Recurrent Polynomial Network for Dialogue State Tracking with Mismatched Semantic Parsers

Qizhe Xie, Kai Sun, Su Zhu, Lu Chen and Kai Yu
Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Eng.
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{cheezer,accreator,paul2204,chenlusz,kai.yu}@sjtu.edu.cn

Abstract

Recently, constrained Markov Bayesian polynomial (CMBP) has been proposed as a data-driven rule-based model for dialog state tracking (DST). CMBP is an approach to bridge rule-based models and statistical models. Recurrent Polynomial Network (RPN) is a recent statistical framework taking advantages of rule-based models and can achieve state-of-the-art performance on the data corpora of DSTC-3, outperforming all submitted trackers in DSTC-3 including RNN. It is widely acknowledged that SLU's reliability influences tracker's performance greatly, especially in cases where the training SLU is poorly matched to the testing SLU. In this paper, this effect is analyzed in detail for RPN. Experiments show that RPN's tracking result is consistently the best compared to rule-based and statistical models investigated on different SLUs including mismatched ones and demonstrate RPN's is very robust to mismatched semantic parsers.

1 Introduction

Dialogue management is the core of a spoken dialogue system. As a dialogue progresses, dialogue management usually accomplishes two missions. One mission is called dialogue state tracking (DST), which is a process to estimate the distribution of the dialogue states. Another mission is to choose semantics-level machine dialogue acts to direct the dialogue given the information of the dialogue state, referred to as dialogue decision making. Due to unpredictable user behaviours, inevitable automatic speech recognition (ASR) and spoken language understanding (SLU) errors, dialogue state tracking and decision making are difficult (Williams and Young, 2007). Consequently, much research has been devoted to statistical dialogue management. In previous studies, dialogue state tracking and decision making are usually investigated together. In recent years, to advance the research of statistical dialogue management, the DST problem is raised out of the statistical dialogue management framework so that a bunch of models can be investigated for DST. Moreover, shared research tasks like the Dialog State Tracking Challenge (DSTC) (Williams et al., 2013; Henderson et al., 2014a; Henderson et al., 2014b) have provided a common testbed and evaluation suite to facilitate direct comparisons among DST models.

Two DST model categories are broadly known, i.e, rule-based models and statistical models. Recent studies on constrained Markov Bayesian polynomial (CMBP) framework took the first step towards bridging the gap between rule-based and statistical approaches for DST (Sun et al., 2014a; Yu et al., 2015). CMBP formulates rule-based DST in a general way and allows data-driven rules to be generated, so the performance can be improved when training data is available. This enables CMBP to achieve competitive performance to the state-of-the-art statistical approaches, while at the same time keeping most of the advantages of rule-based models. Nevertheless, adding features to CMBP is not as easy as in most other statistical approaches because additional prior knowledge is needed to be added to keep the search space tractable (Sun et al., 2014a; Yu et al., 2015). For the same reason, increasing the model complexity is difficult. To tackle the weakness of CMBP, recurrent polynomial network (RPN) (Sun et al., 2015) is proposed to further bridge the gap between rule-based and statistical approaches for DST (Sun et al., 2015). RPN's unique structure enables the framework to have all the advantages of CMBP. Additionally, RPN achieves more properties of statistical approaches than CMBP. RPN uses gradient descent where CMBP uses Hill-climbing. Hence RPN can train its parameters faster and the parameter space are not limited to grid where parameters only takes values which are a multiple of a constant.

SLU is usually the input module of tracker. Hence its performance affect state tracking's performance greatly. However, it is hard to design a reliable parser because of ASR errors and the difficulty of obtaining in-domain data. Further, it is a common case that SLU on a tracker's training data is very different from SLU on a tracker's testing data in real world end-to-end dialogue system. Thus, RPN is evaluated on SLUs with great variance and especially in the case where SLU for training mismatches SLU for testing. RPN shows consistently best results among trackers investigated on all SLUs.

The contribution of this paper is to investigate more complex RPN structures with deeper layers, multiple activation nodes and more features and to evaluate RPN's performance in mismatched SLU condition.

The rest of the paper is organized as follows. Section 2 introduces rule-based models and statistical models used in DST. Section 3 introduces two frameworks – CMBP and RPN bridging rule-based models and statistical models. Complex RPN structures are also introduced in this section. Section 4 discusses the influence of SLU on tracking and the SLU mismatch condition. Section 5 evaluates RPN with different structures and features and these results are compared with state-of-the-art trackers in DSTC-3. Rule-based models, statistical models and mixed models' performance in cases where testing parser mismatches training parser are also compared. Finally, section 6 concludes the paper.

2 Rule-based and Statistical Models for DST

The results of the DSTCs demonstrated the power of statistical approaches, such as Maximum Entropy (MaxEnt) (Lee and Eskenazi, 2013), Conditional Random Field (Lee, 2013), Deep Neural Network (DNN) (Sun et al., 2014b), and Recurrent Neural Network (RNN) (Henderson et al., 2014d). However, statistical approaches have some disadvantages. For example, statistical approaches sometimes show large variation in performance and poor generalisation ability because of lack of data (Williams, 2012). Moreover, statistical models usually have a complex model structure and complex features, and thus can hardly achieve portability and interpretability.

