Modeling ASR Ambiguity for Neural Dialogue State Tracking

Vaishali Pal\textsuperscript{1,2}, Fabien Guillot\textsuperscript{1}, Manish Shrivastava\textsuperscript{3}, Jean-Michel Renders\textsuperscript{1}, Laurent Besacier\textsuperscript{1,2}

\textsuperscript{1}Naver Labs Europe  
\textsuperscript{2}LIG - Université Grenoble Alpes, France  
\textsuperscript{3}International Institute of Information Technology, Hyderabad, India

vaishali.pal@research.iiit.ac.in, jean-michel.renders@naverlabs.com, laurent.besacier@univ-grenoble-alpes.fr

Abstract

Spoken dialogue systems typically use one or several (top-N) ASR sequence(s) for inferring the semantic meaning and tracking the state of the dialogue. However, ASR graphs, such as confusion networks (confnets), provide a compact representation of a richer hypothesis space than a top-N ASR list. In this paper, we study the benefits of using confusion networks with a neural dialogue state tracker (DST). We encode the 2-dimensional confnet into a 1-dimensional sequence of embeddings using a confusion network encoder which can be used with any DST system. Our confnet encoder is plugged into the ‘Global-locally Self-Attentive Dialogue State Tacker’ (GLAD) model for DST and obtains significant improvements in both accuracy and inference time compared to using top-N ASR hypotheses.

Index Terms: speech recognition, dialog state tracking, confusion network, attention model

1. Introduction

Spoken task-oriented dialogue systems guide the user to complete a certain task through speech interaction. While such speech systems generally include an explicit automatic speech recognition (ASR) module (cascade approach), they now tend to be replaced by end-to-end approaches where the systems take speech as input and directly produce a decision from it. Examples include end-to-end architectures for spoken language understanding (SLU) proposed recently [1, 2, 3, 4]. However, those end-to-end models currently lead to equivalent but no better performance compared to cascade approaches based on ASR (see for instance [1, 2]). Besides, in some specific use cases, it may be preferable to deploy modular systems (instead of a monolithic one) for which only one component (ASR, SLU, dialog state tracker) can be modified at a time.

This article is positioned in this latter context and studies how to take better account of ASR ambiguity in voice-based dialog state tracking systems. Most recent work on spoken dialogue systems uses one or several (top-N) ASR sequence(s) to track the dialogue state and infer user needs. However, ASR lattices provide a richer hypothesis space than the top-N hypotheses. More precisely, we revisit the use of word confusion networks (simply denoted as confnets) [5], derived from ASR lattices, as a compact and efficient representation of ASR output.

To encode such graphical representations with existing state-of-the-art dialogue state trackers (DST), we introduce a generic neural confusion network encoder (see Figure 1) which can be used as a plug-in to any dialogue state tracker and achieves better results than using a list of top-N ASR hypotheses.

Our research contributions are the following:

• we introduce a system which encodes the confusion network into a representation that can be used as a plug-in to any state-of-the-art dialogue state tracker,
• we propose and experiment several variants to encode ASR confnets,
• we introduce mechanisms to leverage both ASR confusion network and true transcripts while training the DST system,
• we explore DST using a richer hypothesis space in the form of a confusion network and study whether it leads to better performance / computation trade-off compared to using a list of top-N ASR hypotheses.

2. Related Work

Dialog state tracking. Recent pieces of work on dialogue state trackers [6, 7, 8] infer the state of the dialogue from conversational history and current user utterance. These systems assume a text-based user utterance and accumulate the user goal across multiple user turns in the dialogue. [6] generalizes on rare slot-value pairs by using global modules which share parameters and local modules to learn slot specific feature representations. [7] achieves state-of-the-art performance on DSTC-2 dataset [9] with a universal state tracker which generates fixed-length representation for each slot and compares the distance between the representation and value vectors to make predictions. [10] predicts dialogue states from utterances and schema graphs containing slot relations in edges to achieve state-of-the-art result of MultiWoz 2.0 [11] and MultiWoz 2.1 datasets [12].

