Training Shallow Phrase Structure Parsers From Semantic Frames and Simple NLG

Anonymous Author(s)
Affiliation
Address
email

Abstract

Determining the meaning of customer utterances is an important part of fulfilling customer requests in task-oriented dialogue. Natural Language Understanding (NLU) models can determine this meaning, but typically require many customer utterances that are hand-annotated with meaning representations, which are difficult to obtain and must be repeated for each new target domain. One way to reduce the labor involved in hand annotation is to have the human annotate a meaning representation (a "semantic frame" representation) separate from the corresponding utterance. In this work, we investigate the use of this approach in conjunction with several simple natural language generation (NLG) approaches in order to train shallow parsers to extract phrase structure representations from customer utterances. Our results show the effectivness of this approach for training NLU models.

1 Introduction

Task oriented spoken dialog systems enable automation of many activities such as question answering, home automation, and customer service. A problem with creating these systems is that typically large amount of training data must be prepared by hand in order to obtain models that achieve high performance. This problem is exacerbated if the system is supposed to handle complex tasks, because in this case the annotations of natural language understanding (NLU) data become more complex.

In this paper, we investigate the use of training a NLU model for a complex task requiring hierarchically structured annotations, where the training data is obtained through humans creating the hierarchical structure in real time as a dialogue progresses. Because of the real time constraint, the hierarchical structures that they produce are standoff from the words in the spoken utterance, rather than inline. Also because of this constraint, they are not required to produce any kind of speech transcription. Given that a set of these hierarchical structures are available, we investigate using as training data for NLU models the results of applying several different kinds of natural language generation (NLG) techniques on these structures.

This paper is structured as follows. First, we describe our human in the loop spoken dialog system, including the means in which a human can create a hierarchical annotation after listening to a user utterance. Second, we describe our target NLU models and then several NLG approaches that we use to generate training data for NLU from standoff hierarchical structures. Third, we provide details about the corpora that we use in our experiments before delving into the experimental procedures themselves and their results. Finally, we discuss related work and our conclusions.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

2 Target Spoken Dialog System

In this section, we discuss the architecture of our spoken dialog system, including the human in the loop. We then discuss in greater depth the NLU component of the system, including the hierarchical structures that our NLU model produces. Subsequently, we discuss how the human integrates into the architecture. Rather than producing hierarchical structures that are aligned with phrases in the user utterance, if it is required, the human produces standoff hierarchical structures which we also discuss.

The architecture of our spoken dialogue system is shown in Figure 1. A customer interacts with our system by phone. The speech signal of the customer's utterance is converted to a NLU representation via a pipeline consisting of automatic speech recognition (ASR) and NLU. This pipeline also returns a confidence score. If this score does not reach a certain threshold, the part of our system called the virtual assistant (VA) platform routes the same audio to a human domain expert (DE). Subsequently, through interacting with a graphical user interface, the DE constructs a semantic frame representation roughly equivalent to the NLU representation, which is sent back to the VA platform. Once the VA platform receives a representation, it routes it to the dialog manager (DM).

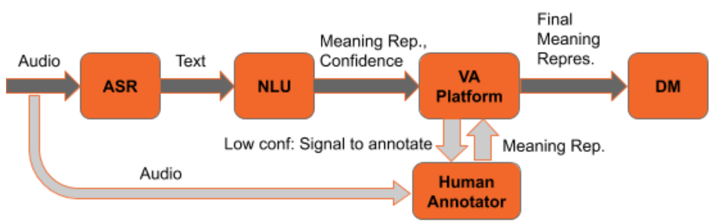

Figure 1: NLU outputs a structured representation given ASR output. Audio corresponding to low confidence NLU outputs are sent to a human domain expert (DE) for semantic frame annotation.

NLU receives as input from ASR an unpunctuated word sequence. It converts that into a structured representation consisting of entities, attributes, and intents. Entities and attributes correspond to noun heads and their modifiers, respectively. Simple head modifier relationships can be grouped into simple noun phrases called *attribute phrases*. In turn, entities, attribute, or attribute phrases themselves be composed into complex noun phrases called *items*. Finally, they can be grouped as arguments of a specific intent. An utterance may have one or more intents.