In addition to statistical approaches, rule-based approaches have also been investigated in DSTC due to their efficiency, portability and interpretability and some of them showed good performance and generalisation ability in DSTC (Zilka et al., 2013; Wang and Lemon, 2013).

However, the performance of rule-based models is usually not competitive to the best statistical approaches. Furthermore, a general way is lacking to design rule-based models with prior knowledge and their performance can hardly be improved when training data is available.

3 Bridging Rule-based models and statistical models

There are two ways of bridging rule-based approaches and statistical approaches. One starts from rule-based models and uses data-driven approaches to find a good rule, while the other one is a statistical model taking advantage of prior knowledge and constraints.

3.1 Constrained Markov Bayesian Polynomial

Constrained Markov Bayesian Polynomial (CMBP) (Sun et al., 2014a; Yu et al., 2015) takes the first way of bridging rule-based models and statistical models.

Several probability features extracted from SLU results shown below are used in CMBP for each slot (Sun et al., 2014a; Yu et al., 2015):

- $P_t^+(v)$: sum of scores of SLU hypotheses informing or affirming value v at turn t
- $P_t^-(v)$: sum of scores of SLU hypotheses denying or negating value v at turn t
- $\tilde{P}_t^+(v) = \sum_{v' \notin \{v, \texttt{None}\}} P_t^+(v')$
- $\tilde{P}_t^-(v) = \sum_{v' \notin \{v, \texttt{None}\}} P_t^-(v')$
- $b_t(v)$: belief of "the value being v at turn t"
- b_t^r: probability of the value being $None$ (the value not mentioned) at turn t.

Because slots and values are assumed independent in CMBP. To simplify the notation, these features are denoted as $P_t^+, P_t^-, \tilde{P}_t^+, \tilde{P}_t^-, b_t^r, b_t$ in the rest of this paper.

With these probability features , a CMBP model is defined by

$$b_t = \mathcal{P}\left(P_t^+, P_t^-, \tilde{P}_t^+, \tilde{P}_t^-, b_{t-1}^r, b_{t-1}\right) \quad \texttt{s.t. constraints} \tag{1}$$

where the $\mathcal{P}$ is a multivariate polynomial function defined as

$$\mathcal{P}(x_1, \cdots, x_D) = \sum_{0 \le k_1 \le \cdots \le k_n \le D} g_{k_1,\cdots,k_n} \prod_{1 \le i \le n} x_{k_i} \tag{2}$$

where k_i is an index into input variables. n called order of the CMBP is the order of the polynomial, D denotes the number of inputs with $x_0 = 1$ and g is the parameter of CMBP.

In CMBP, prior knowledge or intuition is encoded by *constraints* in equation (1). For example, intuition that goal belief should be unchanged or positively correlated with the positive scores from SLU can be written to a constraint:

$$\frac{\partial \mathcal{P}(P_{t+1}^+, P_{t+1}^-, \tilde{P}_{t+1}^+, \tilde{P}_{t+1}^-, b_t^r, b_t)}{\partial P_{t+1}^+} \ge 0 \tag{3}$$

Further, these constraints are approximated to linear forms (Sun et al., 2014a; Yu et al., 2015).

With a set of linear constraints, integer linear programming can be used to get the integer parameters which satisfy the relaxed constraints. Then the tracking accuracy of each parameters can be evaluated and the best one is picked out. Hill-climbing can further be used to extend the best integer-coefficient CMBP to real-coefficient CMBP (Yu et al., 2015).

Note that in practice order 3 (n=3) is used to balance the performance and the complexity (Sun et al., 2014a; Yu et al., 2015). 3-order CMBP has achieved state-of-the art-performance on DSTC-2/3.

3.2 Recurrent Polynomial Network

Recurrent Polynomial network (Sun et al., 2015) takes the second way to bridge rule-based and statistical models. It is a computational network and a statistical framework, which takes advantage of prior knowledge by using CMBP to do initialization.

RPN contains two types of nodes, *input node* or *computational node*. Every node x has a value at every time t, denoted by $u_x^{(t)}$. The values of computational nodes at time t are evaluated using the nodes' values at time t and the nodes' values at time $t-1$ as inputs just like Recurrent Neural Networks (RNNs).

Two types of edges are introduced to denote the time relation between linked nodes. A node at time t takes the value of a node at time $t-1$ as input when they are connected by *type-1* edges, while *type-2* edges indicate that a node at time t takes the value of a node at time t.

Let I_x denote the set of nodes which are connected to node x by *type-1* edges. Similarly, let $\hat{I}_x$ denote the set of nodes which are connected to node x by *type-2* edges.

Generally, three types of computational node are used in RPN, which are *sum node*, *product node* and *activation node*.