Using ASR graphs with neural models. Word lattices from ASR were used by [13] for intent classification in SLU with RNNs. Inspired from [13], [14] proposed to use word confusion networks for DST. However, there are several differences with what we propose in our paper: (1) they only use average pooling to aggregate the hidden GRU states corresponding to the alternative word hypotheses whereas we introduce several variants to pool alternative words (see section 3), (2) our word confusion network encoder can be plugged into any neural architecture for dialogue state tracking, as it basically amounts to have a first layer that transforms a 2D-data structure (confnets) into a 1D-sequence of embeddings, while [14] keeps the 2D-data structure in the hidden layers and, consequently, is limited to simple RNNs such as GRU and (bi-)LSTM, and (3) they experiment with a simpler RNN-based dialog state tracker while we plug our confnet encoder into the more efficient ‘Global-locally Self-Attentive Dialogue State Tacker’ (GLAD) model of [6]. Finally, our confnet encoder is most similar to [15] but they used ASR confnets for classification of user intent, question-type and named-entities while we apply our encoder to a DST task (we also propose several variants over [15]).
3. Word Confusion Network for DST

3.1. Confusion Network Encoder

Inspired from [15], we use a word confusion network encoder to transform the graph to a representation space which can be used with any dialogue state tracker. The multiple aligned ASR hypotheses, represented as parallel arcs at each position of the confusion network, are treated as a set of weighted arcs. More formally, a confnet $C$ is a sequence of parallel weighted arcs, noted as $C = \{\langle w_1^1, \pi_1^1 \rangle, \langle w_2^1, \pi_2^1 \rangle, \ldots, \langle w_m^1, \pi_m^1 \rangle, \ldots, \langle w_1^n, \pi_1^n \rangle, \langle w_2^n, \pi_2^n \rangle, \ldots, \langle w_m^n, \pi_m^n \rangle\}$, where $w_j^i$ is the $j^{th}$ arc (token) at time/position $t$, and $\pi_i^j$ its associated confidence weight. We propose several variants to formulate the embedding representation for a set $C_t$ of parallel arcs at position $t$ in the confnet $C$.

The simplest method to encode the confusion network is as a sequence of weighted-sums of the word embeddings weighted with the ASR confidence scores:

$$p_i^t = \pi_i^t \text{Embedding}(w_i^t)$$  \hspace{1cm} (1)

$$e_{v1}(C_t) = \sum_i p_i^t$$  \hspace{1cm} (2)

In increasing complexity, the second variant to encode the parallel arcs is to apply a weighted sum of non-linear transformations over word embeddings:

$$r_i^t = \pi_i^t \tanh(W_i^t \text{Embedding}(w_i^t))$$  \hspace{1cm} (3)

$$e_{v2}(C_t) = \sum_i r_i^t$$  \hspace{1cm} (4)

The third variation is to formulate the encoding with a self-attention mechanism similar to that described in [15]. In this case, $\pi_i^j$ (ASR) weights can be ignored as the model will use self-attention to weight the parallel arcs. The self-attention weights $\alpha_i^j$ are learnt during training:

$$q_i^t = \tanh(W_i^t \text{Embedding}(w_i^t))$$  \hspace{1cm} (5)

$$\alpha_i^t = \frac{\exp(w_i^t q_i^t)}{\sum_j \exp(w_j^t q_j^t)}$$  \hspace{1cm} (6)

$$e_{v3}(C_t) = \sum_i \alpha_i^t q_i^t$$  \hspace{1cm} (7)

$\text{Embedding}(w_i^t)$ is the embedding representation of word $w_i^t$. The final variation is to use the self-attention mechanism exactly as described in [15]. The ASR weights $\pi_i^j$ are used as an additional feature to weigh the word embeddings of each parallel arc:

$$\bar{q}_i^t = \tanh(W_i^t p_i^t)$$  \hspace{1cm} (8)

$$\alpha_i^t = \frac{\exp(w_i^t q_i^t)}{\sum_i \exp(w_i^t q_i^t)}$$  \hspace{1cm} (9)

$$e_{v4}(C_t) = \sum_i \alpha_i^t \bar{q}_i^t$$  \hspace{1cm} (10)

$e_{v4}$ denotes the 4 variations of the standard trainable embedding layer for word/token $w_i^t$; the matrix $W_i^t$ and the vector $w_2$ are trainable parameters of our model. Note that the training of these parameters is done jointly with the main task (see next subsection).