An example of the structured representation corresponding to the utterance "can i get two root beers oh and do you carry orange fanta" is shown in Figure 2. The various labels shown in the figure correspond to specific words or phrases in the utterance. The labels corresponding to entities/attributes, items, and intents are highlighted in red, green, and blue, respectively. The representation has a tree structure. It is able to represent hierarchical meanings that are necessary for the system to carry out the user request. There can be more than one intent in the same utterance; in this example, there are two intents.

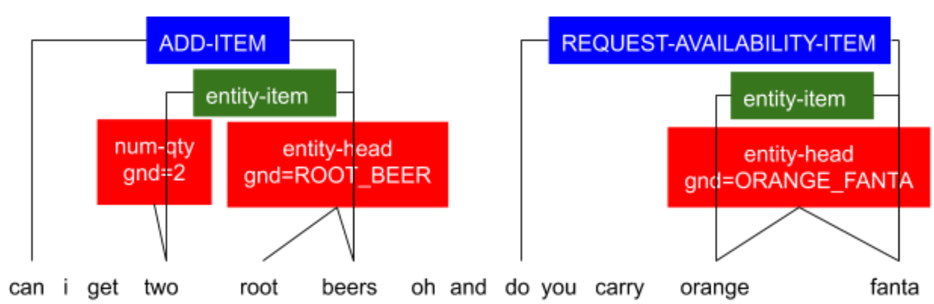

Figure 2: Hierarchical structured NLU representation for the utterance: *can i get two root beers oh and do you carry orange fanta*

Note that each entity or attribute has a *gnd* or "grounding" feature. The system maintains a *catalog* of orderable menu items. After NLU detects phrases in the input utterances corresponding to entities

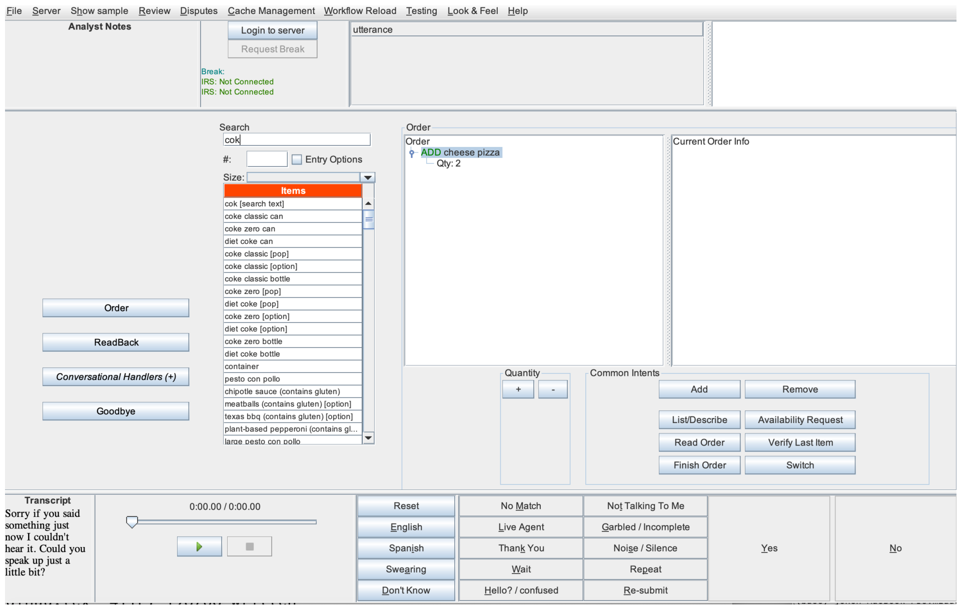

Figure 3: Human domain expert uses a point, click, and type user interface, namely the Domain Expert Desktop (DED), in order to capture the structured intent corresponding to the utterance *Can I have two cheese pizzas and a coke?*

and attributes, a grounding component uses these phrases to link entities and attributes to specific entries in the catalog. For example, in the example sentence, the phrase "root beers" is linked to the catalog entry ROOT_BEER.

