Large-Scale Multi-Domain Belief Tracking with Knowledge Sharing

Osman Ramadan, Paweł Budzianowski, Milica Gašić
Department of Engineering,
University of Cambridge, U.K.
{oior2,pfb30,mg436}@cam.ac.uk

Abstract

Robust dialogue belief tracking is a key component in maintaining good quality dialogue systems. The tasks that dialogue systems are trying to solve are becoming increasingly complex, requiring scalability to multi-domain, semantically rich dialogues. However, most current approaches have difficulty scaling up with domains because of the dependency of the model parameters on the dialogue ontology. In this paper, a novel approach is introduced that fully utilizes semantic similarity between dialogue utterances and the ontology terms, allowing the information to be shared across domains. The evaluation is performed on a recently collected multi-domain dialogues dataset, one order of magnitude larger than currently available corpora. Our model demonstrates great capability in handling multi-domain dialogues, simultaneously outperforming existing state-of-the-art models in single-domain dialogue tracking tasks.

1 Introduction

Spoken Dialogue Systems (SDS) are computer programs that can hold a conversation with a human. These can be task-based systems that help the user achieve specific goals, e.g. finding and booking hotels or restaurants. In order for the SDS to infer the user goals/intentions during the conversation, its *Belief Tracking* (BT) component maintains a distribution of states, called a *belief state*, across dialogue turns (Young et al., 2010). The belief state is used by the system to take actions in each turn until the conversation is concluded and the user goal is achieved. In order to extract these belief states from the conversation, traditional approaches use a Spoken Language Understanding (SLU) unit that utilizes a semantic dictionary to hold all the key terms, rephrasings and alternative mentions of a belief state. The SLU then *delexicalises* each turn using this semantic dictionary, before it passes it to the BT component (Wang and Lemon, 2013; Henderson et al., 2014b; Williams, 2014; Zilka and Jurcicek, 2015; Perez and Liu, 2016; Rastogi et al., 2017). However, this approach is not scalable to multi-domain dialogues because of the effort required to define a semantic dictionary for each domain. More advanced approaches, such as the Neural Belief Tracker (NBT), use word embeddings to alleviate the need for *delexicalisation* and combine the SLU and BT into one unit, mapping directly from turns to belief states (Mrkšić et al., 2017). Nevertheless, the NBT model does not tackle the problem of mixing different domains in a conversation. Moreover, as each slot is trained independently without sharing information between different slots, scaling such approaches to large multi-domain systems is greatly hindered.

In this paper, we propose a model that jointly identifies the domain and tracks the belief states corresponding to that domain. It uses semantic similarity between ontology terms and turn utterances to allow for parameter sharing between different slots across domains and within a single domain. In addition, the model parameters are independent of the ontology/belief states, thus the dimensionality of the parameters does not increase with the size of the ontology, making the model practically feasible to deploy in multi-domain environments without any modifications. Finally, we introduce a new, large-scale corpora of natural, human-human conversations providing new possibilities to train complex, neural-based models. Our model systematically improves upon state-of-the-art neural approaches both in single and multi-domain conversations.

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 432–437
Melbourne, Australia, July 15 - 20, 2018.

2 Background

The belief states of the BT are defined based on an ontology - the structured representation of the database which contains entities the system can talk about. The ontology defines the terms over which the distribution is to be tracked in the dialogue. This ontology is constructed in terms of slots and values in a single domain setting. Or, alternatively, in terms of domains, slots and values in a multi-domain environment. Each domain consists of multiple slots and each slot contains several values, e.g. `domain=hotel`, `slot=price`, `value=expensive`. In each turn, the BT fits a distribution over the values of each slot in each domain, and a *none* value is added to each slot to indicate if the slot is not mentioned so that the distribution sums up to 1. The BT then passes these states to the Policy Optimization unit as full probability distributions to take actions. This allows robustness to noisy environments (Young et al., 2010). The larger the ontology, the more flexible and multi-purposed the system is, but the harder it is to train and maintain a good quality BT.

3 Related Work

In recent years, a plethora of research has been generated on belief tracking (Williams et al., 2016). For the purposes of this paper, two previously proposed models are particularly relevant.

3.1 Neural Belief Tracker (NBT)

The main idea behind the NBT (Mrkšić et al., 2017) is to use semantically specialized pre-trained word embeddings to encode the user utterance, the system act and the candidate slots and values taken from the ontology. These are fed to semantic decoding and context modeling modules that apply a *three-way* gating mechanism and pass the output to a non-linear classifier layer to produce a distribution over the values for each slot. It uses a simple update rule, $p(s_t) = \beta p(s_{t-1}) + \lambda y$, where $p(s_t)$ is the belief state at time step t, y is the output of the binary decision maker of the NBT and β and λ are tunable parameters.

