Zero-label Anaphora Resolution for Off-Script User Queries in Goal-Oriented Dialog Systems

M.H. Maqbool♣, Luxun Xu♠, A.B. Siddique♡, Niloofar Montazeri♠, Vagelis Hristidis♠, Hassan Foroosh♣
hasanmaqbool@knights.ucf.edu♣, lxu051@ucr.edu♠, siddique@cs.uky.edu♡,
niloofar@ucr.edu♠, vagelis@cs.ucr.edu♠, Hassan.Foroosh@ucf.edu♣

University of Central Florida♣, University of California Riverside♠, University of Kentucky♡

2022 IEEE 16th International Conference on Semantic Computing (ICSC) | 978-1-6654-3418-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICSC52841.2022.00043

***Abstract*—Most of the prior work on goal-oriented dialog systems has concentrated on developing systems that heavily rely on the relevant domain APIs to generate a response. However, in the real world, users frequently make such requests that the provided APIs cannot handle, we call them "off-script" queries. Ideally, existing information retrieval approaches could have leveraged relevant enterprise's unstructured data sources to retrieve the appropriate information to synthesize responses for such queries. But, in multi-turn dialogs, these queries oftentimes are not self-contained, rendering most of the existing information retrieval methods ineffective, and the dialog systems end up responding "sorry I don't know this". That is, off-script queries may mention entities from the previous dialog turns (often expressed through pronouns) or do not mention the referred entities at all. These two problems are known as coreference resolution and ellipsis, respectively; extensively studied research problems in the supervised settings. In this paper, we first build a dataset of off-script and contextual user queries for goal-oriented dialog systems. Then, we propose a zero-label approach to rewrite the contextual query as a self-contained one by leveraging the dialog's state. We propose two parallel coreference and ellipsis resolution pipelines to synthesize candidate queries, rank and select the candidates based on the pre-trained language model GPT-2, and refine the selected self-contained query with the pre-trained BERT. We show that our approach leads to higher quality expanded questions compared to state-of-the-art supervised methods, on our dataset and existing datasets. The key advantage of our novel zero-label approach is that it requires no labeled training data and can be applied to any domain seamlessly, in contrast to previous work that requires labeled training data for each new domain.**

***Index Terms*—Zero-label Learning, Contextual Query Rewrite, Dialog Systems, Goal-Oriented Dialog Systems**

I. Introduction

Goal-oriented dialog systems provide humans with an intuitive natural language interface to interact with machines for carrying out tasks (e.g., Amazon Alexa), such as booking event tickets. The majority of prior research on goal-oriented dialog systems has concentrated on developing systems that employ the relevant domain APIs to query data sources (e.g., databases) for retrieving required information to synthesize a response for the user. However, users frequently submit requests that the provided APIs are not supposed to handle, we refer these requests as "off-script" queries. That is, instead of proceeding with the conversation as expected by the dialog system (i.e., supported by APIs), which is typically a series of questions or suggestions with a specific end-goal, the user digresses and asks a question that cannot be answered by the chatbot engine, mainly due to the unavailability of the relevant APIs. Figure 1 presents such a scenario, where the user is interested in knowing about the availability of free parking at "Petco Park" for which the systems designers have not provided any API. The user issues an off-script query, "Is free parking available?", where the location (i.e., "Petco Park") is implicit. This is referred to as zero anaphora or ellipsis. Similarly, the user might have asked, "Does it have free parking?", where the pronoun "it" is referring to "Petco Park". This is referred to as coreference. Please note that pronouns are only one type of anaphoric expressions. The user could also refer to "Petco Park" with a nominal reference (e.g.,"the venue") , or with a locative form (e.g., "there").

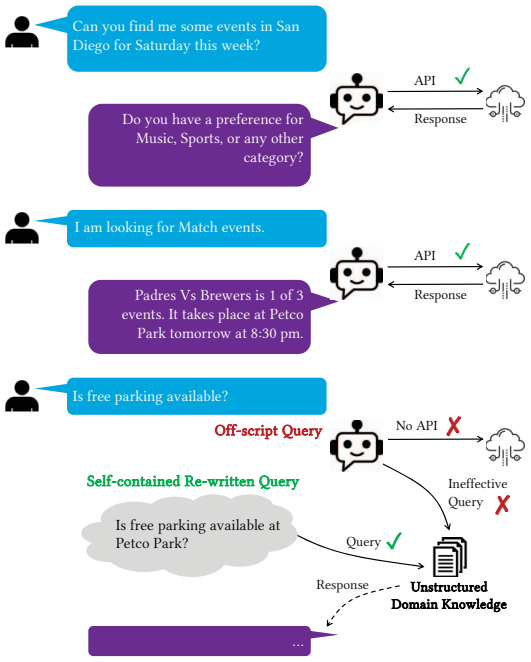

Fig. 1: In multi-turn dialog systems, oftentimes "off-script" queries are not *self-contained*, thus existing IR approaches are rendered ineffective. rewriting the query as self-contained can empower IR methods to retrieve the required information.

