Baby Jarvis: Final Report
CSE 481 N

Divye Jain, Vardhman Mehta, Kevin Zhao

June 2019

Abstract

Task-oriented dialogue systems have seen a recent boom, especially due to the advent and popularity of virtual assistants like Alexa and Google Assistant. One of the core components of these modern dialogue systems is the belief tracker, which approximates the user's goal at each step in the conversation. There has been a significant amount of work done to design these state trackers, like an End-to-End Trainable Task-oriented Dialogue System [Wen et al., 2016] or a more recent model like the Neural Belief Tracker [Mrksic et al., 2016]. In this capstone project we propose an improvement upon the latest iteration on the pre-existing **N**eural **B**elief **T**racker (NBT) model [Mrksic and Vulic, 2018] called **N**eural **B**elief **T**racker - **Bi**directional **L**STM + scalar **A**ttention (NBT-BiLA).

1 Introduction

Statistical dialog systems are created to solve the problem of understanding language. A critical component of these system is the dialogue state tracking (DST) module which is used to model the user's intent at any step in the conversation. These DST models are dependent on domain-specific Spoken Language Understanding (SLU) modules to extract turn-level user intents. The user intents are later consumed by the belief state update mechanism, which is responsible for generating the system's internal probability distribution over the potential dialogue states. The dialogue states are defined by a domain specific ontology: representing the constraints of the user through a collection of slots (e.g. *price range*) and their slot values (e.g. *cheap* or *expensive*) in the case of a restaurant search task. These belief states are then used by the downstream dialogue manager to decide on a system action [Su et al., 2016a,b]. In this work we will primarily focus on the DST component of the task-oriented dialogue system.

Work prior to the NBT model required substantial amounts of domain-specific annotation [Perez and Liu, 2016b, Sun et al., 2015, Vodolán et al., 2017, Wang and Lemon, 2013] or relied on manually constructed semantic dictionaries to identify alternative mentions of ontology items that vary lexically or morphologically [Henderson et al., 2014b,c, Williams et al., 2016]. The NBT framework is an attempt to overcome these obstacles by using dense word embeddings in place of traditional n-gram features. By making use of semantic relations embedded in the vector spaces, the NBT achieves DST performance competitive to lexicon-supplemented delexicalisation-based models without relying on any hand-crafted resources. Moreover, the latest iteration of this model eliminates the rule-based NBT component that was used in previous work, effectively yielding a fully statistical dialogue state tracker.

Contribution: In this project we propose a model that uses the architecture of the **NBT** model [Mrksic and Vulic, 2018] but makes improvement to the *representation learning* module and the *semantic decoding* module with the use of **BiL**STMs and **A**ttention. We call this model NBT-BiLA. This work also shows that the NBT-BiLA improves upon the previous NBT-CNN model on the Wizard-of-OZ v.20 (WoZ2.0) [Wen et al., 2016] and the second Dialogue State Tracking Challenge (DSTC2) [Henderson et al., 2014a] datsets.

2 Technical Idea Section(s)

Belief tracking models are used to capture a user's goal given their utterances and system responses. Constrained sets of goals are represented using *slot-value* mappings such as [food: *italian*] or [price-range: *moderate*]. This set of slots $\mathbf{S}$ and values $\mathbf{V_s}$ make up the ontology for each domain of user-system utterances.

We start our exploration using the CNN architecture described by Mrksic and Vulic [2018]. They illustrate a novel way that (1) extracts n-gram feature representations of the user utterances **r**; (2) use the current candidate slot-value pairs **c** and (3) the preceding system act **m** to make a decision about the slot value pair for each user utterance. Rather than using SLU decoder to create a meaning representation of the user utterances, the system uses turn inputs to extract a large number of n-gram features which do not necessarily capture the relevant meaning of the user's turn.

We make our contributions to the representation learning module **r** and the interaction between utterance representation **r** and the candidate representation **c**. As shown in figure 1, NBT-BiLA is now comprised of an encoder module that consists of a bidirectional LSTM over the user utterances to create **r**. A Scaled Dot product Attention layer is added over the interaction of the candidate representation and the user utterances to produce a representation $\mathbf{cr}'$. This representation is then added in the *semantic decoding* layer and the *context modelling* downstream components, and are finally combined to form a decision for the current slot-value pair. For full details of this downstream setup see the NBT paper [Mrksic et al., 2016].