The NBT leverages semantic information from the word embeddings to resolve lexical/morphological ambiguity and maximize the shared parameters across the values of each slot. However, it only applies to a single domain and does not share parameters across slots.

3.2 Multi-domain Dialogue State Tracking

Recently, Rastogi et al. (2017) proposed a multi-domain approach using *delexicalized* utterances fed to a two layer stacked bi-directional GRU network to extract features from the user and the system utterances. These, combined with the candidate slots and values, are passed to a feed-forward neural network with a softmax in the last layer. The candidate set fed to the network consists of the selected candidates from the previous turn and candidates from the ontology to a limit K, which restricts the maximum size of the chosen set. Consequently, the model does not need an ad-hoc belief state update mechanism like in the NBT.

The parameters of the GRU network are defined for the domain, whereas the parameters of the feed-forward network are defined per slot, allowing transfer learning across different domains. However, the model relies on *delexicalization* to extract the features, which limits the performance of the BT, as it does not scale to the rich variety of the language. Moreover, the number of parameters increases with the number of slots.

4 Method

The core idea is to leverage semantic similarities between the utterances and ontology terms to compute the belief state distribution. In this way, the model parameters only learn to model the interactions between turn utterances and ontology terms in the semantic space, rather than the mapping from utterances to states. Consequently, information is shared between both slots and across domains. Additionally, the number of parameters does not increase with the ontology size. Domain tracking is considered as a separate task but is learned jointly with the belief state tracking of the slots and values. The proposed model uses semantically specialized pre-trained word embeddings (Wieting et al., 2015). To encode the user and system utterances, we employed 7 independent bi-directional LSTMs (Graves and Schmidhuber, 2005). Three of them are used to encode the system utterance for domain, slot and value tracking respectively. Similarly, three Bi-LSTMs encode the user utterance while and the last one is used to track the user affirmation. A variant of the CNNs as a feature extractor, similar to the one used in the NBT-CNN (Mrkšić et al., 2017) is also employed.

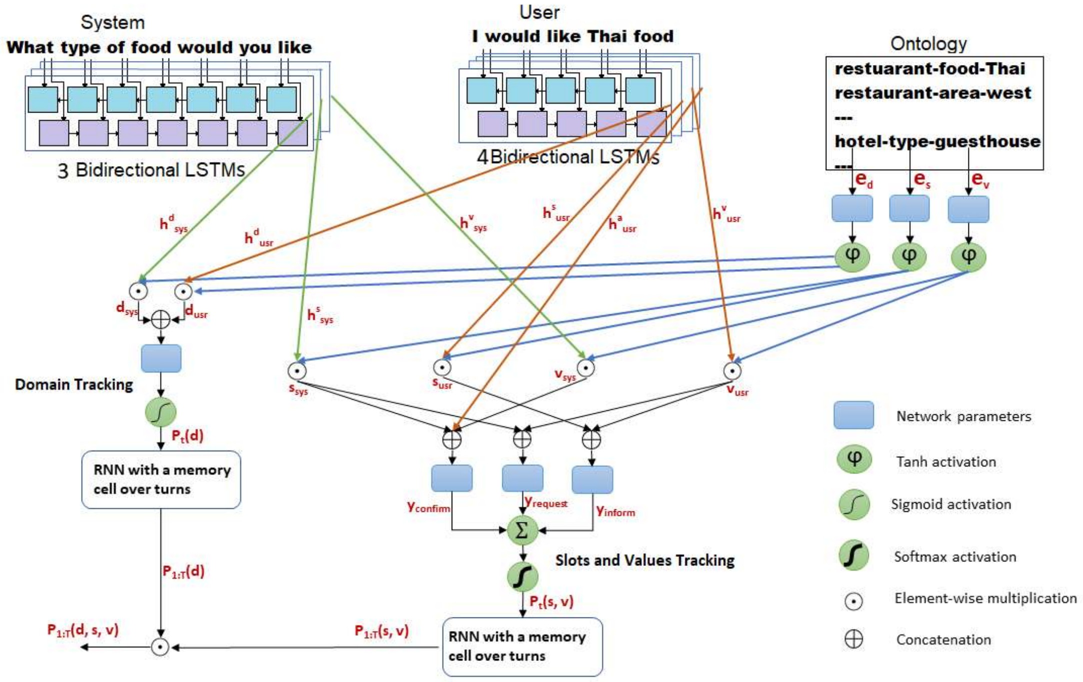

Figure 1: The proposed model architecture, using Bi-LSTMs as encoders. Other variants of the model use CNNs as feature extractors (Kim, 2014; Kalchbrenner et al., 2014).

