracting user attributes from dialogue is challenging. However, since our test data is not human-annotated, these numbers are only for reference.

5.2. External Comparison

We show some generated samples from the test set in Table 5, and compare them with S-OpenIE (Stanovsky et al., 2018) and LLS-OpenIE (Angeli et al., 2015) to suggest the difference. One can observe that existing OpenIE approaches directly parse words from sentences, but our model learns to predict possible predicates. For example, our model successfully predicts *none* if none of the predicates is triggered, but others still return the parsing results, which contain important information. In addition, our model is able to predict relations which are not explicitly mentioned in the sentences. For example, the user utterance "I like cats. I have one" triggers the predicate *have_pet*, and "My wife can spend it" triggers the predicate *marital_status*.

We also provide some negative examples of our generated user attributes. We find three common errors: wrong predicate prediction, ambiguous attribute inference, and missing attribute prediction. First, if our model does not predict predicates correctly, it may generate out-of-context object phrases. For example, our model predicts *like_music* as a triggered predicate for the utterance "I like classic cars!" because it is biased by people mentioning classical music. Second, we find that in some cases our model generates attributes that are relevent but not certain, making the attribute ambiguous. For example, when a user says he/she is "Tired from too many parties," our model predicts the attribute *(I, like_activity, partying)* although the user does not mention it explicitly. Third, sometimes no predicate is triggered, even if there is some useful user information. For example, we should be able to conclude that a user likes to travel if he/she says "I travel a lot. I even studied abroad."

6. Discussion

Once we obtain user attributes, they can be applied to many downstream applications, for example, search, friend recommendation, online advertisement, computational social science, personalized personal assistant, etc. We select two directions we are interested in and discuss them in detail, and point out current limitations.

6.1. Potential Applications

Personalized Dialogue Agents These systems have received considerable attention since they can make chit-chat more engaging and captivating (Serban et al., 2015). There are two perspectives on personalized dialogue agents: the first is giving personalities to the agents (Zhang et al., 2018; Mazare et al., 2018), and the second, which is rarely discussed, is to adapt the agents to their end users via user attributes. Therefore, if we can endow a dialogue system with a user attribute extraction module, we can make a step towards lifelong personalized dialogue systems.

A dialogue system can view user attributes extracted from the history as explicit long-term memory. This information is able to avoid the system repeating the same or similar questions. For example, if a user mentioned "I was born in September 2009" in a previous conversation two days ago, a personalized dialogue system should avoid asking similar questions, such as "Which month is your birthday?" and "How old are you?" In addition, such attributes can be used to filter or suggest what the system should reply. For example, it would not be appropriate for a personalized system to ask "How is your university life?" if the user was born in 2009 and it is 2019. It would be better for the system to reply "Wow! Soon you will be ten years old!" after inferring the time information.

Personalized Recommender System There are three main common systems for personal recommendation: A knowledge-based system has both user and item attributes, and make recommendations based on user-item attribute similarities; A content-based system recommends items similar to those a given user has liked in the past, regardless of the preferences of other users; A collaborative filtering system, meanwhile, is based on past interactions of the whole user-base, e.g., examining k-nearest neighbor users. Most of these recommender systems require real online interactions of users with items, such as mouse clicking and browsing. Our approach provides an alternative way to collect user attributes "offline," which can then be applied to cluster users, or record items that a user has mentioned in the past. For example, if both users are from San Francisco and they all like baseball, we can recommend a Giants game to one user if the other mentions it often.

6.2. Current Limitations

We have presented the idea of extracting user attributes from daily dialogues. Although our two-stage model with distant supervision can achieve reasonable results, we believe there exist limitations that should be addressed in the future.

Most importantly, a suitable dialogue dataset with clean attribute extraction labels is needed. First of all, using the NLI model to determine the relation mapping between persona sentences and utterances is not an ideal solution. As we mentioned in the error analysis, there is an ambiguous attribute inference problem. This problem suggests that using the entailment model may not always capture the real

	S-OpenIE	LLS-OpenIE	Ours
Hello, how are you doing tonight?	*(you, doing, tonight)*	*(you, are doing, tonight)*	*none*
Yeah, I like cats. I have one.	*(I, have, one)*	*(I, have, one), (I, like, cats)*	*(I, have_pet, cat)*
Go work, so my wife can spend it	*(my wife, spend, it)*	*(my wife, can spend, it)*	*(I, marital_status, married)*
They'd not fit into my mustang convertible	*(my, mustang, convertible)*	*none*	*(I, have_vehicle, convertible)*
I'm originally from California though!	*(I, am, from California)*	*(I, am from, California)*	*(I, place_origin, California)*
Lol, I like classic cars!	*(lol, like, classic cars)*	*(I, like, cars)*	*(I, like_music, classic rock)*
Tired from too many parties.	*none*	*none*	*(I, like_activity, partying)*
I am well and you? It is cold	*(I, am, well), (it, is, cold)*	*(it, is, cold)*	*(I, like_general, cold weather)*
I traveled a lot, I even studied abroad.	*(I, travel, a lot), (I, even studied, aboard)*	*none*	*none*

