Hybrid Dialogue State Tracking for Real World Human-to-Human Dialogues

Kai Sun, Su Zhu, Lu Chen, Siqiu Yao, Xueyang Wu, Kai Yu

Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
ks985@cornell.edu, {paul2204,chenlusz,neroysq,wuxueyang,kai.yu}@sjtu.edu.cn

Abstract

Dialogue state tracking is a key sub-task of dialogue management. The fourth Dialog State Tracking Challenge (DSTC-4) focuses on dialogue state tracking for real world human-to-human dialogues. The task is more challenging than previous challenges because of more complex domain and coreferences, more synonyms and abbreviations, sub-dialogue level labelled utterances, and no spoken language understanding output provided. To deal with these challenges, this paper proposes a novel hybrid dialogue state tracking method, which can take advantage of the strength of both rule-based and statistical methods. Thousands of rules are first automatically generated using a template-based rule generation approach and then combined together with several manually designed rules to yield the output of the rule-based method. In parallel, a statistical method is applied to track the state. The tracker finally takes the union of the outputs of the two methods. In DSTC-4 evaluation, the proposed hybrid tracker obtained state-of-the-art results. It ranked the second and significantly outperformed the baseline system and most submissions.

Index Terms: dialogue state tracking (DST), statistical dialogue management, hybrid tracker

1. Introduction

Dialogue management plays an important role in dialogue systems. It has two aims: to track the dialogue state (DST) based on the current input and the conversation history, and choose a response according to its dialogue policy. The current state-of-the-art for statistical dialogue management is to use the partially observable Markov decision process (POMDP) to track the dialogue state and determine the appropriate system response [1]. In early works of POMDP, belief state is updated using Bayes theorem with consideration of reasonable Markov independence assumptions. Recently, to advance the DST research, the dialog state tracking challenges (DSTC-1/2/3) are organized to provide common testbeds for evaluation of different DST models [2, 3, 4]. A lot of DST methods have been proposed, including maximum entropy model (MaxEnt) [5, 6], conditional random field (CRF) [7], deep neural networks (DNN) [8], recurrent neural networks (RNN) [9], decision forest [10], rule-based models [11, 12] and hybrid models [13, 14, 15].

The fourth Dialog State Tracking Challenge (DSTC-4) focuses on human-human dialogues between tourists and guides [16]. Each dialogue session is segmented into a series of sub-dialogs . A sub-dialog consists of successive multi-turns which are related to a same topic, e.g. *accommodation, food*. The dialogue state is defined as a set of *slot-value* pairs consistent with the topic of sub-dialogue. However, compared with the previous DSTCs, some features make DSTC-4 more challenging [17]. First, the domain is more complex, i.e. the ontology includes more slots-value pairs, and some of values are shared across different slots. Second, there are lots of coreferences in dialogues, e.g. “the hotel”, “the bar”, which denote the slot-value pairs mentioned by tourist/guide in previous conversation. Third, the human often uses numerous synonyms and abbreviations to refer to the values in the ontology. Fourth, the dialogue states are defined on the sub-dialogue level. For each sub-dialogue, the label of states for each utterance is the same. Besides, the result of semantic parser is not provided. All of above render the performance degradation of pure machine-learning methods.

This paper describes our hybrid tracker for DSTC-4, which combines a rule-based method and a statistical method. The output of the hybrid tracker is the union of the predictions of two different models. The rule-based method is a combination of thousands of simple rules, most of which are automatically generated using the general rule templates. The statistical model is based on support vector machines (SVMs). In this method, a binary classifier is trained for each *slot-value* pair at the sub-dialogue level, i.e. the input features of the model are extracted from all sentences in a sub-dialogue. While in testing, the input features of the model are extracted from sentences up to the current turn in the sub-dialogue.

The rest of the paper is organized as follows. Section 2 presents the rule-based method in detail. Section 3 describes the statistical model. Section 4 presents the alias detection, followed by experiments in section 5. Finally, section 6 concludes the paper.

2. Rule-based Method

A rule-based method is applied, which basically is a combination of thousands of simple rules. Here, both manually-designed and automatically-generated rules are used. The number of manually-designed rules is small and most of them are designed for handling special cases, while most rules used in our system are automatically generated and designed for handling general cases.

