Abstract
In writing programs that operate on natural language input, a common task is to recognize a specific utterance and extract specific portions of that utterance. We describe different methods for performing this task, focusing on training data requirements and ease of use. To address shortcomings in the easiest to use method, we propose a novel method which learns how to match a natural language template to the utterance, extracting information from the utterance in the process. We highlight how this approach differs from existing work in neural semantic parsing and sentence matching.

3 Overview of Existing Methods
There are many approaches to implementing this process, but three general categories of techniques stand out. The most recently developed method is to learn this entire process end-to-end, typically using Deep Reinforcement Learning. Although efforts are underway to decrease the amount of required data, this method requires significant labelled training specific to the system being built, often making it impractical (Weisz et al., 2018; Zhao and Tresp, 2018).

A more well-developed method is to parse the input into a structured form (e.g. a dependency graph), and use that to extract fields. This implements the first two steps in the above process. Since significant work has gone into developing natural language parsers, this method can be applied with minimal data to dialogue systems (Adams, 2017). However, this method makes dealing with semantically equivalent but grammatically different statements arduous. For a given response (e.g. setting a timer), each grammatical structure which triggers that response will require a function be implemented to process that grammatical structure. For many systems, this level of implementation effort is not worthwhile, especially if the cost of responding incorrectly is low (such as in virtual assistants).

To overcome these difficulties, information retrieval methods can be used (Leuski and Traum, 2011). Instead of specifying a function for each grammatical structure, an example can be provided in natural language. Information retrieval methods are capable of matching patterns similar to the provided example, making it possible to respond to many more unique utterances than is practical with parsing based techniques. However, while information retrieval methods can often determine the correct category of the utterance (e.g.
a greeting or a command to set a timer), they are unable to extract fields from the specific utterance (e.g. the time to set the timer to).

4 Similar Methods

The objective of this project is to provide a method with similar data requirements and ease of use as information retrieval methods, but with the ability to extract fields from the input. Specifically, instead of providing example natural language utterances directly, the implementor provides natural language templates that could be used to generate natural language utterances. These templates differ from utterances themselves by containing “slots” which can be filled in.

The most closely related work to this topic appears to be (Luo et al., 2018), which demonstrates various methods of combining neural networks and regular expressions. Since our templates can be seen as a very simple subtype of regular expression, their problem statement overlaps ours. However, our method varies in several ways from theirs. Since we provide the template to our model, it can learn how to handle imperfections in the template (such as grammatical variability or synonyms). In comparison, Luo et al. evaluate the regular expression directly, and use its output as features at various stages of their neural network. This requires the regular expression be more carefully written, to handle variability that our model learns to handle.

Another significant difference between the methods is that it is straightforward to transform an utterance into a template, given some method for labelling the slots in the utterance. This potentially allows our method to learn from unlabelled data, given some way of recognizing sufficiently similar utterances and choosing what phrases to replace with slots.

Another form of closely related work is in matching between natural language sentences, which is explored in (Wang et al., 2017). However, this method only measures the similarity between sentences, and does not allow for extracting slots. The problem statement could also be categorized as a drastically simplified case of neural semantic parsing. However, no neural semantic parsing work to date exists in this low-data regime (Kamath and Das, 2018).

5 Related Methods

Although we believe the proposal of this project to be novel, our design draws heavily from existing methods. Most obviously, it performs template matching with neural networks, which has previously been explored in the context of image and character recognition (Hashemi et al., 2016; Li et al., 2018). (Hashemi et al., 2016) explores a variety of traditional techniques applicable to template image matching, as well as how to integrate them with CNNs. (Li et al., 2018) trains a Siamese neural network to compare the input to a large number of templates. However, since transformer based models have proven to be much more effective for language tasks (Vaswani et al., 2017; Correia et al., 2018), we propose to instead train a bidirectional transformer between the template and input. Intuitively, this can be seen as learning how to translate between a “language” of templates and a “language” of input utterances. By forcing the model to “translate” from the provided input into the template, each slot can be filled using the translation into it. Furthermore, the likelihood that this template matches the input can be evaluated using the probability of producing the rest of the template without forcing. A similar approach of using translation techniques for other natural language techniques was explored in (Langner and Black, 2009).

One of the advantages of our proposed method is that templates can be generated from natural language utterances by replacing portions of the utterance with slots. Given pairs of utterances which are likely to have the same meaning, we propose to train our model in the following way. First, we generate a template from one of the utterances, by replacing some random portions of it with slots. Then, we use our model to fill those slots from the other utterance. Then, we train the model to minimize the difference between the original slot values and the filled values. Reconstruction losses similar to this have been shown to be very effective for unsupervised language learning (Devlin et al., 2018).

For our initial experiments, we propose to use the AskUbuntu Question Dataset, since it contains pairs of questions manually labelled as similar or not similar. However, we would also like to use larger datasets which contain less labelling, by using proxies for similarity such as time of posting between different utterances.
References


Rui Zhao and Volker Tresp. 2018. Efficient dialog policy learning via positive memory retention.