4.1 Domain Tracking

Figure 1 presents the system architecture with two bi-directional LSTM networks as information encoders running over the word embeddings of the user and system utterances. The last hidden states of the forward and backward layers are concatenated to produce $\mathbf{h}^d_{usr}, \mathbf{h}^d_{sys}$ of size L for the user and system utterances respectively. In the second variant of the model, CNNs are used to produce these vectors (Kim, 2014; Kalchbrenner et al., 2014). To detect the presence of the domain in the dialogue turn, element-wise multiplication is used as a similarity metric between the hidden states and the ontology embeddings of the domain:

$$\mathbf{d}_k = \mathbf{h}^d_k \odot \tanh(\mathbf{W}_d\, \mathbf{e}_d + \mathbf{b}_d),$$

where $k \in \{usr, sys\}$, $\mathbf{e}_d$ is the embedding vector of the domain and $\mathbf{W}_d \in \mathcal{R}^{L\times D}$ transforms the domain word embeddings of dimension D to the hidden representation. The information about semantic similarity is held by $\mathbf{d}_{usr}$ and $\mathbf{d}_{sys}$, which are fed to a non-linear layer to output a binary decision:

$$\mathcal{P}_t(d) = \sigma(\mathbf{w}_d \left\{\mathbf{d}_{usr} \oplus \mathbf{d}_{sys}\right\} + b_d),$$

where $\mathbf{w}_d \in \mathcal{R}^{2L}$ and b_d are learnable parameters that map the semantic similarity to a belief state probability $\mathcal{P}_t(d)$ of a domain d at a turn t.

4.2 Candidate Slots and Values Tracking

Slots and values are tracked using a similar architecture as for domain tracking (Figure 1). However, to correctly model the context of the system-user dialogue at each turn, three different cases are considered when computing the similarity vectors:

1. **Inform:** The user is informing the system about his/her goal, e.g. *'I am looking for a restaurant that serves Turkish food'*.
2. **Request:** The system is requesting information by asking the user about the value of a specific slot. If the system utterance is: *'When do you want the taxi to arrive?'* and the user answers with *'19:30'*.
3. **Confirm:** The system wants to confirm information about the value of a specific slot. If the system asked: *'Would you like free parking?'*, the user can either affirm positively or negatively. The model detects the user affirmation, using a separate bi-directional LSTM or CNN to output $\mathbf{h}^a_{usr}$.

The three cases are modelled as following:

$$\begin{aligned}
y^{s,v}_{inf} &= \mathbf{w}_{inf}\left\{\mathbf{s}_{usr} \oplus \mathbf{v}_{usr}\right\} + b_{inf},\\
y^{s,v}_{req} &= \mathbf{w}_{req}\left\{\mathbf{s}_{sys} \oplus \mathbf{v}_{usr}\right\} + b_{req},\\
y^{s,v}_{af} &= \mathbf{w}_{af}\left\{\mathbf{s}_{sys} \oplus \mathbf{v}_{sys} \oplus \mathbf{h}^a_{usr}\right\} + b_{af},
\end{aligned}$$

where $\mathbf{s}_k, \mathbf{v}_k$ for $k \in \{usr, sys\}$ represent semantic similarity between the user and system utterances and the ontology slot and value terms respectively computed as shown in Figure 1, and $\mathbf{w}$ and b are learnable parameters.

The distribution over the values of slot s in domain d at turn t can be computed by summing the unscaled states, y_{inf}, y_{req} and y_{af} for each value v in s, and applying a softmax to normalize the distribution:

$$\mathcal{P}_t(s, v) = \text{softmax}(y^{s,v}_{inf} + y^{s,v}_{req} + y^{s,v}_{af}).$$