3.2. Dialogue State Tracking with Confnet

The dialogue state is a representation of the user goal at any time in the dialogue. A dialogue state tracker (DST) accumulates evidence as the dialogue progresses at each user turn and updates the state to reflect the changing user goals. The user goal is captured by the tracker as a distribution of slot-value pairs. Each user utterance can be either in textual or spoken form. Conventionally, DST uses top-N list of ASR hypotheses of the spoken user utterances to track the user needs. However, graph based representation such as ASR lattices and confusion network provides a richer hypothesis space in compact form.

Our confusion network encoder can be used as a plug-in to any state-of-the-art DST system. We have used the ‘Global-locally Self-Attentive Dialogue State Tacker’ (GLAD) model [6] with our confusion network encoder. GLAD addresses the issue of rare slot-value pairs which were not explicitly handled by previous DST models. The GLAD encoder module is a global-local self-attentive encoder which separately encodes the transcript/ASR hypothesis, system actions from previous turns and slot-value under consideration. We extend GLAD by replacing the user utterance representation, namely a sequence of trainable token embeddings, by the confnet embedding sequence. Remind that a confnet is also encoded as a 1 dimensional sequence of embeddings that corresponds to each time/position in the confnet. This enables GLAD architecture to use graph-based inputs instead of (or even in addition to) token sequence inputs.

4. Model Training Strategies

At training time, both clean transcript and ASR graph are available. It is therefore tempting to use these two pieces of information to facilitate model training while trying to make it robust to ASR errors in the meantime. We propose two radically different strategies to take into account clean transcript and ASR graph at training time.

4.1. Data Augmentation

The confusion network contains noisy hypotheses with lower ASR confidence scores which makes training hard. Augmenting the confnet dataset with the clean transcript should help the system to converge faster and better. We encode transcript in the form of a confnet with a single arc between nodes. At training time we merge both noisy ASR and clean (single arc) confnet datasets. Consequently, we use each dialog twice at training.
We evaluate our system on the standard Dialogue State Tracking Challenge 2 (DSTC-2) dataset [9]. DSTC is a research challenge focused on improving the state-of-the-art in tracking the state of spoken dialog systems. DSTC 2 contains dialogue in the restaurant information domain where the states may change through the dialogue. There are 1612, 506 and 1117 dialogues in the training, development and test sets respectively. The dialogue state is captured by (user-goal, turn-request, turn-inform). DSTC-2 provides representation of the user speech utterance in the form of top-10 ASR hypotheses and word confusion networks.

We followed a similar pre-processing pipeline of the confusion networks as mentioned in [14], i.e., we removed the interjections (um, ah, etc) and pruned arcs with an ASR confidence score less than 0.001 to reduce the size of the network, thus increasing efficiency without compromising the accuracy of models. To obtain a fair and consistent performance comparison between models based on ASR-top-N hypotheses and on confusion network, the top-N list of ASR hypotheses was chosen as the N best paths extracted from the confusion network.