978-1-6654-3418-8/22/$31.00 ©2022 IEEE
DOI 10.1109/ICSC52841.2022.00043

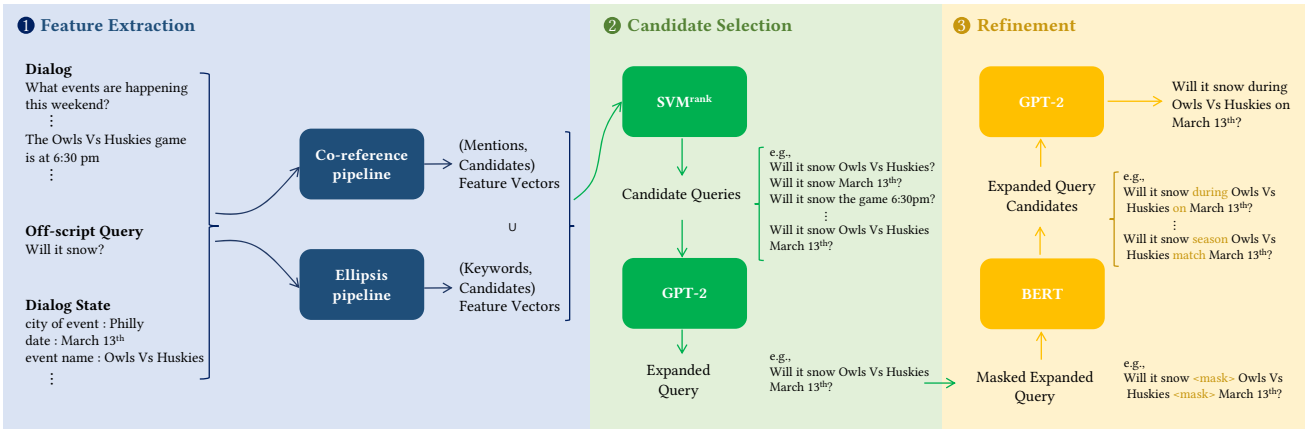

Fig. 2: Overview of the proposed zero-label approach for anaphora resolution in dialog systems.

For a query of this nature, existing information retrieval (IR) methodologies could have used unstructured data sources (i.e., usually available) to retrieve relevant information to generate a response. However, existing off-the-shelf IR methods are not effective for such scenarios, as in multi-turn dialogs, these queries often mention entities from the previous dialog turns usually via pronouns (i.e., coreference) or do not mention the entities at all (i.e., ellipsis). In fact, one study showed that about 70% of utterances in multi-turn dialogues contain either coreference or ellipsis [1]. Proper resolution of the anaphoric references leads to self-contained search queries. For example, a self-contained version of the above queries can be "Is free parking available at Petco Park?" or "Does Petco Park have free parking?". Interacting with the dialog system with off-script queries might result in the breaking of the dialog session. Consequently, the user is highly likely to drop out before completing the goal; resulting in a potential business loss.

The task of using the context to rewrite an incomplete user utterance in order to make it self-contained is referred to as Incomplete Utterance Rewriting (IUR) [2] or Context Rewriting. The other variants of the same task are known as Question Decontextualization [3], Conversational Question Reformulation (CQR) [4], Context-aware Query Reformulation [5], Contextual Query Rewriting [6]. All of these tasks have been studied extensively in the supervised setting and many deep learning based systems have been proposed for anaphora resolution in dialogue systems [1], [7], [8], [9], [10], [3], [11], [5], [6], [12]. However, supervised methods require huge amounts of *labeled* training data that is laborious and expensive to acquire, rendering such approaches *unscalable* [13], [14], [15], [16]. In order to develop a dialog system for a new domain (e.g., restaurants, events), the requirement of having labeled data for each domain is not feasible, and that motivates our zero-label approach for anaphora resolution for off-script queries in the context of goal-oriented dialog systems. Our novel zero-label[1]. approach does not require any labeling of the training data for the given domains.

[1]The source code is available at https://github.com/UC-Riverside-DatabaseLab/DialogQuestionExpansion

Figure 2 shows our proposed pipeline for rewriting the off-script query as self-contained one. That is, adding appropriate context and relevant slot values to resolve anaphora in user's query. The input to the system is the dialog history, off-script query, and current dialog state (i.e., set of key-value pairs for critical entities of the active domain). The output is a self-contained version of the off-script query, where anaphoric mentions (or null mentions, in case of ellipsis) are replaced by appropriate slot values, while keeping the query fluent. The proposed approach works in three logical phases, called, feature extraction, candidate selection, and refinement.

The feature extraction phase employs two parallel pipelines; one for handling coreference and the other for the ellipsis case. Both pipelines consume the input and produce a set of mentions (that can be null in case of ellipsis), corresponding candidate slot values (i.e., candidate reference), and their feature vectors. The union of the outputs of the coreference and ellipsis pipelines is then passed to the candidate selection phase. It ranks the candidates by the weighted sum of the features leveraging SVM^{rank}. The candidate queries are generated by replacing mentions with corresponding slot values (i.e., coreference resolution) or simply adding the candidate slot values to the query (i.e., ellipsis resolution). Then, these candidate queries are scored by a pre-trained language model GPT-2, and the best candidate query is selected. Finally, the refinement phase employs a masked language model BERT and GPT-2 to further refine the selected candidate for fluency.

In addition, we also present a novel dataset that focuses on off-script query rewriting for goal-oriented dialog systems. In Summary, the contributions of this work are as follows:

- We propose a novel zero-label approach that leverages current dialog state to effectively de-contextualize user's query.
- We show that the proposed method achieves better or comparable results to supervised approaches, without a need for labeled training data, making it suitable for chatbots deployment in new domains seamlessly.
- We created a dataset for off-script query rewriting for goal-oriented dialog systems.