4.3 Belief State Update

Since dialogue systems in the real-world operate in noisy environments, a robust BT should utilize the flow of the conversation to reduce the uncertainty in the belief state distribution. This can be achieved by passing the output of the decision maker, at each turn, as an input to an RNN that runs over the dialogue turns as shown in Figure 1, which allows the gradients to be propagated across turns. This alleviates the problem of tuning hyper-parameters for rule-based updates. To avoid the vanishing gradient problem, three networks were tested: a simple RNN, an RNN with a memory cell (Henderson et al., 2014a) and a LSTM. The RNN with a memory cell proved to give the best results. In addition to the fact that it reduces the vanishing gradient problem, this variant is less complex than an LSTM, which makes training easier. Furthermore, a variant of RNN used for domain tracking has all its weights of the form: $\mathbf{W}_i = \alpha_i \mathbf{I}$, where α_i is a distinct learnable parameter for hidden, memory and previous state layers and $\mathbf{I}$ is the identity matrix. Similarly, weights of the RNN used to track the slots and values is of the form: $\mathbf{W}_j = \gamma_j \mathbf{I} + \lambda_j (\mathbf{1} - \mathbf{I})$, where γ_j and λ_j are the learnable parameters. These two variants of RNN are a combination of Henderson et al. (2014a) and Mrkvsić and Vulić (2018) previous works. The output is $\mathcal{P}_{1:T}(\mathbf{d})$ and $\mathcal{P}_{1:T}(\mathbf{s}, \mathbf{v})$, which represents the joint probability distribution of the domains and slots and values respectively over the complete dialogue. Combining these together produces the full belief state distribution of the dialogue:

$$\mathcal{P}_{1:T}(\mathbf{d}, \mathbf{s}, \mathbf{v}) = \mathcal{P}_{1:T}(\mathbf{d})\mathcal{P}_{1:T}(\mathbf{s}, \mathbf{v}).$$

4.4 Training Criteria

Domain tracking and slots and values tracking are trained disjointly. Belief state labels for each turn are split into domains and slots and values. Thanks to the disjoint training, the learning of slot and value belief states are not restricted to a specific domain. Therefore, the model shares the knowledge of slots and values across different domains. The loss function for the domain tracking is:

$$\mathcal{L}_d = -\sum_{n=1}^{N} \sum_{\mathbf{d} \in \mathcal{D}} t^n(\mathbf{d}) \log \mathcal{P}^n_{1:T}(\mathbf{d}),$$

where $\mathbf{d}$ is a vector of domains over the dialogue, $t^n(\mathbf{d})$ is the domain label for the dialogue n and N is the number of dialogues. Similarly, the loss function for the slots and values tracking is:

$$\mathcal{L}_{s,v} = -\sum_{n=1}^{N} \sum_{\mathbf{s},\mathbf{v} \in \mathcal{S},\mathcal{V}} t^n(\mathbf{s}, \mathbf{v}) \log \mathcal{P}^n_{1:T}(\mathbf{s}, \mathbf{v}),$$

where $\mathbf{s}$ and $\mathbf{v}$ are vectors of slots and values over the dialogue and $t^n(\mathbf{s}, \mathbf{v})$ is the joint label vector for the dialogue n.

5 Datasets and Baselines

Neural approaches to statistical dialogue development, especially in a task-oriented paradigm, are greatly hindered by the lack of large scale datasets. That is why, following the Wizard-of-Oz (WOZ) approach (Kelley, 1984; Wen et al., 2017), we ran text-based multi-domain corpus data collection scheme through Amazon MTurk. The main goal of the data collection was to acquire human-human conversations between a tourist visiting a city and a clerk from an information center. At the beginning of each dialogue the user (visitor) was given explicit instructions about the goal to fulfill, which often spanned multiple domains. The task of the system (wizard) is to assist a visitor having an access to databases over domains. The WOZ paradigm allowed us to obtain natural and semantically rich multi-topic dialogues spanning over multiple domains such as hotels, attractions, restaurants, booking trains or taxis. The dialogues cover from 1 up to 5 domains per dialogue greatly varying in length and complexity.

5.1 Data Structure

The data consists of 2480 single-domain dialogues and 7375 multi-domain dialogues usually spanning from 2 up to 5 domains. Some domains consists also of sub-domains like booking. The average sentence lengths are 11.63 and 15.01 for users

	WOZ 2.0			New WOZ (only restaurants)		
Slot	**NBT-CNN**	**Bi-LSTM**	**CNN**	**NBT-CNN**	**Bi-LSTM**	**CNN**
Food	88.9	96.1	**96.4**	78.3	84.7	**85.3**
Price range	93.7	**98.0**	97.9	92.6	**95.6**	93.6
Area	94.3	97.8	**98.1**	78.3	82.6	**86.4**
Joint goals	84.2	85.1	**85.5**	57.7	59.9	**63.7**

