BERT-DST: Scalable End-to-End Dialogue State Tracking with Bidirectional Encoder Representations from Transformer

(Guan-Lin Chao, Ian Lane)

presented by Alexey Korolev

16 April 2020

Content

Introduction to DST

- Core component in today's task-oriented dialogue systems, maintains user's intentional states through the course of a dialogue.
- Our task fill slot(e.g. 'movie_name') with differen target ('12 angry men') from user utterance.

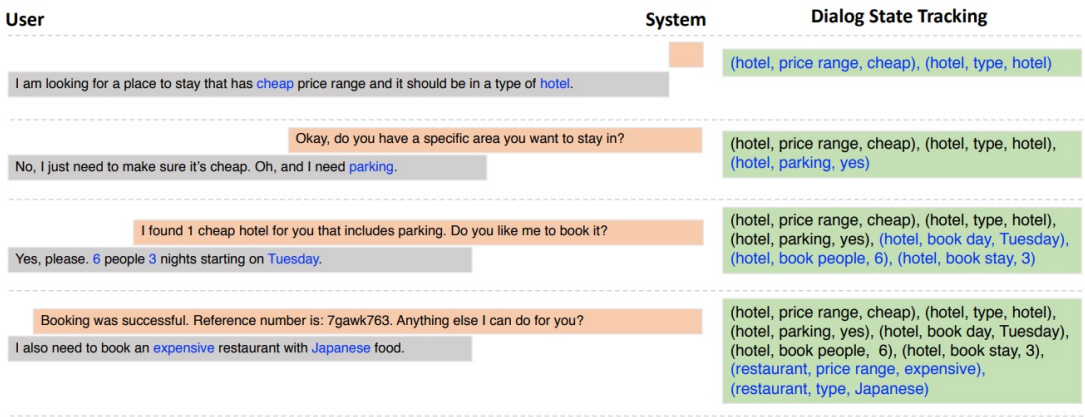


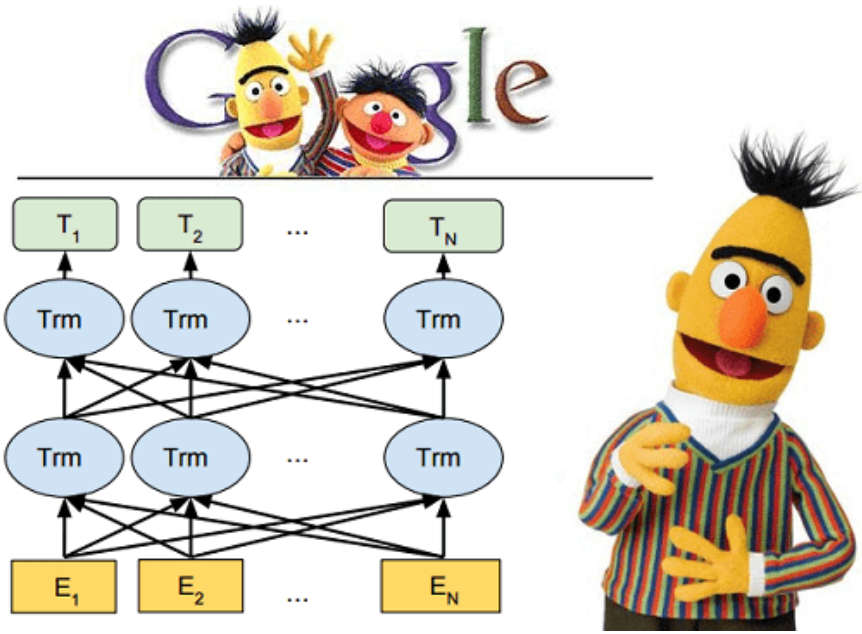
Figure 1: An example of dialog state tracking process for booking a hotel and reserving a restaurant. Each turn contains a user utterance (grey) and a system utterance (orange). The dialog state tracker (green) tracks all the $< domain, slot, value >$ triplets until the current turn. Blue color denotes the new state appeared at that turn. Best viewed in color.

Our task

- We have 3 category: none, dontcare, span. Where 'none' denotes that a domain-slot pair is not mentioned at this turn, 'dontcare' implies that the user can accept any values for this slot 'span' represents that the slot should be processed by the model with a real value.

BERT

- BERT is a multi-layer bidirectional Transformer encoder , which is a stack of multiple identical layers each containing a multi-head self-attention and a fully-connected sub-layer with residual connections.
- Tasks: masked language modeling and next sentence prediction
- BooksCorpus and the English Wikipedia corpora.