If called upon by the VA platform, the DE listens to the audio of the customer utterance. The DE enters the customer's intent using a graphical interface called the Domain Expert Desktop (DED). See Figure 3. The DED contains a set of buttons corresponding to intents. Clicking some of these buttons, such as "yes" or "no," are sufficient to completely specify the corresponding intents. Clicking other buttons, such as "add" and "remove," specify intents taking entities or attributes as arguments. The action of clicking intent buttons results in the associated intents being listed in an DED window pane called "Order." Entities or attributes are specified using a text bar in conjunction with a clickable drop-down menu of entity and attribute names. They are attached to a particular intent by first highlighting that intent in the Order pane before clicking on an entity or attribute. The entries in the drop-down menu correspond to entries in the catalog.

The representation that the DE prepares by listening to the user audio and then using the DED is one or more semantic frames. Specifically, a *semantic frame* is a tree structure with the root being one intent, and its descendants being zero or more entities or attributes, corresponding to arguments of the intent, along with any substructure of each argument, if any. Example of semantic frames are shown in Table 1. There is one semantic frame corresponding to the *add-item* intent, having one argument manifested by the entity grounded as ROOT_BEER which itself contains an attribute reflecting the quantity 2. In a similar manner, The other semantic frame corresponds to the *request-availability-item* intent. It can be seen that it corresponds to the labeled structure in Figure 2, except for the absence of links from the representation to words and phrases in the utterance.

3 Natural Language Understanding

For each customer utterance, the natural language understanding (NLU) representation captures (a) one or more customer intents (b) entity heads, corresponding to noun heads of menu items, (c) attributes corresponding to modifiers of menu items, (d) attribute phrases and items corresponding to noun phrases with attribute and entity heads, respectively.

The NLU representation is predicted using a shallow parsing approach. In particular, four sequence taggers are used, each one predicting a different level in the NLU representation: (a) a head tagger

```
<add-item>
  <entity-item>
    <num-qty ground=2>
    <entity-head ground="ROOT_BEER">
  </entity-item>
</add-item>
<request-availability-item>
  <entity-item>
    <entity-head ground="ORANGE_FANTAS">
  </entity-item>
</request-availability-item>
```

Table 1: Example of semantic frames corresponding to the utterance: *can i get two root beers oh and do you carry orange fanta*

that tags entity heads and attributes, (b) an attribute phrase tagger that tags noun phrases with attribute heads, (c) an item tagger that tags noun phrases with entity heads, (d) an intent tagger that groups carrier phrases and their associate noun phrases together. Each tagger is implemented as a BiLSTM-CRF [1] with the input being the sequence of input words of the utterance.

Semantic frames are insufficient to train our NLU models. Although semantic frames do have semantic categories which our NLU models are trained to predict, they do not associate these categories with phrases in the customer utterance. In order to create this association, we experiment with several basic NLG approaches.

4 Natural Language Generation

Natural language generation (NLG) in this work performs surface realization on semantic frames. We experiment with several kinds of NLG. In *Basic NLG*, leaf nodes in the semantic frame corresponding to menu items or their optional components are replaced with word phrases representing their canonical names. See Figure 4 for an example of how Basic NLG operates on a sample semantic frame. In *CF NLG*, a context-free grammar is induced from a seed set of utterances that are hand-annotated with correct NLU outputs. The grammar is then employed as follows to perform surface realization.

CF NLG presupposes a context free grammar (CFG) $\langle N, T, R, S \rangle$ of nonterminals N, terminals T, rules R where each rule $r \in R$ is of the form $N_0 \rightarrow \alpha$ where $\alpha \in (N \cup T)^*$. Here, N consists of all of our intent, item, phrasal, entity, and attribute labels. For example, N includes intent label ADD and item label ENTITY_ITEM. Each entity label type also encodes the grounding annotation, e.g. "ENTITY_HEAD ground='PIZZA'" instead of "ENTITY_HEAD."