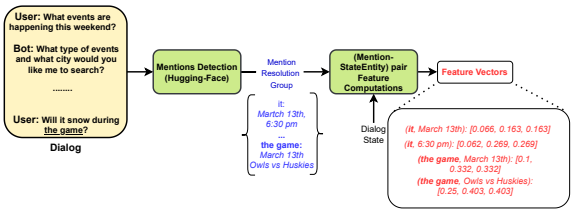

Fig. 3: Coreference resolution (HF-CR) pipeline.

II. Zero-label Anaphora resolution Method

The input to the system is user and system utterances from the beginning of the dialogue up to the (excluding) off-script user query, represented by H, an off-script query denoted by Q, and the current dialog state, DS, in the form of key-value pairs. The goal is to synthesize a self-contained variant Q' of the original query. To achieve this, we propose a novel zero-label approach, that works in three logical phases, called, feature extraction, candidate selection, and refinement. In the following, we explain each phase in detail.

A. Feature Extraction

This phase takes H, Q, DS (i.e., dialog history, query, dialog state, respectively) as inputs and produces a set of mentions M, candidate slot values CSV, and corresponding feature vectors F. This phase employs two parallel pipelines, called, coreference resolution and ellipsis resolution.

1) Coreference Resolution: Our coreference resolution pipeline is shown in Figure 3. We first use HuggingFace's neural coreference resolution system, which we call HF-CR, to identify mentions in the off-script question and their corresponding candidate references. Next, we filter out mentions that contain other mentions. Finally, for each mention and its corresponding references, we check if the reference appears in any of the the dialogue states and if so, we assign a feature vector for the corresponding (mention, reference) pair.

Identifying mentions and candidate references. The input to HF-CR is the full dialogue H, appended by the off-script query Q, and the output can be viewed as a set of (M,$\{\overline{R}\}$) pairs where M is a mention in the off-script query and $\{\overline{R}\}$ is the set of candidate references (in the preceding dialogue) for M. A mention is a reference or representation of an entity or an object that appears in text. We use HF-CR, to identify mentions in the off-script query and their corresponding candidate references. Table I shows sample mentions and their candidate references for the off-script query, corresponding a sample dialog (complete dialog is presented in Table V). We then remove mentions that contain other mentions, M'. In the same example, *the capacity of the stadium* will be dropped and *the stadium* will be processed in the next step. It is important to highlight that we remove candidate references that do not match any slot values. All the remaining references that match a slot value are processed further and hence are called *candidate slot values*, CSV.

Creating Feature Vectors. We assign feature vectors to each mention and candidate slot value pair (M, CSV) by considering, (*i*) $sim(M, CSV)$: semantic similarity between M and CSV, if M is a noun phrase, (*ii*) $sim(K, CSV)$: average semantic similarity between K keywords (see Section II-A2 for details on how we obtain keywords, K) in the off-script query Q and candidate slot values CSV, if M is a pronoun. In cases like *"the stadium"*, the mention itself is representative enough. However, for mentions like *"it"* and other pronouns, we need to find other parts of the sentence to replace mention for feature vector generation. Therefore, in the case of mentions being a pronoun, we try to find keywords that best represent the sentence to compare it against the dialog states. We use the following two metrics to compute the similarity between two words or phrases, (*i*) WordNet [17]: hypernym-relationship distance (the shorter the distance, the closer the relationship). For example, {location, stadium} has a distance of 5, whereas {location, time} has a distance of 9. (*ii*) Cosine similarity between GloVe's [18] pre-trained word vectors (6B tokens, 200d). Moreover, we also noticed that, in cases, where the CSV is a proper noun (e.g. "Petco Park") or an abbreviation (e.g., "SD" for "San Diego"), it helps to use its type T and compute $sim(M, T(CSV))$ and $sim(K, T(CSV))$ in addition to the above values. We use external knowledge bases to find $T(CSV)$. A similar approach is employed for metnions M.

Question	Can you tell me the capacity of the stadium?
Mentions	the capacity of the stadium the stadium
Mention cluster	the capacity of the stadium: {Petco Park, 8:30 pm, ...} the stadium: {Petco Park, 8:30 pm, ...}
Mention resolution group	the stadium: {(event_location, Petco Park), (time, 8:30 pm), ...}

TABLE I: Identifying mentions from off-script query.

Slot name	**Slot value**
date	next Monday
event_location	Petco Park
count	4
event_name	Padres vs Diamondbacks
city_of_event	SD
category	sports
subcategory	baseball

TABLE II: Sample Dialogue State.

Table II lists some examples of slot names and values. In cases like "{Event Location: Petco Park}", the slot name represents the type of the slot value. However, this might not always be the case. The slot names may be abbreviations (e.g., "POI" for "Point of Interest"), or too general (e.g., "event_name" instead of "artist" or "team"). Therefore, we also use Google Knowledge Base (GKB) for identifying the type of slot values.