- Sum node: For sum node x at time t, its value $u_x^{(t)}$ is the weighted sum of its inputs:

$$u_x^{(t)} = \sum_{y \in I_x} w_{x,y} u_y^{(t-1)} + \sum_{y \in \hat{I}_x} \hat{w}_{x,y} u_y^{(t)} \tag{4}$$

  where $w_{x,y}, \hat{w}_{x,y} \in \mathbb{R}$ are the weights of edges.

- Product node: For product node x at time t, its value $u_x^{(t)}$ is the product of its inputs. Note that there may be multiple edges connecting from node y to node x. Then node y's value should be multiplied to $u_x^{(t)}$ multiple times. Formally, let $M_{x,y}$ and $\hat{M}_{x,y}$ be the multiplicity of the *type-1* edge $\overrightarrow{yx}$ and the multiplicity of the *type-2* edge $\overrightarrow{yx}$ respectively. Node x's value $u_x^{(t)}$ is evaluated by

$$u_x^{(t)} = \prod_{y \in I_x} {u_y^{(t-1)}}^{M_{x,y}} \prod_{y \in \hat{I}_x} {u_y^{(t)}}^{\hat{M}_{x,y}} \tag{5}$$

- Activation node: As the value of product nodes and sum nodes are not bounded by certain range while the output belief should lie in $[0, 1]$, activation functions are needed to map values from $\mathbb{R}$ to some interval such as $[0, 1]$. An activation function is a univariate function. If node x is an activation node, there is only one *type-2* edge linked to it.

  Sun et al. (2015) investigated several activation functions and proposed an ascending, continuous function $softclip$ mapping from $\mathbb{R}$ to $[0, 1]$ which is linear on $[\epsilon, 1-\epsilon]$ with ϵ being a small value.

Note that $w, \hat{w}$ are the only parameters in RPN while $M_{x,y}$ and $\hat{M}_{x,y}$ are constant given the structure of RPN and each node can be used as output node in RPN.

3.2.1 Basic Structure

A basic 3-layer RPN shown in figure 1 is introduced here to help understand the correlation between 3-order CMBP and RPN.

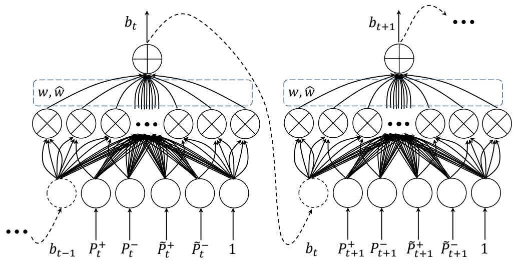
b_t
b_{t+1}
$w, \hat{w}$
$w, \hat{w}$
b_{t-1} P_t^+ P_t^- $\tilde{P}_t^+$ $\tilde{P}_t^-$ 1
b_t P_{t+1}^+ P_{t+1}^- $\tilde{P}_{t+1}^+$ $\tilde{P}_{t+1}^-$ 1

Figure 1: RPN for DST. (Here "+" nodes are sum nodes, "$\times$" nodes are product nodes)

For simplicity, (l, i) is used to denote the index of the i-th node in the l-th layer. Then each layer is defined as follows:

- First layer / Input layer: In this layer, input nodes correspond to the variables in equation (1), i.e. the value of 6 input nodes $u_{(0,0)}^{(t)} \sim u_{(0,5)}^{(t)}$ are the same as variables b_{t-1}, P_t^+, P_t^-, $\tilde{P}_t^+$, $\tilde{P}_t^-$, 1 in equation (1).

  Feature b_{t-1}^r which is belief of the value at time $t-1$ being $None$ is not used here to make the RPN structure clear and compact. Experiments show that performance of CMBP without feature b_{t-1}^r would not degrade. It is not used by CMBP mentioned in the rest of paper either.

- Second layer: Every product node x in the second layer corresponds to a monomial in equation (2). To express different monomials, each triple of input nodes $(1, k_1)$, $(1, k_2)$, $(1, k_3) (0 \leq k_1 \leq k_2 \leq k_3 \leq 5)$ is enumerated to link to a product node $x = (2, i)$ in the second layer and $u_x^{(t)} = u_{(1,k_1)}^{(t)} u_{(1,k_2)}^{(t)} u_{(1,k_3)}^{(t)}$.

- Third layer: There is only one sum node $(3, 0)$ in the third layer corresponding to the belief value calculated by a polynomial. With the parameters set according to g_{k_1,k_2,k_3} in equation (2), the value $u_{(3,0)}^{(t)}$ is equal to b_t outputted by equation (1). It is the only output node in this structure.

From the explanation of basic structure in this section, it can be easily observed that a CMBP can be used to initialize RPN and thus RPN can achieve at least the same results with CMBP. So prior knowledge and constraints are used to find a suboptimum point in RPN parameter space and RPN as a statistical approach, can further optimize its parameters. Hence, RPN is a way of bridging rule-based models and statistical models.

3.2.2 Complete Structure

It is easy to add features to RPN as a statistical model. In the work of Sun et al. (2015), 4 more features about user dialogue acts and machine acts are introduced.