2.1 BiLSTM + Scaled Dot Product Attention encoder

DST datasets tend to be small with a significant proportion of training data dedicated to just several examples of a slot-value pair. Hence, it is necessary to extract the semantic meaning of a user's dialogue while also paying attention to specific words, given the slot-value context. Extracting n-gram features from the user dialog to make the representation **r** captures some of the dialog dynamics but it does not encapsulate the entire semantic meaning. Nor does it look at the context of each word with respect to the slot-value representation. We begin by introducing the BiLSTM + Scaled Dot Product Attention encoder, which makes up the encoder module. In developing this encoder we try to capture these necessary ideas into feature representations to improve the neural belief tracker.

The NBT-BiLA consists of a bidirectional LSTM [Hochreiter and Schmidhuber, 1997], which captures the semantic representation of the entire sequence. For a sequence of length n for a particular slot s, let d_{emb} be the dimension of the word embedding of the sequence and $\mathbf{r} \in \mathbb{R}^{n \times d_{emb}}$ the word embeddings corresponding to the words in the sequence. We produce an encoding $S^{\mathbf{r}}$ of **r** using the bidirectional LSTM.

$$S^{\mathbf{r}} = \text{biLSTM}(\mathbf{r}) \in \mathbb{R}^{n \times d_{rnn}} \tag{1}$$

where d_{rnn} is the dimension of the LSTM state. We produce an equivalent context representation $H^c \in \mathbb{R}^{s_{v-dim}}$ of the current slot-values from **c** where s_{v-dim} are the number of values for the slot s. For each ith element $S_i^{\mathbf{r}}$, we compute the scaled dot product attention score$(S_t, H) \in \mathbb{R}^{s_{v-dim} \times n}$:

$$\mathbf{cr}' = \text{score}(S_t, H) = \sum_{i}^{d_{rnn}} \frac{H^C S_i^{\mathbf{r}}}{\sqrt{d_{rnn}}} \tag{2}$$

This is then added to the existing *semantic decoding* layer downstream. The scaling factor of d_{rnn} is added as a regularizer in a similar way described by Vaswani et al. [2017].

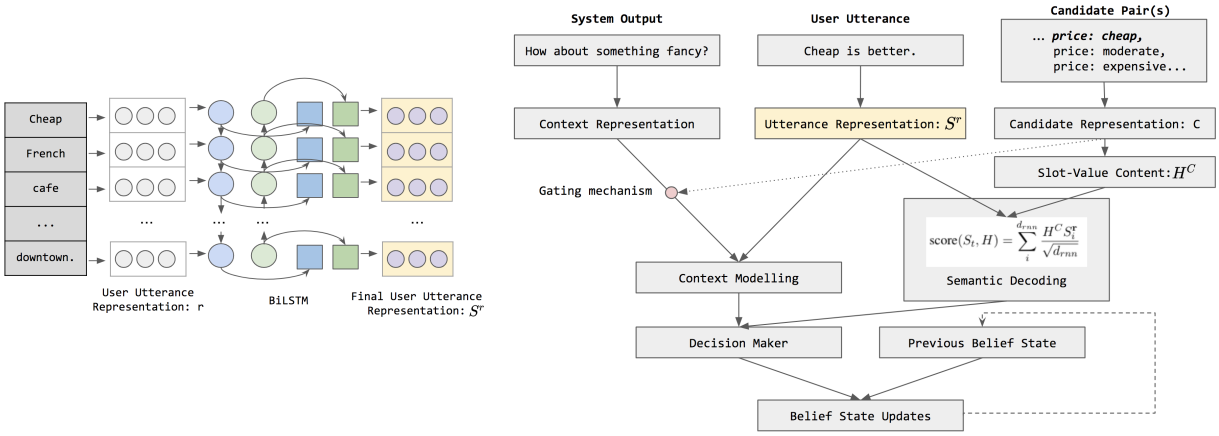

Figure 1: NBT-BiLA Model. A bidirectional LSTM layer is added on top of the user utterances representation. The model concatenates the representation from the forward and backward layers. This representation $S^{\mathbf{r}}$ is then coalesced with the context representation H^c of the slot value pairs by adding an Attention layer. The output is then added to the *semantic decoding* layer of the original NBT model.

2.2 ELMo word embeddings for user utterance

Pre-trained word vectors can be used to handle semantic variation within a piece of dialogue and produce high-quality intermediate representations which can be fed into a belief tracking model. One pre-trained word embedding we explored were the ELMo embeddings [Peters et al., 2018]. Our intuition was that the deep contextualized word representations within ELMo would improve performance metrics in our models by giving the model more insight into the context of a user utterance. We present the results of this experiment in Table 1.