For each (M, CSV) pair, we extract a 3-parameter feature vector $F = (f_1, f_2, f_3)$ (in case of (K, CSV) pairs, we take the average of calculated parameter scores for all Ks). These features represent, (*i*) f_1: reciprocal of WordNet hypernym distance between M and CSV, (*ii*) f_2: Cosine similarity of GloVe vectors between M and CSV, and (*iii*) f_3: cosine similarity of GloVe vectors between M and $T(CSV)$ extracted

Algorithm 1 Coreference Resolution Pipeline

run *Full dialog*, Q concatenated to H through *HuggingFace Neural Coref Tool*
output $MC = \{M_a : (R_{a1}, R_{a2}, ...), M_b : (R_{b1}, R_{b2}, ...), ...\}$
if M appears in query Q **then**
  for every M **do**
    Remove M **if** $M \supset M'$
  end for
  for each remaining R_{ni} **do**
    if $R_{ni} \in DS$ **then**
      output $[f_1, f_2, f_3]$ vector for each $[M_n, R_{ni}]$ pair
    end if
  end for
end if

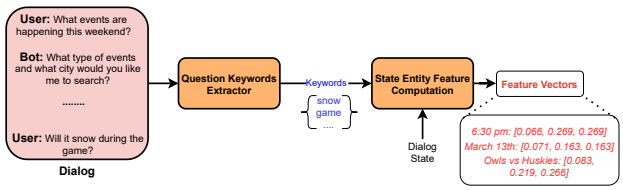

Fig. 4: Ellipsis resolution pipeline.

from the Google KB category. Algorithm 1 also illustrates the coreference resolution pipeline.

2) Ellipsis Resolution: Figure 4 presents the ellipsis resolution pipeline. Since there is no notion of mention in ellipsis case, we first need to find keywords that represent the sentence for adding missing information. We then generate a feature vector for each candidate slot value by considering its average similarity to all the keywords.

Finding keywords. First, we use Stanford CoreNLP library and perform constituency parsing to obtain noun phrases *NP* and verb phrases *VP* from the sentence. Among all the *NP* and *VP*, we select the least frequent ones to be the keywords of the sentence. To avoid introducing extra noise, we only choose at most 3 phrases as keywords K.

Creating Feature Vectors. Once the keywords are found, we compare the keywords with dialog states, similar to section II-A1 and output feature vectors. Algorithm 2 also illustrates the ellipsis resolution pipeline.

B. Candidate Selection

This phase takes in mentions M, candidate slot values S, and corresponding feature vectors F, and selects the best query.

Candidate Queries Generation. First, we generate candidate queries by employing the relevant feature values. Specifically, we assign a score to each resolution case by taking the weighted sum of the feature values in its feature vector, as shown in Table III. We use the SVM^{rank}[2] to learn the weights based

[2]http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

Algorithm 2 Ellipsis Resolution Pipeline

find keywords in *final question*
for each token $t \in Q$ **do**
  output K_1, K_2, K_3 where
  1. K_i is a *NP* or *VP*
  2. $Freq(K_i) < Freq(k_j)$ where $i < j$
end for
for each DS_j **do**
  compute f_{1i}, f_{2i}, f_{3i} for each $[K_i, DS_j]pair$
  output $[f_1, f_2, f_3]$ for each DS_j by taking the average for each f (Eq.: 1)
end for

Off-script question	At what time does it start?				
Coreference cases	f_1	f_2	f_3	Score	Rel.
it -> Nycfc Vs Timbers	0.2500	0.4412	0.5387	2.435	1
it -> New York	0.1000	0.5258	0.5258	1.483	0
Ellipsis cases	f_1	f_2	f_3	Score	Rel.
next Monday	0.1111	0.5258	0.5258	1.941	0
New York	0.0714	0.3878	0.3878	1.384	0

TABLE III: Scores for each case and example feature vectors.

on a small out-of-domain training set (i.e., 10 questions in our experiment). For each data point in the set, we manually label each resolution case that matches the ground truth as relevant and the rest as irrelevant as shown in Table III. Then we use SVM^{rank} on this set to train the weights w_1, w_2, w_3 for the corresponding features f_1, f_2, f_3. Then, candidate queries are generated by replacing mentions with corresponding candidate slot value CSV (i.e., coreference resolution case), or simply adding the slot value CSV to query (i.e., ellipsis resolution case).

Scoring the Candidate Queries. GPT-2 has been employed to generate natural language text and has shown state-of-the-art results on many NLP benchmarks. In this work, we use GPT-2 to score the generated queries by computing its perplexity. The generated candidate queries are scored by a pre-trained language model GPT-2 and those scoring below a threshold ϵ are combined to yield a self-contained query that is further refined in the final phase. Specifically, we use GPT-2 perplexity to select the most promising candidate. First, we re-use the tiny validation set and compute the "benchmark" average GPT-2 perplexity as well as the standard deviation. Then we proceed to the following steps. First, if there are coreference cases, we resolve those by picking the highest-scored (i.e., low perplexity) candidate for each coreference case. Next, we gradually add additional information to resolve potential ellipsis cases using the ranked list of candidate ellipsis resolution cases (i.e., ranked from lowest to highest perplexity), we append the first candidate to the current resolved question and compute its GPT-2 perplexity. If the GPT-2 perplexity is within 2 standard deviation of the "benchmark" average GPT-2 perplexity, we keep this new version of the question and move on to the next candidate ellipsis resolution case. This loop continues and candidates are appended until the GPT-2 perplexity exceeds 2

Question *What time does the game start?*		
Candidate Resolved Question	**GPT-2**	**Output**
1. What time does *the event* start?	1.707	No
2. What time does *the event* start *date*?	1.819	Yes
3. What time does *the event* start *date city*?	1.901	No
Output *What time does Timbers Vs DC United start March 3rd?*		

TABLE IV: GPT-2 score example for final output

standard deviation, at that point we stop and output the best candidate query. Table IV shows an example of how GPT-2 perplexity is used for selecting the candidate query. Candidate 1 resolves the coreference case in the query, and candidates 2 and 3 continue adding missing information. However, since candidate 3's GPT-2 perplexity exceed the threshold, we select candidate 2.