Table 5: External comparison of our attribute extractor and two existing open information extraction approaches, S-OpenIE and LLS-OpenIE. Both positive and negative examples are provided.

causality information. For example, the fact that a person attends many parties does not necessarily mean they like parties. Next, the pre-defined predicates from (Sean et al., 2018) are not collected comprehensively, which may not be able to cover all the relations in a real scenario. Therefore, using clustering techniques to group more predicates automatically is an appealing solution. Lastly, the conversations in the Persona-Chat dataset are not collected naturally, with most of the users tending to ignore what the other said and just talking about themselves. Therefore, it is hard to evaluate whether "understanding your partner" helps agents speak properly. Also, since there is no publicly available data with the same user continually talking to a system, it is hard to evaluate the lifelong setting.

7. Related Work

User Attributes Inference Most previous work has treated user attribute inference from social media as a classification task, such as gender prediction (Ciot et al., 2013), age prediction (Rao et al., 2010; Alekseev and Nikolenko, 2016), occupation (Preoţiuc-Pietro et al., 2015), and political polarity (Pennacchiotti and Popescu, 2011; Johnson and Goldwasser, 2016). (Li et al., 2014) propose to extract three user attributes (spouse, education, and job) from Twitter using weak supervision. (Bastian et al., 2014) present a large-scale topic extraction pipeline, which includes constructing a folksonomy of skills and expertise on LinkedIn.

Information Extraction Closed and open form information extraction are important and well studied NLP tasks (Banko et al., 2007; Wu and Weld, 2010; Berant et al., 2011; Fader et al., 2014). Both rule-based (Mausam et al., 2012; Del Corro and Gemulla, 2013) and learning-based (Zeng et al., 2014; Xu et al., 2015; Angeli et al., 2015; Wang et al., 2016; Stanovsky et al., 2018; Vashishth et al., 2018) methods have been proposed by the research community. However, most approaches are only able to handle information by tagging/parsing part of the input source. Additionally, our work is also related to the dialogue state tracking tasks for task-oriented dialogue systems (Wu et al., 2019a).

Personalized Systems Recommender systems predict the preference a user would give to an item, which is utilized in a variety of areas. Content-based filtering (Pazzani and Billsus, 2007), knowledge-based filtering (Burke, 2000) and collaborative filtering (Sarwar et al., 1998) are the most common approaches for recommender systems. For dialogue applications, (Lucas et al., 2009) and (Joshi et al., 2017) focus on letting the agent be aware of the human pre-defined profile and so adjust the dialogue accordingly. (Zemlyanskiy and Sha, 2018) define a mutual information discovery score to re-rank system generating responses. (Madotto et al., 2019) uses meta-learning to fast adapt to unseen persona scenarios.

8. Conclusion

We utilize conversational data to extract user attributes for better user understanding. Due to lacking a labeled dataset, we apply distant supervision with a natural language inference model to train our proposed two-stage attribute extractor. Our model surpasses several retrieval and generation baselines on human evaluation, and is different from existing open information extraction approaches. In the end, we discuss potential downstream applications and point out current limitations to provide suggestions for future work.

9. Bibliographical References

Alekseev, A. and Nikolenko, S. I. (2016). Predicting the age of social network users from user-generated texts with word embeddings. In *2016 IEEE Artificial Intelligence and Natural Language Conference (AINL)*, pages 1–11. IEEE.

Angeli, G., Johnson Premkumar, M. J., and Manning, C. D. (2015). Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354. Association for Computational Linguistics.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Bastian, M., Hayes, M., Vaughan, W., Shah, S., Skomoroch, P., Kim, H., Uryasev, S., and Lloyd, C. (2014). Linkedin skills: large-scale topic extraction and inference. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 1–8. ACM.

Berant, J., Dagan, I., and Goldberger, J. (2011). Global learning of typed entailment rules. In *Proceedings of*

the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 610–619. Association for Computational Linguistics.

Bordes, A., Boureau, Y.-L., and Weston, J. (2016). Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

Burke, R. (2000). Knowledge-based recommender systems. *Encyclopedia of library and information systems*, 69(Supplement 32):175–186.

Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., and Inkpen, D. (2017). Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668. Association for Computational Linguistics.