CF NLG takes as input a rooted tree consisting only of nonterminals. It processes these nonterminals one by one. For each nonterminal, the nonterminal and all of its children are used as a key into a dictionary whose value is the same children interspersed with zero or more sequences of terminals. The original children in the tree are replaced with the value from the dictionary. See Table 2 for pseudocode of CF NLG and Figure 5 for an example of running CFG NLG on a sample semantic frame, in the form of a tree of nonterminals.

The value of the dictionary also includes a natural number weight. The same dictionary key can map to more than one value. In that case, the value selected for a particular key is determined stochastically according to the various values' weights.

CF-NLG performs generation given a grammar G and an NLU representation N as follows. For each tree $T \in N$, starting from the root node $S \in T$, a breadth-first search of nonterminal nodes in T is performed. At each node N with children $C_1, C_2, \ldots C_m$, the key $N, C_1, C_2, .., C_m$ is formed, and a value is chosen from the dictionary at random, proportional to the weights of key value pairs in the dictionary. The chosen dictionary value is a rule N' that replaces the rule N in the tree.

Notation:
N nonterminals
T terminals
D dictionary, each entry of the form:
$A_0 A_1 A_2 \dots A_m \rightarrow \{ \langle \beta_0^1 A_1 \beta_1^1 A_2 \beta_2^1 \dots A_m \beta_m^1 ; w^1 \rangle,$
$\langle \beta_0^2 A_1 \beta_1^2 A_2 \beta_2^2 \dots A_m \beta_m^2 ; w^2 \rangle,$
$\dots,$
$\langle \beta_0^s A_1 \beta_1^s A_2 \beta_2^s \dots A_m \beta_m^s ; w^s \rangle \}$
where
$m \in \mathbb{N}$
$s \in \mathbb{N}$
$A_i^j \in N, 0 \leq i \leq m, 1 \leq j \leq s$
$\beta_i^j \in T^*, 0 \leq i \leq m, 1 \leq j \leq s$
$w \in \mathbb{N}$

Input: Tree X of nonterminals $B_1, B_2, \dots, B_n$

Algorithm:
for $i \leftarrow 1$ to n do
Let $C_1, C_2, \dots, C_q, q \in \mathbb{N}$ be the children of B_i in tree X.
Find values $\langle \alpha_1, w_1 \rangle, \langle \alpha_2, w_2 \rangle, \dots, \langle \alpha_r, w_r \rangle$
in dictionary D whose key is $B_i, C_1, C_2, \dots, C_q$.
where $\alpha_k = \beta_0 C_1 \beta_1 C_2 \beta_2 \dots C_q, 1 \leq k \leq r$ and $\beta_l \in T^*, 0 \leq l \leq q$
Randomly choose j from $1, \dots, r$ according to weights $w_1, \dots, w_r$.
Replace children $C_1, C_2, \dots, C_q$ of node B_i in X with α_j.
done

Output: Tree X augmented with nonterminals

Table 2: Pseudocode for CF NLG. Input is a tree of nonterminals only. Output is the same tree augmented with terminals.

5 Data

Our main target dataset includes a set of food ordering dialogues that occur between an agent and a customer. We perform most of our experiments on this dataset. The dialogues in this dataset are extracted from customer call logs from pizza restaurants. Out of all the utterances in these logs, only customer utterances that are relevant for food ordering have been selected. Therefore, utterances having to do with providing the customer name and address or having to do with specifying the payment method have been removed. After being hand transcribed, the chosen utterances were hand annotated with hierarchical NLU structures. The data is divided into a seed set, a development set, and a test set. They contain 632, 2240, and 795 utterances, respectively.

Besides the main dataset, there is a small subsidiary dataset of food ordering dialogues. There are 12 of these dialogues. Most of them involve pre-defined scenarios where the user tries to order several menu items. They were used to test the interaction between our DED and DM, exclusive of our ASR or NLU. This dataset contains 44 utterances, which is the subset of all of the utterances in these dialogues that correspond only to food ordering.