A new sum node $x = (3, 1)$ in the third layer is introduced to capture some property across turns just like belief b_t. Like the node $(3, 0)$ that outputs belief in the same layer, node $(3, 1)$ takes input from every product node in the second layer and is used as input features at next time.

Further, to map the output belief to $[0, 1]$, activation nodes with $softclip(\cdot)$ as their activation function are introduced.

The complete structure with the activation function, 4 more features and the new recurrent connection is shown in figure 2.

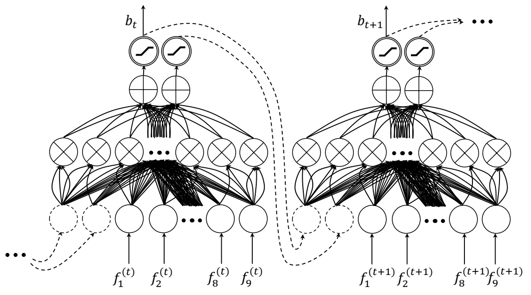
b_t
b_{t+1}
$f_1^{(t)}$ $f_2^{(t)}$ $f_8^{(t)}$ $f_9^{(t)}$
$f_1^{(t+1)}$ $f_2^{(t+1)}$ $f_8^{(t+1)}$ $f_9^{(t+1)}$

Figure 2: RPN with new features and more complex structure for DST (Sigmoid nodes mean activation)

The relation between a 3-order CMBP and the basic structure is shown in section 3.2.1. Similarly, the complete structure can also be initialized using CMBP by setting the weights of edges that do not appear in the basic structure to 0.

3.3 Complex RPN Structure

We next exam RPN's power of utilizing more features, multiple activation functions and a deeper structure with two interesting explorations on RPN structure are shown in this section. Although these extensions do not yield better results, this section covers these extensions to show the flexibility of the RPN approach.

3.3.1 Complex Structure

Firstly, to express a 4-order polynomial, simply using the structure shown in figure 2 with in-degree of nodes in the second layer increased to 4 would be sufficient. However, it can be expressed by a more compact RPN structure. To simplify the explanation, the example RPN expressing $1 - (1 - (b_{t-1})^2)(1 - (P_t^+)^2)$ is shown in figure 3.

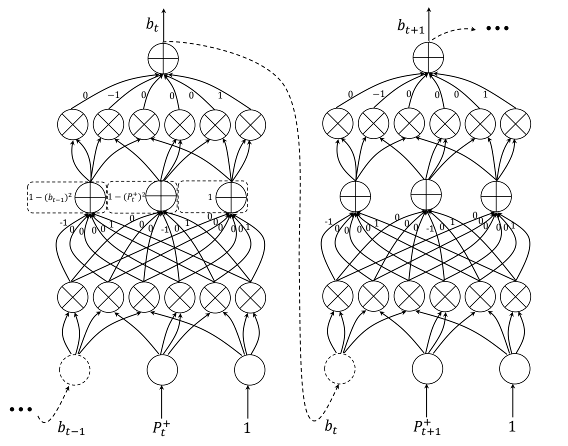

Figure 3: RPN for polynomial $1 - (1 - (b_{t-1})^2)(1 - (P_t^+)^2)$

In figure 3, the first layer is used for input, and the values of the product nodes in the second layer are equal to the products of two features such as $(b_{t-1})^2$, $b_{t-1}P_t^+$, $(P_t^+)^2$ and so on. Every sum node in the third layer can express all the possible 2-order polynomial of features with weights set accordingly. In figure 3, the values of the three sum nodes are $1 - (b_{t-1})^2$, $1 - (P_t^+)^2$ and 1 respectively. Then similarly, with another product nodes layer and sum nodes layer, the value of the output node in the last layer equals the value of the 4-order polynomial $(1 - (b_{t-1})^2)(1 - (P_t^+)^2)$.

The complete RPN structure with same features shown in figure 2, the new recurrent connection and activation nodes that expresses 4-order CMBPs can be obtained similarly.

With limited sum nodes in the third layer, the complexity of the model is much smaller than using a structure shown in figure 2 with product node's in-degree increased to 4 and increasing the number of product nodes accordingly.

3.3.2 Complex Features

Secondly, RNN proposed by Henderson et al. (2014c) uses n-gram of ASR results and machine acts. Similar to that, features of n-gram of ASR results and machine acts are also investigated in RPN. Since RPN used in this paper is a binary classification model and assumes slots independent of each other, the n-gram features proposed by Henderson et al. (2014c) are modified in this paper by removing/merging some features to make the features independent of slots and values. When tracking slot s and value v, the sum of confidence scores of ASR hypothesises of the following cases are extracted:

- V: confidence score of ASR hypothesises where value v appears
- $\tilde{V}$: confidence score of ASR hypothesises where values other than v appear
- V^r: confidence score of ASR hypothesises where no value appear

Similar features for slots can be extracted. Then by looking at both slot and value features for ASR results, we can get the combination of conditions of slots and values.

n-gram features of machine acts about the tracking slot and value are also used as features. For example, given machine acts `hello() | inform(area=center) | inform(food=Chinese) | request(name)`, for slot *food* and value *Chinese*, the n-gram machine act features are `hello`, `inform`, `request`, `inform+slot`, `inform+value`, `inform+slot+value`, `slot`, `value`, `slot+value`. Features such as `request(name)` are about slot *name* and hence `request+slot` are not in the feature list.

