LLM-Based NLU for Explanatory Dialogue

alexander.berman@gu.se

February 27, 2024

1 Introduction

This report presents and discusses the integration of a large language model (LLM) based natural-language understanding (NLU) component into an explanatory dialogue system. The dialogue system is a chatbot that assists users in a game that revolves around assessing whether people are introverted or extraverted based on their music preferences. This chatbot is embedded in an interactive web page together with other elements of the game such as visualizations of music preferences, audio controls and navigation buttons. (The game will be further elaborated in the course project report.)

2 Motivation

One of the reasons for opting for an LLM-based NLU component is to be able to interpret user utterances in a more accurate and robust way than with a simple keyword- or rule-based component. Furthermore, it was hypothesized that an LLM-based solution would require less effort in terms of development time than a more conventional machine-learning based approach (such as logistic regression or support vector machine for intent classification) due to the ability to use zero- or few-shot learning, which in principle can drastically reduce the amount of required training data. Finally, it was deemed that the fine-grained semantics underpinning the dialogue modeling in the project would not be supported well by a more conventional approach based on intent classification and entity extraction. In contrast, with an LLM the problem can be approached as a sequence-to-sequence task, i.e. as a transformation from natural language to an expression in formal language whose syntax and semantics is fed to the LLM in the prompt.

3 Problem formulation

In conventional approaches, NLU for dialogue systems is typically divided into two "tasks": intent classification and entity extraction. For example, the user utterance "I need a ticket to Paris" can be understood as expressing the intent `BookTicket` and the entity `Paris` (possibly assigned to the role `Destination`). In contrast, here the task as framed as a matter of expressing the meaning of a given natural-language utterance in a formal language. For example, the meaning of the utterance above might be expressed as a sequence of dialogue moves such as [`request(BookTicket)`, `answer(dest_city(Paris))`].

In general formal terms, given a user utterance U and a description L of a formal language, the NLU component should return an expression E in L that conveys the meaning of U. The value of L in the context of the specific domain is provided by Listing 1.

```
The output can be one of the following:
- Ask(Question): the user asks a question
- ICM(understanding, negative): the user signals negative
understanding
- ICM(acceptance, positive): the user acknowledges
- None: the input from the user does not match any of the
above

Question can be one of:
- Why(): bare why question
- Why(Proposition): explicit why question concerning a
proposition
- BooleanQuestion(Proposition): question concerning whether a
proposition is true
- WhQuestion(FactorsConsidered): question concerning which
factors the system considers

Proposition can be one of:
- Extraverted(): the person is considered extraverted
- Not(Extraverted()): the person is considered introverted
- HighValue(X): the person likes music with high X, where X
is an audio feature
- Not(HighValue(X)): the person likes music with low X, where
X is an audio feature
- Explains(Explanans, Explanandum): Explanans explains
Explanandum (both being propositions)
- Supports(Antecedent, Consequent): Antecedent supports
Consequent (both being propositions)

Audio features:
- energy_mean
- mode_0_percentage (low values indicate a preference for
music with minor mode, while high values indicate a
preference for major mode)
- loudness_mean
- speechiness_mean
- instrumentalness_mean
- valence_mean
- danceability_mean
```

Listing 1: Description of formal language for conveying meaning of user utterances in the domain at hand.

4 Implementation

5 Evaluation

Specifically, GPT 4 (ref OpenAI) is prompted to parse user utterances