6 Experimental Design

Our main experiments involve comparing NLU models trained with hand-transcribed, hand-annotated corpora against those trained with semantic frames and various types of NLG. One baseline involves training on only the hand-transcribed, hand-annotated seed set (*Gold Small*) and testing on the test set. An upper bound involves training on hand-transcribed, hand-annotated development set in addition to the seed set (*Gold Large*). Another experiment involves training on seed set plus development set semantic frames after Basic NLG is applied (*Gold Small + Basic NLG*). The last experiment is the same, except CF NLG is applied to the semantic frame (*Gold Small + CF NLG*).

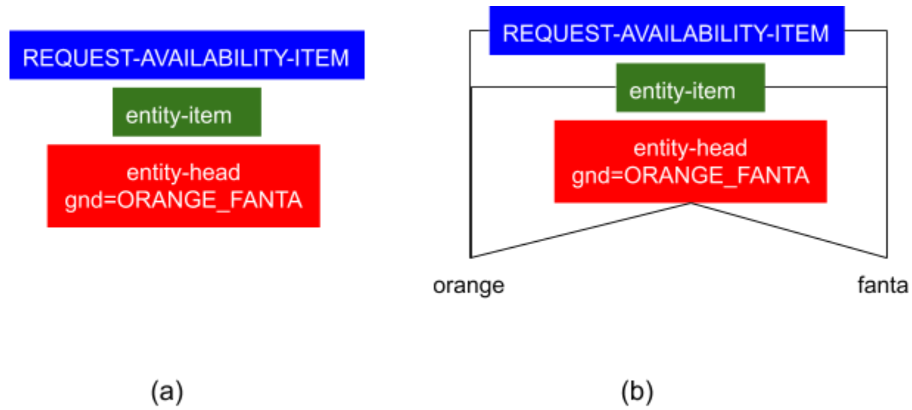

Figure 4: Example of operation of Basic NLG. The input semantic frame (a) is transformed into an output phrase structure representation (b).

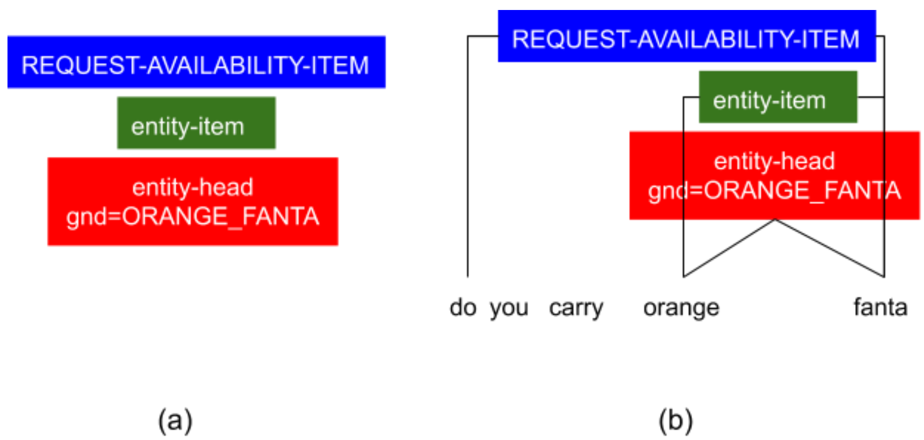

Figure 5: Example of operation of CF NLG. The input semantic frame (a) is transformed into an output phrase structure representation (b).

The experimental results exhibit a stochastic nature at several different points. First, applying NLG to semantic frames is probabilistic in its output. Second, training the BiLSTM-CRF multiple times results in model parameterizations with slight differences. In order to increase the reliability of our results, experiments were performed on five applications of NLG, and for each experimental setting, training of BiLSTM-CRF was repeated seven times. Results that are given below are averages over these repetitions.

We also performed another set of experiments in order to compare the amount of time it takes for a human to hand annotate text with NLU annotations versus the amount of time taken for a DE to produce semantic frames after listening to user audio. The human annotators were timed after a few month's experience with the domain dialogs and the type of annotation to be performed. They were timed for annotating our user utterances, which have been previously hand-transcribed from audio, comprising our main target dataset. The human DE was timed after a week or so of experience with the DED on this domain. The DE was timed for producing semantic frames for our subsidiary dialogs.