This work was done in Shanghai Jiao Tong University and was supported by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC project No. 61573241 and the Interdisciplinary Program (14JCZ03) of Shanghai Jiao Tong University in China . Kai Sun currently is a Ph.D student in Cornell University, USA.

2.1. Manually Designed Rules

The manually designed rules consist of 3 types: fuzzy matching rules, rules dealing with inter-segment dependency, and hard-coded rules.

2.1.1. Fuzzy Matching

Like the organizer-provided baseline [16], a rule based on fuzzy matching is used. Basically, this rule determines the slot values by fuzzy string matching between the possible values specified in the ontology and the transcription of current utterance. If at least one part of the utterance is matched with a possible value in the ontology with over a threshold of similarity, the value of the slot is then set to be the possible value with the highest similarity to the part of the utterance.

2.1.2. Inter-segment Dependency

Since it can be observed from the training data that there are many inter-segment dependencies, a rule is designed to process the transcript to replace a type of word reference with the specific words. In details, for every location type specified in the ontology such as "hotel" and "bar", the last seen value belonging to that type in the dialogue state is maintained in an array, say, $location[\cdot]$. When a new turn begins, all appearances of the location reference in the transcript including (a) "the [$location\_type$]" (b) "this [$location\_type$]" (c) "that [$location\_type$]" (d) "these [$location\_type$]s" (e) "those [$location\_type$]s" are replaced by $location[location\_type]$.

2.1.3. Hard-coded Rules

Several hard-coded rules are used and they are mainly designed for dealing with the values that are relatively abstract and/or require considering some semantic aspects of the dialogue. For example, several rules about the slot "INFO" are used.

2.2. Automatically Generated Rules

The automatically-generated rules are first generated by several different rule templates, and then selected by evaluating their performance on the training set – only those with high state tracking precision on the training set are kept.

2.2.1. Rule Templates

A key observation of the training data is that the value of a given topic t and slot s can be determined by some relatively simple rules. For example, let i, w_i denote the index and the transcript of current utterance in current sub-dialogue respectively, then if $t =$"Food", $s =$"CUISINE", and both "Singapore" and "signature" are substrings of w_i, "Singapore cuisine" should be the value of the slot with high probability. Below are two representative instances:

- "`%Um I think any- %uh we really want to try out like maybe the signature %um dishes in Singapore.`" (utterance 98 of session 1)
- "`If- is that the signature dish of Singapore?`" (utterance 104 of session 1)

Concretely, the rule can be described by Algorithm 1. Another key observation is despite the fact that tens of thousands of these kinds of simple rules exist, most of them can be summarized by several rule templates, which are described in detail in the appendix and Figure 1. For our example here, Algorithm 1 is a specific instance of the rule template shown in Figure 1c.

Algorithm 1: An Example of Rule 3

```
if $t =$ "Food" $\wedge$ $s =$ "CUISINE" then
    if "Singapore" in $w_i$ $\wedge$ "signature" in $w_i$ then
        $state$["Cuisine"] $\leftarrow$ "Singapore cuisine"
```

2.2.2. Rule Generation

All possible rules are generated by enumerating all possible combinations of parameters of the rule templates, and those that are *non-trivial* and have *high* state tracking precision on the training data are kept. In practice, whether a rule is *non-trivial* is mainly evaluated by the following aspects:

- If the template of the rule has the parameter **w**, whether **w** is a stop word[1].
- If the template of the rule has the parameter **v** and **w**, whether **w** is similar to **v** by fuzzy matching.
- The absolute number of correct answers that can be given by the rule.
- The relative number of correct answers that can be given by the rule, compared to the other rules generated by the same template.

2.3. Rule Combination

Algorithm 2 describes the procedure of combining the manually designed rules and automatically generated rules.

Algorithm 2: Rule Combination

```
Process the transcript of the current utterance with the rule of
inter-segment dependency and use the processed transcript in all
following steps
Apply hard-coded rules as well as automatically generated
rules and combine their results with voting. Let $vote[s][v]$
denote the weighted[2]number of votes received by value $v$ for
slot $s$
foreach slot $s$ of current topic do
    if $max_v\{vote[s][v]\} \geq threshold$ then
        $state[s] \leftarrow argmax_v\{vote[s][v]\}$
    else
        Apply the rule of fuzzy matching to slot $s$
    end
end
```

Algorithm 2 is our final rule-based model, and it is run every time a new utterance comes (i.e. a new turn comes) to update the dialogue state.