C. Refinement

In this phase, we further refine the expanded query using pre-trained BERT and GPT-2 to transform it into a well-formed and fluent one. This phase should not be confused with the traditional supervised task-specific fine-tuning using BERT or GPT-2. We employ pre-trained BERT to infer token for the masked ones, i.e., one of the original objective of the BERT, that does not require any task-specific labeled training data. Specifically, we insert $[mask]$ token before and after the expansions, that the previous phase made. The pre-trained BERT makes predictions for the masked tokens. Using the above process, we generate multiple hypothesis queries (i.e., 5 queires in our experiments). Finally, we employ GPT-2 to score the hypothesis and select the best query Q'. Figure 2 also provides an overview of this step. In our implementation, we use HuggingFace's pre-trained *bert-large-cased* and *GPT-2*.

III. Off-Script Dataset Collection

Although several datasets have been proposed for the task of contextual query re-writing, none of these tackle off-script queries. In this vein, this is the first effort, to the best of our knowledge. A sample dialog from our proposed dataset is presented in Figure 1. Our goal is to create a challenging yet realistic dataset consisting of multi-turn dialogues and off-script queries pertaining to the dialogue. Our strategy involves augmenting existing datasets, adding off-script queries to one of the most recent and comprehensive dialogue datasets. Particularly we chose Schema-Guided Dialogue (SGD) Dataset [19] because of the realism of the conversation as well as its well-maintained dialogue states for both the user and the system. Out of 20 domains in the SGD dataset, we chose the *Events* domain to build our dataset because it is the most diverse (spanning from music to sports events and more), thus has high potential to trigger diverse off-script queries from crowd-sourcing workers.

We used Amazon Mechanical Turk (MTurk) to get off-script queries. The setup is as follows. For every dialogue in the *Events* domain, depending on the number of turns, we insert up to 3 blanks (evenly spaced out) right after a system utterance. Table V shows an example of the setup where the MTurk worker's job is as simple as typing a question. The instructions, we provided to the workers are as follows: *You are given a dialog between a user and a virtual assistant, which involves a particular task such as making an appointment or booking a ticket. We want to evaluate how the virtual assistant will perform when the user goes off-script, and asks a question related to the dialogue so far, which the virtual assistant may not know how to answer, but the answer may be on a Web page.*

Speaker	**Utterance**
User:	I feel like watching some baseball. Can you find a Match around me?
System:	In which city would that be?
User:	Around SD please.
System:	I found 4 matches. There's Padres Vs Brewers at Petco Park tomorrow at 8:30 pm.
User:	That's nice but is anything else happening?
System:	There's Padres vs Diamondbacks at Petco Park next Monday at 6 pm.
Off-script Query:	*Can you tell me the capacity of the stadium?*
	Labeling of Question
Case:	Coreference
Truth:	Can you tell me the capacity of Petco Park?

TABLE V: off-script query collection template for Mechanical Turkers. The first highlighted placeholder is filled by one Mechanical Turk worker and the other two by another.

After getting the questions from the Mechanical Turkers, we reject the obviously bad questions and label the rest of the questions coreference, ellipsis or both. Then we manually resolve the questions and add them as the ground truth. This is shown in the bottom section in Table V as well. Our final dataset consist of 386 dialogs containing off-script queries. Out of these, 252 are coreference resolution cases, 124 are ellipsis resolution cases, and 10 are both coreference and ellipsis (i.e., zero anaphora) case.

IV. Evaluation

To evaluate the efficacy of our proposed zero-label approach for anaphora resolution approach, we compare it to the state-of-the-art supervised methods: (*i*) Gecor [12] and (*ii*) RUN [2]. Both are the most relevant state-of-the-art systems for coreference and ellipsis resolution in dialogues. As evaluation metrics, we use **sentence-level BLEU**, **precision**, **recall** and **F1** scores when comparing the system-generated vs. ground truth rewrites.

A. Datasets

Our proposed anaphora resolution method relies on availability of dialogue states (i.e., usually available in real-world for the goal-oriented dialog systems). Therefore, in order to compare our method to other anaphora resolution systems, we are restricted to goal-oriented dialogue datasets. In particular, We run evaluations on the following datasets which provide ground truth rewrites for coreference and ellipsis resolution in their goal-oriented dialogue datasets:

- Our proposed off-script questions dataset that covers *Events* (see Section III).

	Coreference	Ellipsis	Total
Ours	**252**	**124**	**386**
Gecor	161	165	326
CQR	-	-	163

TABLE VI: Dataset statistics.

Speaker	Utterance
User:	I am looking for a restaurant that is moderately priced and serves Cantonese food.
System:	There are no restaurants that serve Cantonese food in the moderate price range.
User:	How about Chinese type of food?
System:	The Golden wok serves Chinese food and is in the moderate price range. Would you like their location?
User:	Their phone number please.
Complete:	I would like the phone number of **the Golden wok** please.

TABLE VII: Sample dialog from TASK dataset.

Speaker	Utterance
User:	What gas stations are here?
System:	There is a Chevron.
User:	That's good! Please pick the quickest route to get there and avoid all heavy traffic!
System:	Taking you to Chevron.
User:	What is the address?
Complete:	What is the address of **the gas station Chevron**?