6. Experiments and Results

Our baseline is the model trained on augmented dataset composed of ASR-N hypotheses and transcripts, where the final prediction is the weighted sum of prediction probabilities from each ASR hypothesis (Augmented ASR-N; see table 1). To demonstrate that a richer hypothesis space (confusion network) helps in improving accuracy, we trained 3 separate models on confusion networks with non-augmented (Non-augmented Cnet-N), augmented (Augmented Cnet-N) dataset and jointly-trained model (JCnet-N). In these models, we restricted the number of arcs per token by keeping only the ones with the top-N weights ($N \in [5, 9]$). The augmented dataset is composed of transcripts modeled as a graph with one arc between nodes, and ASR confusion networks. In addition to the above data augmentation techniques, we also evaluate the impact of using a similarity loss to train the confnet embeddings as introduced in section 4.2 (JCnet-N). We use a learning rate of 0.01, a batch size of 50, a dropout of 0.2 and a value of 0.5 to train our models. We concatenate pre-trained word embeddings (GloVe) [16] and Kazuma character embeddings [17] to encode words. The embedding layer is frozen and not updated during training.

<table>
<thead>
<tr>
<th>List-Size</th>
<th>Joint-Goal</th>
<th>Turn-Inform</th>
<th>Turn-Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR-1</td>
<td>0.6846(0.0017)</td>
<td>0.8326(0.0012)</td>
<td>0.9686(0.0003)</td>
</tr>
<tr>
<td>ASR-5</td>
<td>0.6980(0.0075)</td>
<td>0.8375(0.0050)</td>
<td>0.9680(0.0010)</td>
</tr>
<tr>
<td>ASR-9</td>
<td>0.6942(0.0077)</td>
<td>0.8395(0.0055)</td>
<td>0.9680(0.0073)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arc-Size</th>
<th>Joint-Goal</th>
<th>Turn-Inform</th>
<th>Turn-Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCnet-1</td>
<td>0.6883(0.0027)</td>
<td>0.8361(0.0021)</td>
<td>0.9672(0.0001)</td>
</tr>
<tr>
<td>JCnet-5</td>
<td>0.7088(0.0021)</td>
<td>0.8403(0.0015)</td>
<td>0.9773(0.0003)</td>
</tr>
<tr>
<td>JCnet-9</td>
<td>0.7063(0.0049)</td>
<td>0.8461(0.0048)</td>
<td>0.9700(0.0005)</td>
</tr>
</tbody>
</table>

Table 1: Scores for baseline model trained on augmented ASR-N list of hypothesis. Each cell contains the mean accuracy $\mu(0-1)$ and standard error $\sigma(0-1)$ in the format $\mu(\sigma)$ for 4 runs of each setting.

Table 2: Scores for Jointly-trained (Similarity Loss) model on augmented confnet dataset. Each cell contains the mean accuracy $\mu(0-1)$ and standard error $\sigma(0-1)$ in the format $\mu(\sigma)$ for 4 runs of each setting.
Figure 3: Attention Weights of the confusion network encoder (variant $e_{j}$) for the confnet with transcript ‘i don’t care about the price range i need basque food’. Dark colors represent high values and light colors low values. The columns represent subsequent parallel arcs. The parallel arcs are sorted from highest to lowest scored hypothesis. The 1st row is the best-pass through the network. The highest attention-weights are for the words ‘i don’t care about the price range i any basque food good’.

Table 3: Results on confusion network model encoding variants with different numbers of parallel arcs (5 or 9). With or w/o confnet augmentation. Each cell contains the mean accuracy $\mu$(0-1) and standard error $\sigma$(0-1) of the 3 metrics on 10 runs of each setting in the format $\mu$($\sigma$)