3. Statistical Method

A statistical discriminative method based on SVMs is also applied. In the main task of DSTC-4, since the *slot-value* pairs are only annotated at sub-dialogue level rather than at turn level, it is hard to learn a model turn by turn, so the proposed model is trained at sub-dialogue level. Specifically, each sub-dialogue segment (all sentences ordered turn by turn in the segment) is

[1] The stop word list is generated by picking the words occurred frequently and deemed unhelpful for state tracking in the training data.

[2] Different types of rules are manually set to have different weights intuitively based on the number of parameters and rules of the type, and the number of instances that can be observed from the training data. Little effort is spent on fine tuning the weights in our system submitted to DSTC-4.

```
Template rule_1 ($\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{w}$)
    if $t = \boldsymbol{t} \wedge s = \boldsymbol{s} \wedge \boldsymbol{w}$ in $w_i$ then
        $v \leftarrow lookup(t, s, w_i)$
        if $v \neq NULL$ then
            $state[s] \leftarrow v$
```

(a) Rule Template 1

```
Template rule_2 ($\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{p}$)
    if $t = \boldsymbol{t} \wedge s = \boldsymbol{s}$ then
        $v \leftarrow lookup(t, s, w_i, \boldsymbol{p})$
        if $v \neq NULL$ then
            $state[s] \leftarrow v$
```

(b) Rule Template 2

```
Template rule_3 ($\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{v}$)
    if $t = \boldsymbol{t} \wedge s = \boldsymbol{s}$ then
        if $\boldsymbol{w}_1$ in $w_i \wedge \boldsymbol{w}_2$ in $w_i$ then
            $state[s] \leftarrow \boldsymbol{v}$
```

(c) Rule Template 3

```
Template rule_4 ($\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{v}$)
    if $t = \boldsymbol{t} \wedge s = \boldsymbol{s}$ then
        if $\boldsymbol{w}_1$ in $w_i \wedge \boldsymbol{w}_2$ in $w_i \wedge \boldsymbol{w}_3$ in $w_{1 \dots i-1}$ then
            $state[s] \leftarrow \boldsymbol{v}$
```

(d) Rule Template 4

```
Template rule_5 ($\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{w}$)
    if $t = \boldsymbol{t} \wedge s = \boldsymbol{s} \wedge \boldsymbol{w}$ in $w_i$ then
        $state[s] \leftarrow \tilde{state}[t][s]$
```

(e) Rule Template 5

```
Template rule_6 ($\boldsymbol{t}, \boldsymbol{s}, \boldsymbol{a}_g, \boldsymbol{a}_t$)
    if $t = \boldsymbol{t} \wedge s = \boldsymbol{s}$ then
        if $a_g = \boldsymbol{a}_g \wedge a_t = \boldsymbol{a}_t$ then
            $state[s] \leftarrow \tilde{state}[t][s]$
```

(f) Rule Template 6

Figure 1: 6 Representative Rule Templates

regarded as a long utterance, and its corresponding frame label containing a set of *slot-value* pairs is used as the annotation. Following the *semantic tuple classifiers* approach [18], a binary classifier is trained for each *slot-value* pair in a topic, and also predicts the presence of that slot-value pair in the long utterance.

3.1. Feature Representation

The features used in this method consist of n-gram features of the utterance [18], the similarities between the utterance and the *slot value* based on TF-IDF (term frequency - inverse document frequency) of them and the indication of whether a word or n-gram exists in them. Specially, for *slot* "FROM", features from subsequences of the utterance that is between "from" and "to" or "goto" are caught. Similarly, for *slot* "TO", features from subsequences after "to" or "goto" but before "from" are caught.

n-gram Features

Extracting n-grams from the long utterance provides the feature representations needed for input into the SVM. It indicates whether the n-gram occurs in the long utterance ($n = 1, 2$). Specially, n-grams crossing two sentences are ignored, i.e. 2-gram cannot start with a question mark, exclamatory mark or full stop.

Similarity Features

The similarities between the utterance and text description of the *slot-value* pair are also exploited. Commonly the text description of one *slot-value* pair is just the *value*, except for *slot* being "FROM" or "TO" where the description is pre-designed manually. For instance, the description of ("FROM", "AMK Hub") pair is "from amk hub".