7 Results

7.1 Accuracy of NLU Models Trained with Different Kinds of Training Data

Results showing the accuracy of NLU models trained with different kinds of training data are shown in Table 3. The base model, trained on the Gold Small data set, achieves a labeled, bracketed F measure accuracy of 62.70%. When we add the Basic NLG version of the development set to the training data, the accuracy of the resulting model actually decreases by 1.5%. In contrast, if instead we add the CFG NLG version to the training data, the accuracy increases by 3.2% over baseline.

Looking at the results across different levels of tagging, we see that the addition of Basic NLG data is most effective at the lower levels of head and item tagging. Attribute phrase level results do not show improvement, but generally this level's results are not reliable because they occur quite infrequently in the data. As for the intent level, a regression in accuracy is seen, perhaps unsurprising because the Basic NLG data is devoid of carrier phrase intents. Switching focus to CF NLG data results, we see improvements at all levels of tagging. It is interesting that intent tagging should show an improvement, because CF NLG is not sophisticated enough to generate carrier phrases for intents that differ from those already found in the seed set (Gold Small). The improvement seems to have come from the new information that the semantic frames in the development set do provide, which includes information about possible placements of already known intents in multi-intent utterances and also information about how entities and items may be distributed inside an intent.

Eval Type	Training Data Type			
	Gold Small (base)	Gold Large (upper bound)	Gold Small + Basic NLG	Gold Small + CF NLG
Head	0.6904	0.7667	0.7101	0.7348
Attribute Phrase	0.0000	0.2784	0.0000	0.1097
Item	0.5923	0.6696	0.6047	0.6283
Intent	0.5846	0.6687	0.5142	0.6043
TOTAL	0.6270	0.7065	0.6111	0.6593

Table 3: Labeled, Bracketed F measure for NLU models trained with different types of training data. Up to 3% improvement in accuracy over baseline Gold Small is observed with Gold Small + CF NLG.

7.2 Speed of Hand Annotating Hierarchical Phrase Structures Versus Semantic Frames

The results of measuring the relative speed of hand annotating the data with either hierarchical NLU structures or semantic frames are as follows. On our subsidiary dialog dataset, a DE with minimal experience handling these types of dialogs took about 15 seconds to process each utterance. In contrast, a team of four people with a few months of experience with this dialog set took about 30 seconds to hand annotate each utterance. Because these measurements do not even take into account the time cost of hand transcribing the audio for the case of annotation with hierarchical NLU structures (unnecessary in the case of human annotating semantic frames after listening to audio) we find human semantic frame tagging to be much more efficient.

8 Related Work

Much previous work in training NLU models involves learning from a labeled corpus of utterances with inline NLU annotations. For example, [2] implement a CNN-CRF that performs joint intent detection and slot filling that is trained over a version of the ATIS corpus [3] that is tagged with intents and slots. [4] also train on hand-annotated data to obtain bi-directional RNNs that perform not only intent detection and slot filing, but also domain classification. [5] use a hand-annotated data set to extract hierarchical intent representations that are unlike the flat slot fillers used in ATIS but are more similar to our hierarchical representations.

There is also work in training NLU models that learn from a corpus of utterances where each utterance is paired with meaning representation that is not mapped to words or phrases in the utterance. They differ from our work in that the meaning representation is always some kind of logical form representation, whereas our representation are nested labeled brackets, each associated with a particular phrase in the input utterance. [6] train an NL parser that maps natural language utterances to a logical form representation using a log-linear model. [7] do the same using a hierarchical seq2seq modeling approach.

[8] jointly train an NLU model and an NLG model on data from the E2E Challenge [9]. Their model is not directly applicable to our task because our target semantic space is more structured, and because we are more interested in optimizing NLU performance rather than joint optimization of NLU and NLG performance.

Data augmentation is an approach where a model is created using a small hand annotated training set, and it is subsequently used to generate more data. Basic Seq2Seq [10], Seq2Seq that is aware of neighboring data samples [11], or extensions of variational autoencoders [12] can be used to for this kind of modeling. These works differ from ours in that their only input is a small hand annotated set, whereas we focus on how semantic frames can be leveraged for training improved models. Also, they assume simpler NLU representations than the ones we investigate.