TABLE VIII: Sample dialog from CQR dataset.

- TASK [12]: A dataset based on CamRest676 covering *Restaurants* domain. This dataset only provides user dialogue states and no system dialogue states. Table VII presents a sample dialog from this dataset.
- CQR (Contextual Query Rewrite) [20]: A dataset based on the Stanford dialog corpus [21] covering *Calendar Scheduling*, *Weather*, and *Navigation* domains. This corpus includes crowd-sourced rewrites in addition to ground truth ones and uses dialogue slot values as context for the rewriting task. A sample from this dataset is shown in Table VIII.

The statistics of the datasets are summarized in Table VI.

Dataset Pre-processing. Our dataset, which is built on top of *dstc8-schema-guided-dialogue*, follows its format where the dialogue states for both the user and the system are provided. However, we noticed that both TASK and CQR datasets do not maintain dialogue states for system utterances. Therefore, we automatically added system dialogue states: For TASK dataset, we extracted the dialogue states from the system utterances by matching sub-strings with the values from the given database *CamRestDB.json*. For CQR dataset, we performed the same step as for TASK dataset since the equivalent "database" is given in the same json file under *"scenario"* -> *"kb"*. Moreover, we find that in TASK test set, there are test cases such as *thank you, goodbye!* which do not require any resolution, so we exclude those cases.

B. Experimental setup

We performed evaluations on all data from our dataset, on 20% test data on filtered TASK dataset, and on the standard test set of the CQR dataset. While our proposed approach is zero-label, Gecor and RUN are supervised and need training data: For Gecor, we use 80% of TASK and for RUN, we use the "rewrite_bert" pre-trained model, as described in [2] which they report as achieving the highest performance. Moreover, we use sentence level calculations of the metrics, meaning we consider each utterance that needs to be resolved and compute its score, such as BLEU-4 or F1 against the ground truth and take the average over the number of test cases.

		BLEU-4	F1	Recall	Precision
Events domain (our dataset)					
Coreference	Gecor	0.410	0.698	0.671	0.728
	Ours	**0.654**	**0.831**	**0.810**	**0.850**
Ellipsis	Gecor	0.438	0.633	0.649	0.742
	Ours	**0.592**	**0.828**	**0.783**	**0.877**
Overall	Gecor	0.407	0.641	0.606	0.681
	RUN	0.494	0.716	0.647	0.801
	Ours	**0.562**	**0.809**	**0.791**	**0.829**
Restaurants domain (Gecor dataset)					
Coreference	Gecor	0.635	0.691	0.669	0.714
	Ours	**0.661**	**0.816**	**0.778**	**0.859**
Ellipsis	Gecor	**0.518**	0.707	0.59	0.88
	Ours	0.503	**0.777**	**0.684**	**0.899**
Overall	Gecor	**0.576**	0.699	0.630	0.795
	RUN	0.452	0.694	0.640	0.758
	Ours	0.526	**0.770**	**0.697**	**0.859**
Mixed domains (CQR dataset)					
Overall	Gecor	0.161	0.364	0.308	0.444
	RUN	**0.299**	0.613	0.502	0.785
	Ours	0.298	**0.651**	**0.516**	**0.879**

TABLE IX: Results: our proposed zero-label approach consistently outperforms other SOTA supervised methods on F-1 and is very competitive on BLEU-4.

C. Results

The results of the experiments are presented in Table IX. We can see that our proposed zero-label approach outperforms both Gecor and RUN in all three datasets for precision, recall, and F1 scores. Specifically, it is up to 19% more accurate on F1 scores for coreference resolution, 31% for ellipsis resolution, and 26% overall on events dataset (i.e., our dataset). Similar observation can be made for the other datasets, where our zero-label approach is up to 18%, 10%, and 11% more accurate on F1 score for coreference resolution, ellipsis resolution, and overall, respectively, on restaurants dataset. Since CQR dataset does not present coreference and ellipsis cases seprately, we present overall results on this dataset, where our proposed approach outperform Gecor by large margin of 28+ for F1 score. It is important to highlight that our zero-label approach does not need any labeling of the training data and seamlessly work on the new unseen domains. In terms of performance on BLEU-4 metrics, our proposed approach either outperforms other SOTA supervised methods or provides very competitive performance, in spite of having no access to ground truth labels. Specifically, our method achieves better BLEU-4 score on events datasets,

whereas it shows almost similar performance for CQR dataset. Our method is slightly outperformed by Gecor on TASK dataset. Given that our method works with zero-labels, we argue that the gap is negligible because forming fluent sentences with 4-gram matching with the ground truth is very challenging, especially for the cases when there are rewrite operation in the self-contained query other than resolving references or adding missing information. Table VII presents such an example for TASK dataset. For the user query, "Their phone number please.", the ground truth rewritten version of the query is, "I would like the phone number of the Golden wok please". For the same query, our method rewrites it as, "Golden wok's phone number please.", which is still acceptable, but not as close to the ground truth label.