To combine RPN with RNN proposed by Henderson et al. (2014c), input nodes of these n-gram features are not linked to product nodes in the second layer. Instead, a layer of sum nodes followed by a layer of activation nodes with sigmoid activation function, which are equivalent to a layer of neurons are introduced. These activation nodes are linked to sum nodes in the third layer just like product nodes in the second layer. The structure is illustrated by figure 4 clearly.

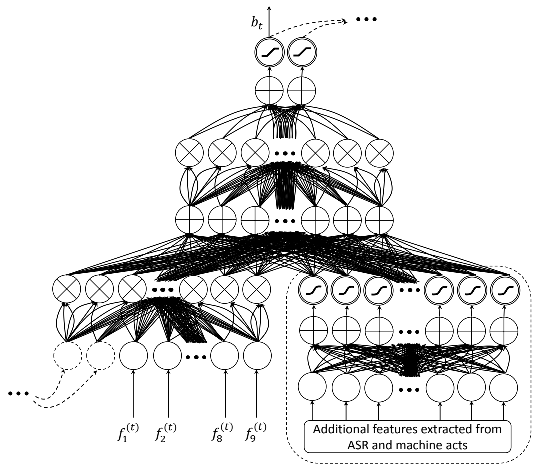
b_t
...
$f_1^{(t)}$ $f_2^{(t)}$ $f_8^{(t)}$ $f_9^{(t)}$
Additional features extracted from ASR and machine acts

Figure 4: RPN structure combined with RNN features and structures

Experiments in section 5 show that these two structures do not yield better results when initialized randomly or initialized using 3-order CMBPs, although the model complexity increases a lot. This indicates the briefness and effectiveness of the simple structure shown in figure 2.

4 Uncertainty in SLU

In an end-to-end dialogue system, there are two challenges in spoken language understanding: ASR errors and insufficient in-domain dialogue data.

ASR errors make information contained in the user's utterance distorted or even missed. Thankfully, statistical approaches to SLU, trained on labeled in-domain examples, have been shown to be relatively robust to ASR errors. (Mairesse et al., 2009).

Even with an effective way to get SLU robust to ASR errors, it is hard to implement these SLUs for a new domain due to insufficient labelled data. In DSTC-3, only little data of new dialogue domain is provided.

Following the work of Zhu et al. (2014), the following steps are used to handle the two challenges stated above:

- Data generation: with sufficient data in restaurants domain in DSTC-2, data on tourists domain using ontology of DSTC-3 can be generated. Utterance patterns of data in the original domain are used to generate data for the new domain of DSTC-3. After preparing both the original data in DSTC-2 and the generated data of DSTC-3, a more general parser for these two domains can be built.
- ASR error simulation: after data generation, ASR error simulation (Zhu et al., 2014) is needed to make the prepared data resemble ASR output with speech recognition errors to train a parser robust to ASR errors. With a simple mapping from the pattern of transcription to the corresponding patterns of ASR n-best hypotheses learned from existing data and phone-based confusion for slot-values, pseudo ASR n-best hypotheses can be obtained. Note that methods proposed by Zhu et al. (2014) only do ASR error simulation for generated data in domain of DSTC-3 and leave the original data in DSTC-2 as its original ASR form,which may introduce the difference in the distribution between training data and testing data on two different domains for the tracker. So ASR error is simulated in data on both domains instead.
- Training: Using the data got from the previous steps, a statistical parser can be trained (Henderson et al., 2012). By varying the fraction of simulated vs. real data, and the simulated error rate, prior expectations about operating conditions can be expressed.

Although a semantic parser with state-of-the-art techniques can achieve good performance in some degree, parsing without any error is impossible because it is typical that a semantic parser gets high performance in speech patterns existing in the training dataset, while it fails to predict the correct semantics for some utterances unseen in training dataset. So it is common for SLU performance to differ significantly between training and test conditions in real world end-to-end systems.

It has been widely observed that SLU influences state tracking greatly because the confidence scores of SLU hypotheses are usually the key inputs for dialogue state tracking. When these confidence scores become unreliable, the performance of tracker is sure to degrade. Studies have shown that it is possible to improve SLU accuracy as compared to the live SLU in the DSTC data (Zhu et al., 2014; Sun et al., 2014b). Hence, most of the state-of-the-art results from DSTC-2 and DSTC-3 used refined SLU (either explicitly rebuild a SLU component or take the ASR hypotheses into the trackers (Williams, 2014; Sun et al., 2014b;

Henderson et al., 2014d; Henderson et al., 2014c; Kadlec et al., 2014; Sun et al., 2014a)). Kadlec et al.(2014) gets a tracking accuracy improvement of 7.6% when they use SLU refined by themselves instead of organiser-provided live SLU.