Our semantic frame is a meaning representation where elements of the representation are not aligned to any words or phrases of its corresponding user utterance. [13] describes work on similar representations that pair natural language sentences with logical form queries. This is used to induce a parser via inductive logical programming that is capable of parsing sentences into queries that can be executed on a database of geographic information. Similarly, [14] describes work on representations pairing natural language sentences with lambda calculus expressions. From such pairs, they induce a CCG parser that is capable of parsing sentences into queries on geographic or jobs databases. While both [13] and [14] deal with limited domains, [15] describes work on creating Abstract Meaning Representation (AMR) corpora, which they term as "sembanking." The AMR corpus pairs natural language sentences from broad domains with AMRs. The aforementioned work talk about the advantages in speed of annotation of sentences with meaning representations where pairing representation elements with specific words or phrases is not required, but they do not present concrete timings as we do here that show how much more efficient this approach is over the alternative. Also, while [13] and [14] describe parsers built upon their meaning representations, their approaches are orthogonal to ours in that only we experiment with NLG approaches to create training data for parsers. Finally, none of their approaches discuss integration of human semantic frame annotation into any kind of human in the loop architecture, not to mention one that is used in a virtual assistant scenario.

9 Conclusions

We experiment with training NLU models with semantic frame meaning representations, as an labor saving alternative to training with inline representations. We experiment with training the NLU models by using as training data the results of applying NLG to the semantic frames. We find that by training with these semantic frames along with a simple NLG component (Basic NLG), we were able to obtain some improvement at the lower levels of the NLU output representation. By applying a slightly more sophisticated NLG (CF NLG) to a CFG component to the semantic frames, we obtained substantial increases in accuracy for all levels of NLU output. The impoverished nature of the NLG approaches suggests that increases in accuracy were due in part to the purely semantic information in the semantic frames and in part to the NLG linking semantic categories in the frames to specific phrases in utterances.

There are many avenues for future work. One possibility is to apply more sophisticated NLG approaches to the existing framework in order to examine the effect on the accuracy of the resulting NLU model. Another is to train NLU and NLG models in concert. A third possibility is to use online machine learning to incrementally train new NLU models as soon as new semantic frames are tagged by the DE.

Broader Impact

Insofar as this work will increase automation of tasks traditionally handled by human human dialog, a negative outcome would be that it may increase unemployment by decreasing the number of available service jobs. This must be balanced by the fact that service jobs are typically seen as undesirable because they are low wage jobs. Therefore, this work may have the positive outcome of increasing the proportion of desirable jobs out of all available jobs in an economy.

References

[1] Guilaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT 2016*, pages 260–270, 2016.

[2] Puyang Xu and Ruhi Sarikaya. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *Automatic Speech Recognition and Understanding (ASRU)*, pages 78–83, December 2013.

[3] Patti Price. Evaluation of spoken language systems: The atis domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

[4] Dilek Hakkani-Tür, Gokhan Tür, Asli Celikyilmaz, Yun-Nung Vivian Chen, Jianfeng Gao, Li Dung, and Ye yi Wang. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association (INTERSPEECH 2016)*, June 2016.

[5] Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, 2018.

[6] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI 05: Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 658–666, July 2005.

[7] Li Dong and Mirella Lapata. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany, August 2016. Association for Computational Linguistics.

[8] Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen. Dual supervised learning for natural language understanding and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5472–5477, Florence, Italy, 2019.

[9] Yevgeniy Puzikov and Iryna Gurevych. E2e nlg challenge: Neural models vs. templates. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 463–471, 2018.

[10] Gakuto Kurata, Bing Xiang, and Bowen Zhou. Labeled data generation with encoder-decoder lstm for semantic slot filling. In *INTERSPEECH*, pages 725–729, 2016.

[11] Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1234–1245, 2018.

[12] Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. Data augmentation for spoken language understanding via joint variational generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7402–7409, 2019.

[13] John M Zelle and Raymond J Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055, 1996.

[14] Luke S Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *arXiv preprint arXiv:1207.1420*, 2012.

[15] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186, 2013.