3 Empirical Sections

3.1 Experimental Setup

Evaluation: Data and metrics As in prior work [Mrksic and Vulic, 2018, Mrksic et al., 2016], the DST model is evaluated on the Wizard-of-OZ v.20 (WoZ2.0) dataset [Wen et al., 2016], comprising of 1,200 dialogues split into training (600 dialogues), validation (200 dialogues), and test data (400 dialogues). Each dialogue has an approximate length of 4 turns.

Additionally, the model is evaluated on the second Dialogue State Tracking Challenge (DSTC2) dataset [Henderson et al., 2014a]. The training data contains 2207 dialogues and the test set consists of 1117 dialogues. Each dialogue has an approximate length of 5.5 turns.

In all experiments, we report the standard DST performance measure: (1) joint goal accuracy, which is defined as the proportion of dialogue turns where all the users search goal constraints were correctly identified and the (2) per slot accuracy.

Training Setup We set all the NBT components as suggested by Mrksic and Vulic [2018]: the latest iteration on NBT-CNN, optimized by Adam [Kingma and Ba, 2015] with dropout [Srivastava et al., 2014] of 0.5, gradient clipping, a batch size of 256, and a training duration of 400 epochs. All model hyperparameters were tuned on the validation sets.

Model Variations We compare the previous NBT model with a variety of model variations. We modified the user utterance *representation learning* and the *semantic decoding* modules of the NBT model as intuitively we believed that this would allow for the greatest performance improvements.

Embedding Variant	Validation Set Accuracy			
	Food	Joint*	Price Range	Area
Paragram + BiLSTM	**97.1**	**95.3**	98.7	**98.3**
ELMo + BiLSTM	96.3	94.1	**98.8**	98.1

Table 1: DST performance (*per slot and joint goal accuracy*) showcasing the representation learning variants.

Semantic Decoding	Validation Set Accuracy			
	Food	Joint*	Price Range	Area
General Attention	0.0	0.0	10.8	7.4
Dot Product Attention	7.9	31.9	98.7	99.3
Scalar Dot Product Attention	**99.2**	**98.9**	**99.5**	**100**

Table 2: DST performance (*per slot and joint goal accuracy*) showcasing the semantic decoding variants. We used the best performing (*Paragram + BiLSTM*) for the representation learning module.

DataSet	Model	Test Set Accuracy			
		Food	Joint*	Price Range	Area
WOZ	NBT-CNN	-	84.8	-	-
WOZ	NBT-BiLA	90.4	**86.3**	95.1	96.3
DSTC2	NBT-CNN	80.2	68.7	**90.9**	88.2
DSTC2	NBT-BiLA	**86.1**	**73.2**	82.1	**92.6**

Table 3: DST performance (*per slot and joint goal accuracy*) comparing the performance of our final NBT-BiLA vs NBT-CNN model on the WoZ2.0 and DSTC2 datasets.

3.2 Results and Discussion

Table 1 compares the two embedding variants of the representation learning. We see that Paragram + BiLSTM performed the best. This was a surprise as we expected ELMo + BiLSTM to outperform all. We believe this is the case because for a lot of slots have a similar context, for example for area the user can say: *"I want a restaurant in the north part of town."* vs *"I want a restaurant in the south part of town"*. Both these sentences have the same context for the area, we believe ELMo has a harder time separating north/south as compared to Paragram as the similar context ends up adding noise to the model rather than helping it.

Table 2 compares the three variants of semantic decoding. Here we see that scaled attention performed the best. Other attention methods performed surprisingly poorly. We believe general attention introduced additional parameters which might have led to over-fitting to the training data. In case of dot-product attention, without the normalizing factor of $\sqrt{d_{rnn}}$, the magnitude of the matrix multiply grew too large, eventually pushing the softmax function at the end of the architecture into regions where it had extremely small gradients.

Table 3 compares our best model NBT-BiLA with the best NBT-CNN model. We can see that on both the datasets we out-performed the NBT-CNN model.

* We want to pick the model according to the best joint accuracy, as that is the most effective representation of the model's overall performance.

4 Related Work

Belief Tracker The belief trackers main purpose has been within the context of a spoken dialogue system problem. Typically, these systems function with an Automatic Speech Recognition (ASR) output being fed into a Spoken Language Understanding (SLU) decoder to detect the slot-value pairs expressed by a user [Perez and Liu, 2016a, Thomson and Young, 2010].

However, this methodology has been recently outclassed as it relies on learning independent parameters for each slot-value pair, and is not so adjustable to larger, more complex domains. Instead of determining independent slot-value pairs, models that share parameters across slots and values (using delexicalization) have outperformed the traditional models. However, these have relied on semantic, hard-coded dictionaries to do so, and are not as flexible [Mrkšić et al., 2016].