In semantic parser mismatch condition, the accuracy of state tracking can degrade badly. Mismatched SLU problem is a main challenge in DST. Trackers under mismatched SLU conditions are investigated in this paper.

5 Experiments

5.1 RPN with Different Structures

In this section, the performance of three structures shown in this paper is compared and RPN with the simple structure is evaluated on DSTC-3 and compared with the best submitted trackers. Only joint goal accuracy which is the most difficult task of DSTC-3 is of interest. Note that the integer-coefficient CMBP with the best performance on DSTC-2 is used to initialize RPN. As it is stated in section 4, SLU designed in this paper focuses on domain extension, so trackers are only evaluated on DSTC-3.

Order	n-gram features	Acc	L2
3	No	0.652	0.540
4	No	0.648	0.541
4	Yes	0.648	0.541

Table 1: Performance comparison among RPNs with three structures on `dstc3eval`

The RPN structures that express 3-order CMBP, 4-order CMBP without n-gram features and 4-order CMBP with n-gram features are evaluated. *Acc* is the accuracy of tracker's 1-best joint goal hypothesis, the larger the better. *L2* is the L2 norm between correct joint goal distribution and distribution tracker outputs, the smaller the better.

It can be seen from table 1 that the simple structure yields the best result. Note that parser used here is explained in work (Zhu et al., 2014). Experiments of the mismatched SLU case also use this SLU for training.

For DSTC-3, it can be seen from table 2, RPN trained on DSTC-2 can achieve state-of-the-art performance on DSTC-3 without modifying tracking method, outperforming all the submitted trackers in DSTC-3 including the RNN system.

Note that the simple structure is used here with SLU refined described in section 4. We picked the best practical one on dstc2-test among SLUs introduced in the following section as the training SLU and testing SLU.

System	Approach	Rank	Acc	L2
Baseline*	Rule	6	0.575	0.691
Henderson et al. (2014c)	RNN	1	0.646	0.538
Kadlec et al. (2014)	Rule	2	0.630	0.627
Sun et al. (2014a)	Int CMBP	3	0.610	0.556
RPN	RPN	0.5	0.660	0.518

Table 2: Performance comparison among RPN, real-coefficient CMBP and best trackers of DSTC-3 on `dstc3eval`. Baseline* is the best results from the 4 baselines in DSTC3.

5.2 RPN with Mismatched Semantic Parsers

As section 4 stated, SLU is the input module for dialogue state tracking whose confidence score is usually directly used as probability features and hence has tremendous effect on trackers. Handling mismatched semantic parsers is a main challenge to DST.

In this section, different tracking methods are evaluated when there is a mismatch between training data and testing data. More specifically, different tracking models are trained with the same fixed SLU and tested with different SLUs.

Three main categories of tracking models are investigated: rule-based models, statistical models and mixed models.

MaxEnt (Sun et al., 2014b) is a statistical model. HWU baseline (Wang, 2013) is selected as a competitive rule-based model. CMBP and RPN are mixed models.

Four type of SLUs with different levels of performance are used:

1 *Original*: SLU results provided by DSTC-3 organizer.

2 *Train*: SLU introduced in section 4 with $k(k = 25, 50)$ percent training data adding ASR error simulation and parsed on ASR-hypotheses.

3 *Combined*: SLU combining the *Original* type SLU and *Train* type SLU using averaging.

4 *Transcript*: SLU introduced in section 4 with k percent training data adding ASR error simulation and parsed on transcription. This setup assumes an oracle speech recognizer: it is not practical, and is included only for comparison.

It has been shown that the organiser-provided live SLU can be improved upon and so it is used as the worst SLU in the following comparison. Past work has shown that trained parser gets a performance improvement when combined with the one the organiser provided (Zhu et al., 2014). Using transcription for parsing gives a much more reliable SLU results than using ASR hypotheses. So generally speaking, performance of SLUs of different types is quite distinguished to each other. Six different SLUs whose performance score shown in table 3 are investigated.

SLU type	ASR error	ICE	Fscore	Precision	Recall
Original	-	1.719	0.824	0.852	0.797
Train	25%	1.441	0.836	0.863	0.811
	50%	1.425	0.837	0.862	0.813
Combined	25%	1.241	0.834	0.870	0.801
	50%	1.235	0.835	0.869	0.803
Transcript	50%	0.893	0.915	0.956	0.877

Table 3: Performance of six different SLUs

Note that ASR error here is the percent of training data with ASR error simulation when training SLU. The Item Cross Entropy (ICE) (Thomson et al., 2008) between the N-best SLU hypotheses and the semantic label assesses the overall quality of the semantic items distribution, and is shown to give a consistent performance ranking for both the confidence scores and the overall correctness of the semantic parser (Zhu et al., 2014). SLU with the lower ICE has better performance.

Precision and recall are evaluated using only SLU's 1-best hypothesis where ICE takes all hypothesises and their confidence score into consideration.