V. Related Work

A. Systems

Coreference resolution is a well-studied and active task in computational linguistics. The most dominant paradigm in coreference resolution employs scoring span or mention pairs [22], [23], [24], [25], [26], [27], [28]. Using unsupervised contextualized representations, particularly BERT [29] which can model long-range dependencies more effectively, have further improved performance on this task. In particular, fine-tune BERT for coreference resolution [30]. Among the best performing systems are [31] and [32] that use SpanBERT [31] which can better represent and predict spans of text. Without a large amount of annotated dialogue data (i.e., syntactic norms and candidate antecedents), the above state-of-the-art (SOTA) coreference resolution methods do not perform well in the multi-turn dialogue settings. Moreover, coreference resolution in dialogues is being addressed by deep learning methods used in machine translation and summarization such as seq2seq methods, PGNs, and transformers, that rewrite incomplete user utterances with context to make them self-contained. IUR has been used to improve the performance of a variety of dialogue-based tasks. In *open-domain conversations* [1], [7], [8], [9], it can help to improve response-generation [33] or dialog act prediction. In *conversational question answering* [10], [34], IUR is used to restore non-sentential user utterances [3], [11] rewrites anaphoric (including zero-anaphora) follow-up questions into stand-alone questions by augmenting them with appropriate context. In *conversational search*, [5] reformulates incomplete search queries to add the necessary context (from previous queries and answers/results). In *task-oriented dialogues*[35], parallel works have been conducted on resolving anaphora (including zero anaphora) in user utterances [6], [12] which can benefit downstream dialogue tasks such as understanding user's intention or dialogue state tracking leading to a higher task completion rate. Most recently, [2] formulated IUR as a semantic segmentation task (as opposed to a translation task) and achieved SOTA performance in resolving coreference and ellipsis across a variety of domains and datasets. All the above rewriting methods are supervised and need training data that might not be available when deploying a chatbot on a new domain. In this paper we propose a zero-label method for anaphora resolution in task-oriented dialogues that uses external knowledge bases and word similarity data to rewrite the incomplete utterance (off-script query) as a self-contained question. We compare the performance of our method against SOTA supervised deep learning-based systems.

B. Datasets

With the rise of interest in conversational AI, many datasets are proposed in the form of parallel corpora with incomplete user utterances and their corresponding resolved utterances. In *conversational QA*, [36] created a crowd-sourced dataset of about 7K labeled conversations where each labeled conversation has four parts: a previous complete question, a previous answer, an incomplete follow-up question and the corresponding complete question. [3] created a dataset of about 40K questions (named CANARD) by crowdsourcing context-independent paraphrases of QUAC questions [37]. In *task-oriented dialogues*, [20] have created a multi-domain dataset (named CQR) of rewrites based on Stanford dialog corpus [21] consisting of ground truth and crowd-sourced rewrites that augment anaphoric/incomplete user utterances with corresponding slot values from the dialogue history. [12] created a dataset based on CamRest676 (Restaurant domain) for ellipsis and coreference resolution. [38] added coreference annotations to the MultiWOZ dataset which consists of 10K dialogues across eight different domains. In *open domain dialogues*, [7] annotated a large-scale multi-turn Chinese dataset (named Restoration) consisting of 200K multi-turn conversations from internet communities. Each utterance is assigned 1) a labeled specifying whether or not it is context-dependent, and 2) the resolved and context-free form. [1] created a high-quality corpus of 40K tuples (dialog history+incomplete utterance, complete utterance) where the original conversations were crawled from popular Chinese social media platforms. Focusing on ellipsis resolution only, [9] present an open-domain human-machine conversation dataset consisting of about 200 social conversations with Alexa. For utterances with ellipsis, a completed version is generated manually. Our dataset differs from the above datasets in that we include a query that is unanswerable by the chatbot engine without tapping into an external knowledge base, i.e., no API is provided for such queries.

VI. Conclusion

We have presented a zero-label approach for anaphora resolution of the off-script user queries in goal-oriented dialog systems. Our proposed approach consistently outperforms existing state-of-the-art supervised methods on F1 score for a wide range of datasets. Moreover, our novel method also generates plausiable and fluent self-contained queries that achieves comparable performance to other supervised approaches for BLEU-4 metric. The key advantage of our proposed approach is that it works with zero-labels and can be employed for any domain seamlessly, in contrast to supervised approaches that require huge amounts of labeled training data for each

new domain. Our proposed approach uses current dialog state of the goal-orinted conversations and leverages the inference capabilities of the pre-trained language models to synthesize a self-contained version of the off-script query. Additionally, we propose a dataset of off-script queries that contain 386 dialogs.

REFERENCES

[1] H. Su, X. Shen, R. Zhang, F. Sun, P. Hu, C. Niu, and J. Zhou, "Improving multi-turn dialogue modelling with utterance rewriter," *arXiv preprint arXiv:1906.07004*, 2019.

[2] Q. Liu, B. Chen, J.-G. Lou, B. Zhou, and D. Zhang, "Incomplete utterance rewriting as semantic segmentation," *arXiv preprint arXiv:2009.13166*, 2020.

[3] A. Elgohary, D. Peskov, and J. Boyd-Graber, "Can you unpack that? learning to rewrite questions-in-context," *Can You Unpack That? Learning to Rewrite Questions-in-Context*, 2019.

[4] S.-C. Lin, J.-H. Yang, R. Nogueira, M.-F. Tsai, C.-J. Wang, and J. Lin, "Conversational question reformulation via sequence-to-sequence architectures and pretrained language models," *arXiv preprint arXiv:2004.01909*, 2020.

[5] G. Ren, X. Ni, M. Malik, and Q. Ke, "Conversational query understanding using sequence to sequence modeling," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1715–1724.

