Knowledge Base Evolution via Counterfactual Conversations

Antian Wang

November 3, 2020

1 Project Description

This project targets workflow problems in AI conversational systems, in which the system tries to collect information from the user on a fixed set of questions and will output a suggestion or an executable accordingly. Note the two types of questions are similar as the system is telling the user its understanding of the user's query and is awaiting the user to either confirm or reject this understanding. We will implement four different scenarios: planning a trip, finding a restaurant, doing errands, and ordering a pizza. In more details,

- Planning a trip requires the system to gauge the user's origin, destination, departure time, preferable transportation methods (bus, subway, Uber, etc.), etc., after which a suggestion for the trip details will be provided.
- Finding a restaurant requires the system to gauge the user's preferable location, type of cuisines (American, Asian, Indian, Mexican, etc.), dietary restrictions (gluten-free, vegan, etc.), etc., after which a restaurant will be suggested according to the criteria provided.
- Doing errands requires the system to gauge the user's shopping list containing items with their brands and quantities, after which an executable shopping list will be generated.
- Ordering a pizza requires the system to gauge the user's preferable crust, toppings, sauce, etc., after which an executable pizza order will be generated.

The innovation is to embed counterfactual logic as updates in Knowledge Bases (KB). Specifically, we propose using a Symbolic Reasoning proof system that generates a probabilistic proof based on the current context and knowledge base and invokes an interaction with the user to check if the proof is correct and update KB if not. The updates include adding an atom, adding an action, adding a predicate, adding a rule, etc.

The challenges foreseeable are challenges in Natural Language Processing (entity resolution, domain-agnostic intent recognition, etc.) and challenges in Symbolic Reasoning. The latter, in particular, includes questions such as "What degree of freedom should be given to the users?" and "Do the users have infinite patience in providing instructions on updating the Knowledge base?".

If successful, this project will provide a new interaction paradigm that gives the users more power in defining personalized reasoning flows and therefore advancing the hands-free theme of conversational systems as a whole. This project can also be viewed as coding conversational logic as dynamic KB traversals, thus providing a new way of integrating Knowledge Graphs with conversational systems.

I will be working under the guidance of professor Aaron Steinfeld and research project scientist Oscar J. Romero at the Robotics Institute.

2 Project Goals

2.1 Natural Language Processing

The first step in constructing a conversational system is to have an NLP wrapper that does sentence parsing, intent recognition, entity resolution, etc. This is essential to the general performance of the system because any errors that occurred in this process will carry on to future reasoning analysis.

We have built, in the past summer, such a system based on Uber's Plato dialogue system; however, it is having some trouble dealing with intent recognition and entity resolution. We plan to make the system more robust by collecting more conversation data on these scenarios by running an Amazon Turk experiment, part of which will be used to tune the Machine Learning models in our NLP structure. The interface for this experiment is ready and will be published for data collection momentarily.

2.2 Symbolic Reasoning

The Symbolic Reasoning proof system will be the core of this project. The current plan is to use probabilistic Prolog - a logic programming language associated with computational linguistics. The rules encoded in Prolog represents the KB and will be dynamically updated through the conversations.

For each scenario, we need to initialize a set of rules that the program can use to "prove" each potential outcome, or more precisely to calculate, with "proof", the probability each potential outcome is likely. In the pizza ordering scenario, for instance, we can initialize the rule of the ingredients of a pizza order as

```
pizza(crust(C), topping([T1, T2, T3]), sauce(S), drink(D)) :-
    crust(C),
    topping([T1, T2, T3]),
    sauce(S),
    drink(D).
```

and the program will evaluate the probability of each ingredient combination as the product of the probability of each subcategory, which can be derived from either current conversational context (e.g. the user says `I really like thin crust pizza` then the probability of crust being `thin` should be higher than other options) or the KB itself (e.g. a more detailed rule defining how the probability of a list of toppings is calculated).

The system, when the most probable outcome is above a certain confidence level, will inform its understanding of the ingredients to the user and check for agreement. If the user disagrees, the system will be prompted to inquire ways to change the proof system to match the user's personalized criteria for determining the probability of each outcome, whether it be a major change of the entire structure of the pizza (e.g. the user says `A pizza is made of brick, metal, and glass`) or a minor change of one of the categories (e.g. the user says `I only want 2 toppings instead of 3`).

A critically important usage for the dataset introduced from the previous section will be for the training and testing of this proof system. While the participants are making conversations, with one simulating the bot and one simulating the user, we also require the bot to fill in a table with all information it obtained from the user on all necessary fields of the specific workflow problem. We will then be able to compare human interpretations with that derived from our reasoning module to train and test its effectiveness.

2.3 Goals

The exact 100% goal is to finish, for all four scenarios, refining the NLP module, establishing a baseline for the proof system, training it with a subset of the dataset we collected, and evaluating the performance on a separate test set.

If things go unexpectedly slow, the 75% goal is to reduce the number of scenarios to 3 or even 2. Notice that if we have one scenario running, all other scenarios are largely going to be replicates of it in terms of the general modeling pipeline (especially, the updating mechanism of the symbolic reasoning KB is shared across all workflow problems), where the only variations lie in the different intents in NLP and the different initialization rules in symbolic reasoning system. Despite the pursuit of building a scalable system, these two tasks can be safely postponed if time doesn't allow.

On the contrary, if things go unexpectedly smooth, the 125% goal is to tackle the question proposed in the description section "What degree of freedom should be given to the users?". This requires a significant amount of real-world testing to see how user's insensible actions can break the KB and/or the system, and strengthen the updating mechanism to avoid these issues.

3 Milestones

By the end of this semester: finish dataset collection and preprocessing, as well as the entire NLP module.

February 15th: define initialization rules for Scenario 1 and implement a basic boolean Prolog program that outputs yes/no for a given situation.

March 1st: upgrade the boolean component to a probabilistic Prolog program that outputs numerical probabilities for a given situation.

March 15th: formalize an exhaustive list of KB update rules and test with examples to ensure the updates are expected and finish the entire system on Scenario 1.

March 29th: repeat all above for Scenario 2.

April 12th: repeat all above for Scenario 3.

April 26th: repeat all above for Scenario 4.

May 10th: test all scenarios and critically analyze the model's performance.

4 Literature Search

I have read various papers on knowledge graph based conversational systems, as summarized in the previous assignment. They all provide insights into the critical problem of designing the update mechanism for the symbolic reasoning system. I am currently taking 15-317 in which basics on the programming language Prolog is being taught. I will further need to learn more about how to code in a probabilistic Prolog.

5 Resources Needed

I will be using the server from my mentor's lab, and I have access to it via ssh. I have all the resources to conduct this research, and if I find anything missing, I will seek help from Oscar, who will mostly be working toward a parallel goal.

6 Project Web Page

`https://AntianWang.github.io/15300-Project/`