Three similarity features are designed to represent the relativity between the utterance and the *slot-value* description. The first one is based on 1-gram and its TF-IDF value. The IDF is calculated for each word in the descriptions of all *slot-value* pairs in a topic. Each description is considered as a document for the purpose of computing IDF measure.

Two word tables of the utterance u and the description d can be obtained, denoted as T^u, T^d respectively. For a table T^x, it consists of a set of word W^x and the value V_w^x of each word w, $w \in W^x, x \in \{u, d\}$. The V_w^x is

$$V_w^x = TF_w^x * IDF_w \tag{1}$$

where TF_w^x is the frequency of word w in x, IDF_w is the IDF value of word w, $w \in W^x, x \in \{u, d\}$. If word w only appears in utterances but not descriptions, $V_w^x = 0$.

Then the first similarity value S_1 can be obtained:

$$S_1 = \begin{cases} \frac{\sum_{w \in W^d} V_w^u V_w^d}{L_d L_d} & \text{if } L_u \leq L_d \\ \frac{\sum_{w \in W^d} V_w^u V_w^d}{L_u L_d} & \text{otherwise} \end{cases} \tag{2}$$

where $L_u = \sqrt{\sum_{w \in W^d} {V_w^u}^2}$, $L_d = \sqrt{\sum_{w \in W^d} {V_w^d}^2}$, which are the 2-norm. The first condition "if $L_u \leq L_d$" in this function is used to restrain the similarity score rigidly when less w^d appear in the utterance.

The other two similarity features are about the indication of 1-gram and n-gram ($n = 1, 2$) in u and d, i.e. setting IDF_w to 1.

History Feature

Additionally, the influence of previous *slot-value* pairs that appear in the dialogue history is modelled. Simply, a list of (*slot*, *value*) is built to keep the dialogue history. It saves all *slot-value* pairs of the previous dialogue turns, and keeps the *value* to be the newest one. The history feature is an indication whether the *slot-value* candidate has appeared in the memory list.

3.2. Training and Decoding

As described previously, a binary classifier for each *slot-value* pair is trained by feeding these features. LibSVM [19] which can output probabilities is used to train the SVM classifiers with linear kernel. In testing, only sentences from the segment beginning to the current turn is considered as a long utterance. Then all binary classifiers are run to get the existence of *slot-value* pairs[3]. Finally, all of the *slot-value* pairs that exist from the segment beginning to the current turn are collected as the current dialogue states.

4. Alias Detection

People often mention a place, region or food by its simplified name. For instance, "Rendezvous Grand Hotel Singapore", alias "Rendezvous Hotel". So intuitively detecting aliases of each value based on domain knowledge as well as training data can help better extract *slot-value* pairs.

A value may have one or several aliases. To detect aliases, for each value v, first, a candidate set is built, which includes the text chunk of every utterance with v labelled in the training data with minimum edit distance to v, and subsequences of v of length> 1. Next, every candidate that belongs to more than one

[3] The acceptance threshold of the existing probability is set to 0.75 according to grid searching results in the development dataset

candidate set is rejected. Finally, the candidates whose detection precision of *slot-value* pair is less than 0.8 in the training data (if the *slot-value* pair appears in the annotation) is rejected.

5. Experiments

5.1. The DSTC-4 dataset

The dataset of DSTC-4 for main task consists of 29 dialogue sessions with 25419 utterances [16]. These dialogues are divided into three parts: *dstc4-train, dstc4-dev* and *dstc4-test*. The first two datasets are used for the development of tracker, and the last one is used for evaluation. In DSTC-4, four evaluation metrics are used: segment accuracy (SAcc), precision, recall and F-score. SAcc is the fraction of segments in which the trackers output is equivalent to the label of state, while the other three metrics are used to evaluate the performance of tracker on *slot-value* level. For all metrics, a higher value is better.