BERT-DST

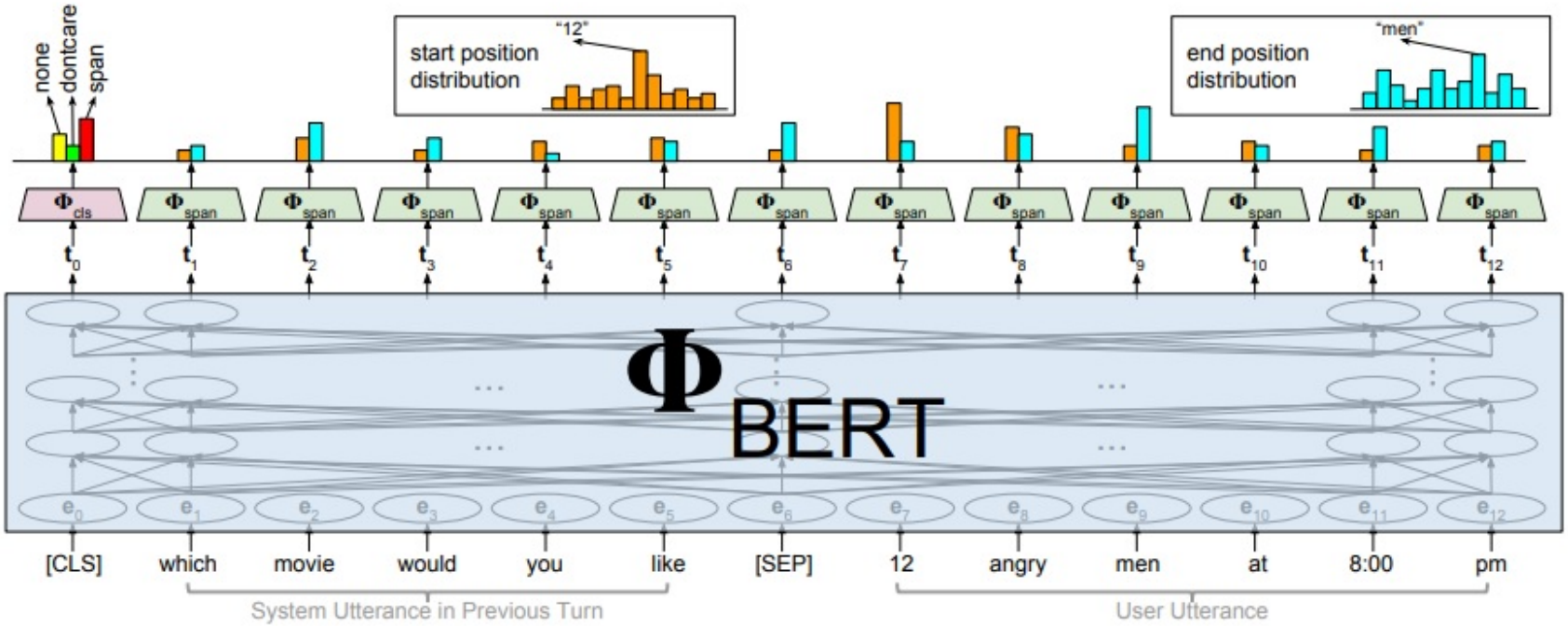


Figure 1: *Architecture of the proposed BERT-DST framework. The diagram is color-coded such that modules with the same color share the same parameters. For each user turn, BERT-DST takes as input the recent dialogue context (system utterance in previous turn and the user utterance), and outputs turn-level dialogue state. BERT dialogue context encoding module Φ_{BERT} (blue) produces contextualized sentence-level and token-level representations of the dialogue context. The per-slot classification module Φ_{cls} (red) uses the sentence-level representation to generate a categorical distribution over three types of slot values {none, dontcare and span}. The per-slot span prediction module Φ_{span} (green) gathers the token-level representations and output the start and end positions (span) of the slot value. Note that the dialogue context encoding module Φ_{BERT} allows parameter sharing across all slots.*

BERT-DST: Φ_{BERT}

$$BERTinput(x_i) = E_{tok}(x_i) + E_{seg}(i) + E_{pos}(i) = \\ = \mathbf{e_i}, \tag{1}$$

where $E_{toc}(x_i)$ — WordPiece embedding for x_i
$E_{seg}(i) \in \{e_{first}, e_{second}\}$ — segment embedding whose value is determined by whether the token belongs to the first or second sentence
$E_{pos}(i)$ — positional embedding for the i-th token.

$$\Phi_{BERT}([e_0, ...e_n]) = [t_0, ...t_n], \tag{2}$$

t_0 use for classification $[t_1, ..., t_n]$ for span prediction.