[6] P. Rastogi, A. Gupta, T. Chen, and L. Mathias, "Scaling multi-domain dialogue state tracking via query reformulation," *arXiv preprint arXiv:1903.05164*, 2019.

[7] Z. Pan, K. Bai, Y. Wang, L. Zhou, and X. Liu, "Improving open-domain dialogue systems via multi-turn incomplete utterance restoration," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 1824–1833.

[8] K. Zhou, K. Zhang, Y. Wu, S. Liu, and J. Yu, "Unsupervised context rewriting for open domain conversation," *arXiv preprint arXiv:1910.08282*, 2019.

[9] X. Zhang, C. Li, D. Yu, S. Davidson, and Z. Yu, "Filling conversation ellipsis for better social dialog understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 9587–9595.

[10] V. Kumar and S. Joshi, "Non-sentential question resolution using sequence to sequence learning," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2022–2031.

[11] S. Vakulenko, S. Longpre, Z. Tu, and R. Anantha, "Question rewriting for conversational question answering," *arXiv preprint arXiv:2004.14652*, 2020.

[12] J. Quan, D. Xiong, B. Webber, and C. Hu, "Gecor: An end-to-end generative ellipsis and co-reference resolution model for task-oriented dialogue," *arXiv preprint arXiv:1909.12086*, 2019.

[13] A. Siddique, F. Jamour, L. Xu, and V. Hristidis, "Generalized zero-shot intent detection via commonsense knowledge," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1925–1929. [Online]. Available: https://doi.org/10.1145/3404835.3462985

[14] A. Siddique, S. Oymak, and V. Hristidis, "Unsupervised paraphrasing via deep reinforcement learning," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1800–1809.

[15] A. Siddique, F. Jamour, and V. Hristidis, "Linguistically-enriched and context-awarezero-shot slot filling," in *Proceedings of the Web Conference 2021*, ser. WWW '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 3279–3290. [Online]. Available: https://doi.org/10.1145/3442381.3449870

[16] M. A. B. Siddique, "Unsupervised and zero-shot learning for open-domain natural language processing," Ph.D. dissertation, University of California, Riverside, 2021.

[17] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[19] A. Rastogi, X. Zang, S. Sunkara, R. Gupta, and P. Khaitan, "Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset," *arXiv preprint arXiv:1909.05855*, 2019.

[20] M. Regan, P. Rastogi, A. Gupta, and L. Mathias, "A dataset for resolving referring expressions in spoken dialogue via contextual query rewrites (cqr)," *arXiv preprint arXiv:1903.11783*, 2019.

[21] M. Eric and C. D. Manning, "Key-value retrieval networks for task-oriented dialogue," *arXiv preprint arXiv:1705.05414*, 2017.

[22] V. Ng and C. Cardie, "Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution," in *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[23] E. Bengtson and D. Roth, "Understanding the value of features for coreference resolution," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 294–303.

[24] P. Denis and J. Baldridge, "Specialized models and ranking for coreference resolution," in *Proceedings of the 2008 conference on empirical methods in natural language processing*, 2008, pp. 660–669.

[25] E. Fernandes, C. dos Santos, and R. L. Milidiú, "Latent structure perceptron with feature induction for unrestricted coreference resolution," in *Joint Conference on EMNLP and CoNLL-Shared Task*, 2012, pp. 41–48.

[26] S. J. Wiseman, A. M. Rush, S. M. Shieber, and J. Weston, "Learning anaphoricity and antecedent ranking features for coreference resolution," 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)*. Association for Computational Linguistics, 2015.

[27] K. Clark and C. D. Manning, "Entity-centric coreference resolution with model stacking," 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)*, 2015, pp. 1405–1415.

[28] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," *arXiv preprint arXiv:1707.07045*, 2017.

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

[30] M. Joshi, O. Levy, D. S. Weld, and L. Zettlemoyer, "Bert for coreference resolution: Baselines and analysis," *arXiv preprint arXiv:1908.09091*, 2019.

[31] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.

[32] W. Wu, F. Wang, A. Yuan, F. Wu, and J. Li, "Corefqa: Coreference resolution as query-based span prediction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6953–6963.

[33] U. Farooq, A. B. Siddique, F. Jamour, Z. Zhao, and V. Hristidis, "App-aware response synthesis for user reviews," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 699–708.

[34] V. Kumar and S. Joshi, "Incomplete follow-up question resolution using retrieval based sequence to sequence learning," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 705–714. [Online]. Available: https://doi.org/10.1145/3077136.3080801

[35] N. Le, A. B. Siddique, F. Jamour, S. Oymak, and V. Hristidis, "Predictable and adaptive goal-oriented dialog policy generation," in *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, 2021, pp. 40–47.

[36] D. Raghu, S. R. Indurthi, J. Ajmera, and S. Joshi, "A statistical approach for non-sentential utterance resolution for interactive qa system," in *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2015, pp. 335–343.

[37] E. Choi, H. He, M. Iyyer, M. Yatskar, W.-t. Yih, Y. Choi, P. Liang, and L. Zettlemoyer, "Quac: Question answering in context," *arXiv preprint arXiv:1808.07036*, 2018.

[38] T. Han, X. Liu, R. Takanobu, Y. Lian, C. Huang, W. Peng, and M. Huang, "Multiwoz 2.3: A multi-domain task-oriented dataset enhanced with annotation corrections and co-reference annotation," *arXiv preprint arXiv:2010.05594*, 2020.