5.2. Rule-based Method

Rules	Prec.	Recall	F-score	SAcc
Fuzzy Matching Only	47.28	18.48	26.57	4.75
Manually Designed Rules	50.79	22.86	31.53	6.49
+ Auto. Generated Rules	61.83	42.56	50.42	16.93

Table 1: *Performances of rule-based models with different rules on the dstc4-dev on schedule 2 [16].*

It can be observed from Table 1 that compared with only using fuzzy matching, on the one hand, adding more manually designed rules can help improve the performance modestly, on the other hand, the automatically generated rules help improve the performance greatly. That well demonstrates that compared to the manually designed rules that apply to relatively special cases, the rules generated by the proposed template-based rule generation approach are more general and effective.

5.3. Statistical Method

Features	Prec.	Recall	F-score	SAcc
n-gram	80.44	18.80	30.48	5.85
+ Similarities	79.27	20.93	33.11	6.17
++ History	78.21	21.96	34.29	6.49

Table 2: *Performances of statistical models with different features on the dstc4-dev on schedule 2.*

As illustrated in Table 2, both similarity features and history features can help improve the system on the metrics of F-score and SAcc. However, in general the single system of the proposed statistical method shows poor performance in contrast to the rule-based method except for the much higher precision of *slot-value* pair detection. Therefore, it is promising to combine it with the rule-based method.

5.4. Model Combination

System	Alias Detection	F-score	SAcc
Rule	No	50.42	16.93
Rule	Yes	52.30	17.88
Rule + Stat. (+Similarities)	Yes	55.60	17.88
Rule + Stat. (++History)	Yes	55.61	17.88

Table 3: *Performances of model combinations on the dstc4-dev on schedule 2.*

As illustrated in Table 3, both statistical model and alias detection can help improve the performance by output union. In particular, alias-detection contributes to improving the F-score and SAcc, while the statistical model only helps improve F-score. It can be seen that the history feature does not contribute for the combined model. Hence, only n-gram and similarity features are fed into the statistical model in our final system. This *combined* system gets a significant improvement compared to the baseline in DSTC-4 as shown in Table 4. It is ranked the 2nd among 7 research teams participating in the main task of DSTC-4[4].

System	Precision	Recall	F-score	SAcc
Baseline	37.50	25.19	30.14	4.88
Combined	56.66	44.55	49.88	12.64

Table 4: *One of our submitted entries in DSTC-4. These metrics are evaluated on the dstc4-test on schedule 2. The models are trained on dstc4-train and dstc4-dev.*

6. Conclusion

This paper describes a hybrid tracker, which combines a rule-based method and a statistical method. Experiments demonstrate the proposed template-based rule generation approach is effective for bootstrapping using domain knowledge, and show the effectiveness of model combination. The results show that the proposed hybrid model is competitive and outperforms most of the other systems in DSTC-4 on test datasets.

7. Appendix

Given the topic t, slot s, guide act a_g, tourist act a_t, the index i of current utterance in current sub-dialogue, and the utterances $w_1, w_2, \ldots, w_i$ in current sub-dialogue, every rule generated by rule templates follows similar ways to decide whether to update the value or not. For every template, the variables in bold are parameters of templates. Specifically, $\boldsymbol{t}$, $\boldsymbol{s}$, $\boldsymbol{v}$, $\boldsymbol{a_g}$, $\boldsymbol{a_t}$ represent a possible topic, slot, value, guide act, tourist act respectively whose domain is specified in the ontology; $\boldsymbol{w}$ represents a word or a phrase; $\boldsymbol{p}$ represents a word.

In some templates, $lookup(t, s, w_i)$ is called, whose function is to look up the ontology to see whether there exists any value of the topic t, slot s such that the value is a substring of w_i, return the value if the answer is "yes", otherwise return $NULL$. In the template 3, $lookup(t, s, w_i, \boldsymbol{p})$ is called, whose function is the same as $lookup(t, s, w_i)$ except that it checks whether the concatenation of the prefix $\boldsymbol{p}$ and the value is a substring of w_i. In addition, in some templates $\tilde{state}[t][s]$ is referred, which is the "accumulated state" from the first sub-dialogue to the previous sub-dialogue, maintained by Algorithm 3.

Algorithm 3: Maintenance of $state[s]$ and $\tilde{state}[t][s]$

```
if A new sub-dialogue begins then
    Let $t$ be the topic of last sub-dialogue
    foreach slot $s$ of topic $t$ do
        if $state[s] \neq NULL$ then
            $\tilde{state}[t][s] \leftarrow state[s]$
            $state[s] \leftarrow NULL$
    end
```

[4] The SAcc of the systems ranked the 1st and 3rd is 15.00 and 7.06 respectively. A detailed comparison can be found in [16].