Other work has been done in an attempt to add attention to this belief tracking problem. The recent Global-Locally Self-Attentive Dialogue State Tracker (GLAD) by Zhong et al. [2018] includes an attentive encoder which shares parameters between slots and allows for slot-specific feature learning.

Task-Oriented Dialogue Ultimately, the neural belief tracker described in this paper is a component of a much larger system. Wen et al. [2016] looked to create an end-to-end trainable goal-oriented system that responds to and ascertains the objective of a user's requests through dialogue.

The belief tracker's role in the context of this larger model is to keep track of a user's goal at each step of the dialogue, while also passing on the belief state as input to the Policy Network and Database Operator which then can generate a system response.

This modularly connected approach to a task-oriented dialogue system allows for improvement across each component of the model, and by designing a more accurate and robust neural belief tracker we can improve the performance of the entire system.

5 Conclusion

This project shows that by improving the representation learning along with semantic decoding with the help of a recurrent architecture (BiLSTM) along with attention we can get a reasonable gain on the joint accuracy on the WoZ2.0 and the DSTC2 datset.

6 Acknowledgments

We would like to acknowledge Noah Smith, Elizabeth Clark, and Lucy Lin for their great mentorship throughout this capstone course. Also, Akshat Shrivastava, Dan Tran, Weihan Ji, Jack Khuu, Blarry Wang, Ravi Patel and Mitali Palekar provided great feedback on various drafts of this paper.

References

Matthew Henderson, Blaise Thomson, and Jason D. Williams. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A., June 2014a. Association for Computational Linguistics. doi: 10.3115/v1/W14-4337. URL `https://www.aclweb.org/anthology/W14-4337`.

Matthew Henderson, Blaise Thomson, and Steve Young. 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, Philadelphia, PA, U.S.A., June 2014b. Association for Computational Linguistics. doi: 10.3115/v1/W14-4340. URL `https://www.aclweb.org/anthology/W14-4340`.

Matthew Henderson, Blaise Thomson, and Steve J. Young. Robust dialog state tracking using delexicalised

recurrent neural networks and unsupervised adaptation. *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 360–365, 2014c.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Nikola Mrksic and Ivan Vulic. Fully statistical neural belief tracking. *CoRR*, abs/1805.11350, 2018. URL http://arxiv.org/abs/1805.11350.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1018. URL https://www.aclweb.org/anthology/N16-1018.

Nikola Mrksic, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve J. Young. Neural belief tracker: Data-driven dialogue state tracking. *CoRR*, abs/1606.03777, 2016. URL http://arxiv.org/abs/1606.03777.

Julien Perez and Fei Liu. Gated end-to-end memory networks. *CoRR*, abs/1610.04211, 2016a. URL http://arxiv.org/abs/1610.04211.

Julien Perez and Fei Liu. Gated end-to-end memory networks. *CoRR*, abs/1610.04211, 2016b. URL http://arxiv.org/abs/1610.04211.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15: 1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. On-line active reward learning for policy optimisation in spoken dialogue systems. *CoRR*, abs/1605.07669, 2016a. URL http://arxiv.org/abs/1605.07669.

Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. Continuously learning neural dialogue management. *CoRR*, abs/1606.02689, 2016b. URL http://arxiv.org/abs/1606.02689.

Kai Sun, Qizhe Xie, and Kai Yu. Recurrent polynomial network for dialogue state tracking. *CoRR*, abs/1507.03934, 2015. URL http://arxiv.org/abs/1507.03934.

Blaise Thomson and Steve Young. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Comput. Speech Lang.*, 24(4):562–588, October 2010. ISSN 0885-2308. doi: 10.1016/j.csl.2009.07.003. URL http://dx.doi.org/10.1016/j.csl.2009.07.003.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Miroslav Vodolán, Rudolf Kadlec, and Jan Kleindienst. Hybrid dialog state tracker with ASR features. *CoRR*, abs/1702.06336, 2017. URL http://arxiv.org/abs/1702.06336.

Zhuoran Wang and Oliver Lemon. 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, Metz, France, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W13-4067.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve J. Young. A network-based end-to-end trainable task-oriented dialogue system. *CoRR*, abs/1604.04562, 2016. URL http://arxiv.org/abs/1604.04562.

Jason D. Williams, Antoine Raux, and Matthew Henderson. The dialog state tracking challenge series: A review. *DD*, 7:4–33, 2016.

Victor Zhong, Caiming Xiong, and Richard Socher. Global-locally self-attentive dialogue state tracker. *CoRR*, abs/1805.09655, 2018. URL http://arxiv.org/abs/1805.09655.