Ciot, M., Sonderegger, M., and Ruths, D. (2013). Gender inference of twitter users in non-english contexts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1136–1145.

Del Corro, L. and Gemulla, R. (2013). Clausie: Clause-based open information extraction. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 355–366, New York, NY, USA. ACM.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dinan, E., Logacheva, V., Malykh, V., Miller, A., Shuster, K., Urbanek, J., Kiela, D., Szlam, A., Serban, I., Lowe, R., et al. (2019). The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*.

Fader, A., Zettlemoyer, L., and Etzioni, O. (2014). Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1156–1165, New York, NY, USA. ACM.

Hashimoto, K., Xiong, C., Tsuruoka, Y., and Socher, R. (2016). A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*.

Johnson, K. and Goldwasser, D. (2016). "all I know about politics is what I read in twitter": Weakly supervised models for extracting politicians' stances from twitter. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2966–2977, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Joshi, C. K., Mi, F., and Faltings, B. (2017). Personalization in goal-oriented dialog. *arXiv preprint arXiv:1706.07503*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, J., Ritter, A., and Hovy, E. (2014). Weakly supervised user profile extraction from twitter. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 165–174.

Lucas, J., Fernández, F., Salazar, J., Ferreiros, J., and San Segundo, R. (2009). Managing speaker identity and user profiles in a spoken dialogue system. *Procesamiento del lenguaje natural*.

Madotto, A., Wu, C.-S., and Fung, P. (2018). Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478.

Madotto, A., Lin, Z., Wu, C.-S., and Fung, P. (2019). Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Mausam, Schmitz, M., Soderland, S., Bart, R., and Etzioni, O. (2012). Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics.

Mazare, P.-E., Humeau, S., Raison, M., and Bordes, A. (2018). Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779. Association for Computational Linguistics.

Miller, A., Fisch, A., Dodge, J., Karimi, A.-H., Bordes, A., and Weston, J. (2016). Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer.

Pennacchiotti, M. and Popescu, A.-M. (2011). A machine learning approach to twitter user classification. In *Fifth International AAAI Conference on Weblogs and Social Media*.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Preoţiuc-Pietro, D., Lampos, V., and Aletras, N. (2015). An analysis of the user occupational class through twitter content. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language*

Processing (Volume 1: Long Papers), volume 1, pages 1754–1764.

Rao, D., Yarowsky, D., Shreevats, A., and Gupta, M. (2010). Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44. ACM.

Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., and Riedl, J. (1998). Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *in the GroupLens Research Collaborative Filtering System???. Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW*.

Sean, W., Weston, J., Szlam, A., and Cho, K. (2018). Dialogue natural language inference. *arXiv preprint arXiv:1811.00671*.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1073–1083.

Semih Yavuz, Abhinav Rastogi, G.-l. C. D. H.-T. (2018). Deepcopy: Grounded response generation with hierarchical pointer networks. *NeurIPS Conversational AI Workshop*.

Serban, I. V., Lowe, R., Henderson, P., Charlin, L., and Pineau, J. (2015). A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*.

Stanovsky, G., Michael, J., Zettlemoyer, L. S., and Dagan, I. (2018). Supervised open information extraction. In *NAACL-HLT*.

Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Z. Ghahramani, et al., editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Vashishth, S., Joshi, R., Prayaga, S. S., Bhattacharyya, C., and Talukdar, P. (2018). Reside: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In I. Guyon, et al., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Wang, L., Cao, Z., de Melo, G., and Liu, Z. (2016). Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307. Association for Computational Linguistics.

Weston, J., Dinan, E., and Miller, A. H. (2018). Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*.

Wolf, T., Sanh, V., Chaumond, J., and Delangue, C. (2019). Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.

Wu, F. and Weld, D. S. (2010). Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127. Association for Computational Linguistics.

Wu, C.-S., Madotto, A., Winata, G., and Fung, P. (2018). End-to-end dynamic query memory network for entity-value independent task-oriented dialog. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6154–6158, April.

Wu, C.-S., Madotto, A., Hosseini-Asl, E., Xiong, C., Socher, R., and Fung, P. (2019a). Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Wu, C.-S., Socher, R., and Xiong, C. (2019b). Global-to-local memory pointer networks for task-oriented dialogue. In *Proceedings of the 7th International Conference on Learning Representations*.

Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., and Jin, Z. (2015). Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794. Association for Computational Linguistics.

Zemlyanskiy, Y. and Sha, F. (2018). Aiming to know you better perhaps makes me a more engaging dialogue partner. *CoNLL 2018*, page 551.

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344. Dublin City University and Association for Computational Linguistics.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. (2018). Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213. Association for Computational Linguistics.