8. References

[1] S. Young, M. Gasic, B. Thomson, and J. D. Williams, "POMDP-based statistical spoken dialog systems: A review," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.

[2] J. Williams, A. Raux, D. Ramachandran, and A. Black, "The dialog state tracking challenge," in *Proceedings of the SIGDIAL 2013 Conference*. Metz, France: Association for Computational Linguistics, August 2013, pp. 404–413. [Online]. Available: http://www.aclweb.org/anthology/W/W13/W13-4065

[3] M. Henderson, B. Thomson, and J. 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)*. Philadelphia, PA, U.S.A.: Association for Computational Linguistics, June 2014, pp. 263–272. [Online]. Available: http://www.aclweb.org/anthology/W14-4337

[4] ——, "The third dialog state tracking challenge," in *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, December 2014, pp. 324–329.

[5] S. Lee and M. Eskenazi, "Recipe for building robust spoken dialog state trackers: Dialog state tracking challenge system description," in *Proceedings of the SIGDIAL 2013 Conference*. Metz, France: Association for Computational Linguistics, August 2013, pp. 414–422. [Online]. Available: http://www.aclweb.org/anthology/W/W13/W13-4066

[6] K. Sun, L. Chen, S. Zhu, and K. Yu, "The SJTU system for dialog state tracking challenge 2," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. Philadelphia, PA, U.S.A.: Association for Computational Linguistics, June 2014, pp. 318–326. [Online]. Available: http://www.aclweb.org/anthology/W14-4343

[7] H. Ren, W. Xu, Y. Zhang, and Y. Yan, "Dialog state tracking using conditional random fields," in *Proceedings of the SIGDIAL Conference, Metz, France*, August 2013.

[8] M. Henderson, B. Thomson, and S. Young, "Deep neural network approach for the dialog state tracking challenge," in *Proceedings of the SIGDIAL 2013 Conference*. Metz, France: Association for Computational Linguistics, August 2013, pp. 467–471. [Online]. Available: http://www.aclweb.org/anthology/W/W13/W13-4073

[9] ——, "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)*. Philadelphia, PA, U.S.A.: Association for Computational Linguistics, June 2014, pp. 292–299. [Online]. Available: http://www.aclweb.org/anthology/W14-4340

[10] J. D. Williams, "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)*. Philadelphia, PA, U.S.A.: Association for Computational Linguistics, June 2014, pp. 282–291. [Online]. Available: http://www.aclweb.org/anthology/W14-4339

[11] Z. Wang and O. 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*. Metz, France: Association for Computational Linguistics, August 2013, pp. 423–432. [Online]. Available: http://www.aclweb.org/anthology/W/W13/W13-4067

[12] K. Sun, L. Chen, S. Zhu, and K. Yu, "A generalized rule based tracker for dialogue state tracking," in *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, December 2014.

[13] K. Yu, K. Sun, L. Chen, and S. Zhu, "Constrained markov bayesian polynomial for efficient dialogue state tracking," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2177–2188, December 2015.

[14] K. Sun, Q. Xie, and K. Yu, "Recurrent polynomial network for dialogue state tracking," *submitted to Dialogue and Discourse*, 2015.

[15] M. Vodolán, R. Kadlec, and J. Kleindienst, "Hybrid dialog state tracker," in *Proceedings of NIPS 2015 Workshop on Machine Learning for Spoken Language Understanding and Interaction*, La Jolla, CA, 2015.

[16] S. Kim, L. F. D'Haro, R. E. Banchs, J. Williams, and M. Henderson, "The Fourth Dialog State Tracking Challenge," in *Proceedings of the 7th International Workshop on Spoken Dialogue Systems (IWSDS)*, 2016.

[17] F. Dernoncourt, J. Y. Lee, T. H. Bui, and H. H. Bui, "Robust dialog state tracking for large ontologies," in *Proceedings of International Workshop on Spoken Dialog Systems*, Finland, 2016.

[18] M. Henderson, M. Gasic, B. Thomson, P. Tsiakoulis, K. Yu, and S. Young, "Discriminative spoken language understanding using word confusion networks," in *SLT*, 2012, pp. 176–181.

[19] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.