BERT-DST : Φ_{cls} *and* Φ_{span}

1. Φ_{cls}

$$a^s = W^s_{cls} t_0 + b^s_{cls} \quad (3)$$

$$p^s = softmax(a^s) = \quad (4)$$

$$= [p^s_{none}, p^s_{doncare}, p^s_{span}]$$

$$slot\_value^s = argmax(p^s_c) \quad (5)$$

2. Φ_{slot}

$$[\alpha^s, \beta^s] = W^s_{span} t_0 + b^s_{span} \quad (6)$$

$$p^s_\alpha = softmax(\alpha^s) \quad (7)$$

$$p^s_\beta = softmax(\beta^s) \quad (8)$$

$$start\_pos^s = argmax_i(p^s_\alpha) \quad (9)$$

$$end\_pos^s = argmax_i(p^s_\alpha) \quad (10)$$

Parameters sharing and dropout

- BERT-DST SS(slot specific)
  Model work only with one slot. For different slot we train different BERT-DST module.
- BERT-DST PS (parameters sharing)
  We can apply parameter sharing in the dialogue context encoding module across all slots. So we reduce nuber of model parameters.

Implementation detail

- To improve the robustness for unseen slot values, in the training phase, we replace each of the target slot value tokens by a special [UNK] token at a certain probability(30% dropout rate).
- $Loss_{total} = 0.8L^{xent}_{cls} + 0.1L^{xent}_{span\_start} + 0.1L^{xent}_{span\_end}$, where L^{exnt} denotes the cross entropy loss for the corresponding prediction target
- Update all layers in the model using ADAM optimization with an initial learning rate $2e^{-5}$ and early stopping on the validation set.

Datasets

- Sim-M, Sim-R - automatically synthesize labeled datasets in the movie and restaurant domain.
- DSTC2 and WOZ 2.0 are standard benchmarks for taskoriented dialogue systems, which are both in the restaurant domain and share the same ontology. In DSTC2, automatic speech recognition (ASR) hypotheses of user utterances are provided to assess DST models' robustness against ASR errors, so we use the top ASR hypo

Datasets	# Dialogues (train, dev, test)	Slots
Sim-M	384, 120, 264	date, time, num_tickets, theatre_name, **movie** (5/5; 26/26)
Sim-R	1116, 349, 775	date, time, category, price_range, rating, num_people, location, meal, **restaurant_name** (5/19; 9/23)
DSTC2	1612, 506, 1117	area, price range, **food** (1/73; 0/74)
WOZ 2.0	600, 200, 400	**area** (0/6; 1/7), price range, **food** (1/65; 2/72)

Table 3: *Dataset statistics. The number of dialogues is given for train, dev and test sets respectively. The slots containing OOV values are marked in bold. Parentheses represent (# unique OOV values in dev set / # unique values in dev set; # unique OOV values in test set / # unique values in test set).*

Results

- BERT-DST_PS gives less accuracu than BERT-DST_SS.
- Model doesn't give close to SOTA results and it's scalable.
- Slot value improves models.

DST Models	Sim-M	Sim-R
DST + LU Candidates [7]	50.4%	87.1%
DST + Oracle Candidates† [5]	96.8%	94.4%
BERT-DST_SS	71.6%	87.4%
+ slot value dropout	76.3%*	87.6%
BERT-DST_PS	72.3%	88.6%*
+ slot value dropout	**80.1%***	**89.6%***

Table 1: *Comparison with prior approaches on Sim-M and Sim-R datasets (joint goal accuracy). * indicates statistically significant improvement over BERT-DST model (paired sample t-test; $p < 0.01$). † indicates the corresponding model should be considered as a kind of oracle because the candidates are ground truth slot-tagging labels, i.e. the targeted slot value is guaranteed to be in the candidate list and considered by DST.*

DST Models	DSTC2	WOZ 2.0
DST + LU Candidates [7]	67.0%	-
DST + n-gram Candidates [8]	68.2±1.8%	-
DST + Oracle Candidates [5]	70.3%	-
Pointer Network [6]	72.1%	-
Delex.-Based Model [2]	69.1%	70.8%
Delex. + Semantic Dict. [2]	72.9%	83.7%
Neural Belief Tracker [2]	73.4%	84.2%
GLAD [3]	74.5±0.2%	88.1±0.4%
StateNet [4]	**75.5%**	**88.9%**
BERT-DST_PS	69.3±0.4%	87.7±1.1%

Table 2: *Comparison with prior approaches on DSTC2 and WOZ 2.0 datasets (joint goal accuracy). We report the average and standard deviation of test set accuracy of 5 model runs with random training data shuffling and normal initialization on classification and span prediction weights.*

Thank for attentions!

https://github.com/guanlinchao/bert-dst