<table>
<thead>
<tr>
<th>Accuracy [$\mu$($\sigma$)]</th>
<th>Joint-Goal</th>
<th>Turn-Inform</th>
<th>Turn-Request</th>
<th>Joint-Goal</th>
<th>Turn-Inform</th>
<th>Turn-Request</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parallel Arcs: 5</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confnet Encoding Variation 1: $e_{v1} = \sum_{i} p_{i}$</td>
<td>0.7019(0.0026)</td>
<td>0.8350(0.0003)</td>
<td>0.9690(0.0004)</td>
<td>0.7008(0.0027)</td>
<td>0.8389(0.0002)</td>
<td>0.9677(0.0003)</td>
</tr>
<tr>
<td>Confnet Encoding Variation 2: $e_{v2} = \sum_{i} r_{i}$</td>
<td>0.7121(0.0019)</td>
<td>0.8470(0.0011)</td>
<td>0.9698(0.0004)</td>
<td>0.7115(0.0019)</td>
<td>0.8465(0.0011)</td>
<td>0.9686(0.0005)</td>
</tr>
<tr>
<td><strong>Parallel Arcs: 9</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confnet Encoding Variation 3: $e_{v3} = \sum_{i} \alpha_{i} q_{i}$</td>
<td>0.6912(0.0057)</td>
<td>0.8302(0.0037)</td>
<td>0.9645(0.0005)</td>
<td>0.6975(0.0016)</td>
<td>0.8361(0.0013)</td>
<td>0.9686(0.0004)</td>
</tr>
<tr>
<td>Confnet Encoding Variation 4: $e_{v4} = \sum_{i} \alpha_{i} q_{i}$</td>
<td>0.6925(0.0032)</td>
<td>0.8372(0.0008)</td>
<td>0.9650(0.0006)</td>
<td>0.7056(0.0023)</td>
<td>0.8413(0.0015)</td>
<td>0.9689(0.0007)</td>
</tr>
</tbody>
</table>

We train each experimental setting multiple times with different seeds and report the mean accuracy ($\mu$) and standard error ($\sigma$) of the joint-goal, turn-inform and turn-request for the ASR-N models in table 1, JCnet-N models in table 2 and the variants of confusion network encoder models in table 3. Our comments will mostly focus on the joint-goal accuracy metric which is the most important. We use 10 runs to calculate $\mu$ and $\sigma$ for Non-augmented Cnet-N and Augmented Cnet-N models and 4 runs to calculate $\mu$ and $\sigma$ for Augmented ASR-N and JCnet-N models. As illustrated in table 3, the joint-goal accuracy of the models trained on augmented dataset (leveraging clean transcripts at training time) performs better than those trained on non-augmented ones. These models also outperform the ASR-N baseline. The best variants seem to be $e_{v3}$ and $e_{v2}$ which do not use attention, both outperforming $e_{v4}$ [15]. However, attention models like $e_{v4}$ interestingly learn to assign higher weights to relevant words with lower ASR score over the top-1 hypothesis as shown in the figure 3. Content words, such as ‘basque’ which aids in classifier discriminability, is chosen by the attention-weights over function words such as ‘that’s’ in spite of the later being the top hypothesis. Furthermore, the performance of jointly-trained (similarity loss) confnet models in table 2 perform similar to the data augmented confnet models in table 3. Augmented confusion network with parallel arc size 5 (variant $e_{v5}$) outperforms jointly-trained model by a margin of 1% in joint-goal accuracy whereas both models have similar performance with parallel arc size 9.

The confusion network models lead to significant inference time gains over those trained on the list of ASR hypothesis. ASR-N models aggregate the predictions over each hypothesis to formulate the final prediction, resulting in a time complexity of $O(NM)$ where $N$ is the number of ASR hypothesis and $M$ is the size of the neural network. The confusion network models eliminate the additional time complexity introduced by the ASR hypothesis size without compromising on the rich hypothesis space. Thus, all variations of the confusion network models have an inference time complexity of $O(M)$. Inference time for all the varying confnet models (augmented, non-augmented, and joint goal) is on an average 0.82 sec per batch over a batch-size of 50. The average inference per batch for the ASR-N models progressively increases from 0.57 sec for ASR-1 to a maximum of 7 sec for ASR-10.

7. Conclusion

In this paper, we have demonstrated that exploiting the rich hypothesis space of a confusion network instead of being limited to the top-1 ASR hypotheses for DST leads to performance gain in time and accuracy. The time gain is significant if we want to incorporate a larger set of alternative ASR hypotheses. Moreover, we explore variations of designing the initial embedding layer transforming the confnet as a one-dimensional sequence of position-wise embeddings which can be plugged into numerous state-of-the-art text-based DST systems.
8. References