Table 1: WOZ 2.0 and new dataset test set accuracies of the NBT-CNN and the two variants of the proposed model, for slots *food, price range, area* and *joint goals.*

and wizards respectively. The combined ontology consists of 5 domains, 27 slots and 663 values making it significantly larger than observed in other datasets. To enforce reproducibility of results, we distribute the corpus with a pre-specified train/test/development random split. The test and development sets contain 1k examples each. Each dialogues consists of a goal, user and system utterances and a belief state per turn. The data and model is publicly available.[1]

5.2 Evaluation

We also used the extended WOZ 2.0 dataset (Wen et al., 2017).[2] WOZ2 dataset consists of 1200 single topic dialogues constrained to the restaurant domain. All the weights were initialised using normal distribution of zero mean and unit variance and biases were initialised to zero. ADAM optimizer (Kingma and Ba, 2014) (with 64 batch size) is used to train all the models for 600 epochs. Dropout (Srivastava et al., 2014) was used for regularisation (50% dropout rate on all the intermediate representations). For each of the two datasets we compare our proposed architecture (using either Bi-LSTM or CNN as encoders) to the NBT model[3] (Mrkšić et al., 2017).

6 Results

Table 1 shows the performance of our model in tracking the belief state of single-domain dialogues, compared to the NBT-CNN variant of the NBT discussed in Section 3.1. Our model outperforms NBT in all the three slots and the joint goals for the two datasets. NBT previously achieved state-of-the-art results (Mrkšić et al., 2017). Moreover, the performance of all models is worse on the new dataset for restaurant compared to WOZ 2.0. This is because the dialogues in the new dataset are richer and more noisier, as a closer resemblance to real environment dialogues.

New WOZ (multi-domain)		
Model	**F1 score**	**Accuracy %**
Uniform Sampling	0.108	10.8
Bi-LSTM	0.876	**93.7**
CNN	**0.878**	93.2

Table 2: The overall F1 score and accuracy for the multi-domain dialogues test set.[4]

Table 2 presents the results on multi-domain dialogues from the new dataset described in Section 5. To demonstrate the difficulty of the multi-domain belief tracking problem, values of a theoretical baseline that samples the belief state uniformly at random are also presented. Our model gracefully handles such a difficult task. In most of the cases, CNNs demonstrate better performance than Bi-LSTMs. We hypothesize that this comes from the effectiveness of extracting local and position-invariant features, which are crucial for semantic similarities (Yin et al., 2017).

7 Conclusions

In this paper, we proposed a new approach that tackles the issue of multi-domain belief tracking, such as model parameter scalability with the ontology size. Our model shows improved performance in single-domain tasks compared to the state-of-the-art NBT method. By exploiting semantic similarities between dialogue utterances and ontology terms, the model alleviates the need for ontology-dependent parameters and maximizes the amount of information shared between slots and across domains. In future, we intend to investigate introducing new domains and ontology terms without further training thus performing zero-shot learning.

[1] http://dialogue.mi.eng.cam.ac.uk/index.php/corpus/

[2] Publicly available at `https://mi.eng.cam.ac.uk/~nm480/woz_2.0.zip`.

[3] Publicly available at `https://github.com/nmrksic/neural-belief-tracker`.

[4] F1-score is computed by considering all the values in each slot of each domain as positive and the ”none” state of the slot as negative.

Acknowledgments

The authors would like to thank Nikola Mrkšić, Jacquie Rowe, the Cambridge Dialogue Systems Group and the ACL reviewers for their constructive feedback. Paweł Budzianowski is supported by EPSRC Council and Toshiba Research Europe Ltd, Cambridge Research Laboratory. The data collection was funded through Google Faculty Award.

References

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks* 18(5-6):602–610.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014a. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, pages 360–365.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014b. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. pages 292–299.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *In Proceedings of ACL* .

John F Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)* 2(1):26–41.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *In Proceedings of EMNLP* .

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *ICLR* .

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1777–1788.

Nikola Mrkvsić and Ivan Vulić. 2018. Fully statistical neural belief tracking. In *Proceedings of ACL*.

Julien Perez and Fei Liu. 2016. Dialog state tracking, a machine reading approach using memory network. *arXiv preprint arXiv:1606.04052* .

Abhinav Rastogi, Dilek Hakkani-Tur, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. *arXiv preprint arXiv:1712.10224* .

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*. pages 423–432.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. *In Proceedings on EACL* .

John Wieting, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Dan Roth. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics* 3:345–358.

Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse* 7(3):4–33.

Jason D Williams. 2014. Web-style ranking and slu combination for dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. pages 282–291.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923* .

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language* 24(2):150–174.

Lukas Zilka and Filip Jurcicek. 2015. Incremental lstm-based dialog state tracker. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, pages 757–762.