In results shown in figure 5, the training dataset for tracker is fixed, while testing dataset is outputted by different SLUs. The X-axis gives the SLU ICE and Y-axis gives the tracking accuracy on DSTC3-test. It can be observed that RPN achieves highest accuracy on every SLU among rule-based models, statistical models and mixed models. Thus, RPN shows its robustness on mismatched semantic parsers, which demonstrates the power of using both prior knowledge and being a statistical approach.

After evaluating the mismatched case, the matched case is also tested. When training dataset and testing dataset are outputted by the same SLU, RPN also outperforms all other models, shown in figure 6.

It can be observed that RPN achieves the highest accuracy among RPN, CMBP, MaxEnt, and

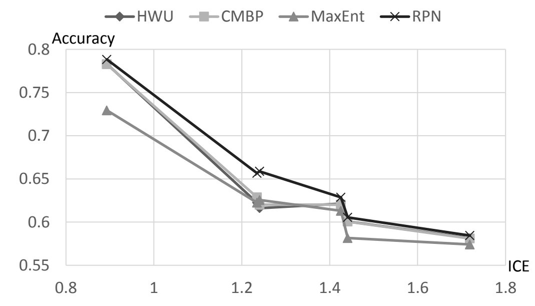

Figure 5: Trackers' performances with mismatched semantic parsers

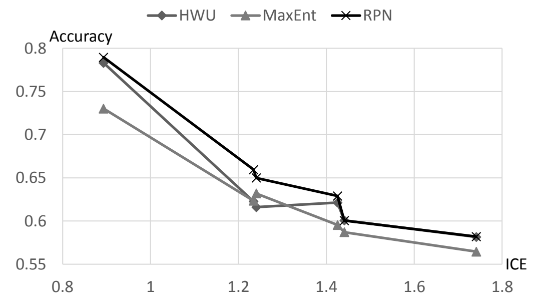

Figure 6: Trackers' performances with matched semantic parser

HWU baseline whether there is a mismatch between training SLU and testing SLU or not.

6 Conclusion

Recurrent Polynomial Network demonstrated in this paper is a recent framework to bridge rule-based and statistical models. Several networks are explored and the simple structure's performance outperforms others. Experiments show that RPN outperforms many state-of-the-art trackers on DSTC-3 and RPN performs best on all SLUs with mismatched SLU.

Acknowledgments

This work was supported by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC project No. 61222208 and the Chun-Tsung Program of SJTU. We thank Jason Williams for reviewing and providing suggestions to this paper.

Appendix

Activation function

An activation function $softclip(\cdot)$ is a combination of logistic function and clip function. Let ϵ denote a small value such as 0.01, δ denote the offset of sigmoid function such that $sigmoid\,(\epsilon - 0.5 + \delta) = \epsilon$. *sigmoid* function here is defined as

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

The softclip function is defined as

$$softclip(x) \triangleq \begin{cases} sigmoid\,(x - 0.5 + \delta) \\ \qquad \text{if } x \leq \epsilon \\ x \\ \qquad \text{if } \epsilon < x < 1 - \epsilon \\ sigmoid\,(x - 0.5 - \delta) \\ \qquad \text{if } x \geq 1 - \epsilon \end{cases} \tag{7}$$

It is a non-decreasing, continuous function, which is linear on $[\epsilon, 1 - \epsilon]$. Its derivative is defined as follows:

$$\frac{\partial softclip(x)}{\partial x} \triangleq \begin{cases} \frac{\partial sigmoid(x - 0.5 + \delta)}{\partial x} \\ \qquad \text{if } x \leq \epsilon \\ 1 \\ \qquad \text{if } \epsilon < x < 1 - \epsilon \\ \frac{\partial sigmoid(x - 0.5 - \delta)}{\partial x} \\ \qquad \text{if } x \geq 1 - \epsilon \end{cases} \tag{8}$$

Training

Backpropagation through time (BPTT) using mini-batch is used to train the network with batch size 50. Gradients of weights are calculated and accumulated within each batch. Gradients computed for each timestep are propagated to the first timestep. Mean squared error (MSE) is used as the criterion to measure the distance of the output belief to the correct belief distribution.

Derivative calculation

Let $\delta_x^{(t)}$ be the partial derivative of the cost function over value of node x, i.e., $\delta_x^{(t)} = \frac{\partial \mathcal{L}}{\partial u_x}$. Suppose node $x = (d, i)$ is a sum node, then when node x passes its error, the error of child node $y \in \hat{I}_x$ is updated as

$$\begin{aligned} \delta_y^{(t)} &= \delta_y^{(t)} + \frac{\partial \mathcal{L}}{\partial u_x^{(t)}} \frac{\partial u_x^{(t)}}{\partial u_y^{(t)}} \\ &= \delta_y^{(t)} + \delta_x^{(t)} \hat{w}_{x,y} \end{aligned} \tag{9}$$

Similarly, error of node $y \in I_x$ is updated as

$$\begin{aligned} \delta_y^{(t)} &= \delta_y^{(t)} + \frac{\partial \mathcal{L}}{\partial u_x^{(t)}} \frac{\partial u_x^{(t)}}{\partial u_y^{(t-1)}} \\ &= \delta_y^{(t)} + \delta_x^{(t)} w_{x,y} \end{aligned} \tag{10}$$

Suppose node $x = (d, i)$ is a product node, then when node x passes its error, error of node $y \in \hat{I}_x$ is updated as

$$\begin{aligned} \delta_y^{(t)} &= \delta_y^{(t)} + \frac{\partial \mathcal{L}}{\partial u_x^{(t)}} \frac{\partial u_x^{(t)}}{\partial u_y^{(t)}} \\ &= \delta_y^{(t)} + \\ &\quad \delta_x^{(t)} \hat{M}_{x,y} {u_y^{(t)}}^{\hat{M}_{x,y} - 1} \\ &\quad \prod_{z \in \hat{I}_x - \{y\}} {u_z^{(t)}}^{\hat{M}_{x,z}} \prod_{z \in I_x} {u_z^{(t-1)}}^{M_{x,z}} \end{aligned} \tag{11}$$

Similarly, error of node $y \in I_x$ is updated as

$$\begin{aligned} \delta_y^{(t)} &= \delta_y^{(t)} + \frac{\partial \mathcal{L}}{\partial u_x^{(t)}} \frac{\partial u_x^{(t)}}{\partial u_y^{(t-1)}} \\ &= \delta_y^{(t)} + \\ &\quad \delta_x^{(t)} M_{x,y} {u_y^{(t-1)}}^{M_{x,y} - 1} \\ &\quad \prod_{z \in \hat{I}_x} {u_z^{(t)}}^{\hat{M}_{x,z}} \prod_{z \in I_x - \{y\}} {u_z^{(t-1)}}^{M_{x,z}} \end{aligned} \tag{12}$$

References

Matthew Henderson, Milica Gasic, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young. 2012. Discriminative spoken language understanding using word confusion networks. In *SLT*, pages 176–181.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014a. 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. Association for Computational Linguistics.

Matthew. Henderson, Blaise. Thomson, and Jason D. Williams. 2014b. The third dialog state tracking challenge. In *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, December.

Matthew. Henderson, Blaise. Thomson, and Steve. Young. 2014c. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, December.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014d. 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. Association for Computational Linguistics.

Rudolf Kadlec, Miroslav Vodoln, Jindrich Libovick, Jan Macek, and Jan Kleindienst. 2014. Knowledge-based dialog state tracking. In *Proceedings 2014 IEEE Spoken Language Technology Workshop*, South Lake Tahoe, USA, December.

Sungjin Lee and Maxine Eskenazi. 2013. Recipe for building robust spoken dialog state trackers: Dialog state tracking challenge system description. In *Proceedings of the SIGDIAL 2013 Conference*, pages 414–422, Metz, France, August. Association for Computational Linguistics.

Sungjin Lee. 2013. Structured discriminative model for dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 442–451, Metz, France, August. Association for Computational Linguistics.

François Mairesse, Milica Gasic, Filip Jurcícek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2009. Spoken language understanding from unaligned data using discriminative classification models. In *Proceedings of ICASSP*.

Kai. Sun, Lu. Chen, Su. Zhu, and Kai. Yu. 2014a. A generalized rule based tracker for dialogue state tracking. In *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, December.

Kai Sun, Lu Chen, Su Zhu, and Kai Yu. 2014b. 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)*, pages 318–326, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.

Kai Sun, Qizhe Xie, and Kai Yu. 2015. Recurrent polynomial network for dialogue state tracking. *submitted to Dialogue and Discourse*.

Blaise Thomson, Kai Yu, Milica Gasic, Simon Keizer, Francois Mairesse, Jost Schatzmann, and Steve Young. 2008. Evaluating semantic-level confidence scores with multiple hypotheses. In *INTERSPEECH*, pages 1153–1156.

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, Metz, France, August. Association for Computational Linguistics.

Zhuoran Wang. 2013. HWU baseline belief tracker for dstc 2 & 3. Technical report, October.

Jason D. Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, Metz, France, August. Association for Computational Linguistics.

Jason D. Williams. 2012. Challenges and opportunities for state tracking in statistical spoken dialog systems: Results from two public deployments. *Selected Topics in Signal Processing, IEEE Journal of*, 6(8):959–970.

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, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.

Kai Yu, Kai Sun, Lu Chen, and Su Zhu. 2015. Constrained markov bayesian polynomial for efficient dialogue state tracking. *submitted to IEEE Transactions on Audio, Speech and Language Processing*.

Su Zhu, Lu Chen, Kai Sun, Da Zheng, and Kai Yu. 2014. Semantic parser enhancement for dialogue domain extension with little data. In *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, December.

Lukas Zilka, David Marek, Matej Korvas, and Filip Jurcicek. 2013. Comparison of bayesian discriminative and generative models for dialogue state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 452–456, Metz, France, August. Association for Computational Linguistics.