Front. Comput. Sci.
DOI 10.1007/s11704-015-5209-4

REVIEW ARTICLE

Evolvable dialogue state tracking for statistical dialogue management

Kai YU (✉)[1], Lu CHEN[1], Kai SUN[2], Qizhe XIE[1], Su ZHU[1]

1 Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
2 Department of Computer Science, Cornell University, Ithaca, NY 14853, USA

Abstract Statistical dialogue management is the core of cognitive *spoken dialogue systems* (SDS) and has attracted great research interest. In recent years, SDS with the ability of evolution is of particular interest and becomes the cutting-edge of SDS research. Dialogue state tracking (DST) is a process to estimate the *distribution* of the dialogue states at each dialogue turn, given the previous interaction history. It plays an important role in statistical dialogue management. To provide a common testbed for advancing the research of DST, international DST challenges (DSTC) have been organised and well-attended by major SDS groups in the world. This paper reviews recent progresses on rule-based and statistical approaches during the challenges. In particular, this paper is focused on evolvable DST approaches for dialogue domain extension. The two primary aspects for evolution, semantic parsing and tracker, are discussed. Semantic enhancement and a DST framework which bridges rule-based and statistical models are introduced in detail. By effectively incorporating prior knowledge of dialogue state transition and the ability of being data-driven, the new framework supports reliable domain extension with little data and can continuously improve with more data available. This makes it excellent candidate for DST evolution. Experiments show that the evolvable DST approaches can achieve the state-of-the-art performance and outperform all previously submitted trackers in the third DSTC.

Received June 1, 2015; accepted September 28, 2015

E-mail: kai.yu@sjtu.edu.cn

Keywords dialogue management, domain extension, evolvable dialogue state tracking, parser, tracker

1 Introduction

A task-oriented spoken dialogue system (SDS) is a system that can continuously interact with human to accomplish a predefined task through *speech*. It usually consists of three modules: input, output and control, as shown in Fig. 1. The input module mainly consists of automatic speech recognition (ASR) and spoken language understanding (SLU), with which semantics-level user dialogue acts are extracted from acoustic speech signals. With the input user dialogue acts, the control module, referred to as dialogue management, accomplishes two missions. One is to maintain its internal state, an encoding of the machine's understanding about the whole conversation. When information is received from the input module, the state space is updated, which is called *dialogue state tracking* (DST). Another mission is to choose a machine action, also at semantics-level, based on the dialogue state space according to a *policy* to direct the dialogue. This is referred to as *dialogue decision making*. The output module consists of natural language generation (NLG) and text-to-speech (TTS) synthesis, with which machine dialogue acts are converted to audio.

Dialogue management is the core of a dialogue system. Traditionally, most commercial spoken dialogue systems assume observable dialogue states and employ hand-crafted rules for dialogue management, such as dialogue flow-chart.

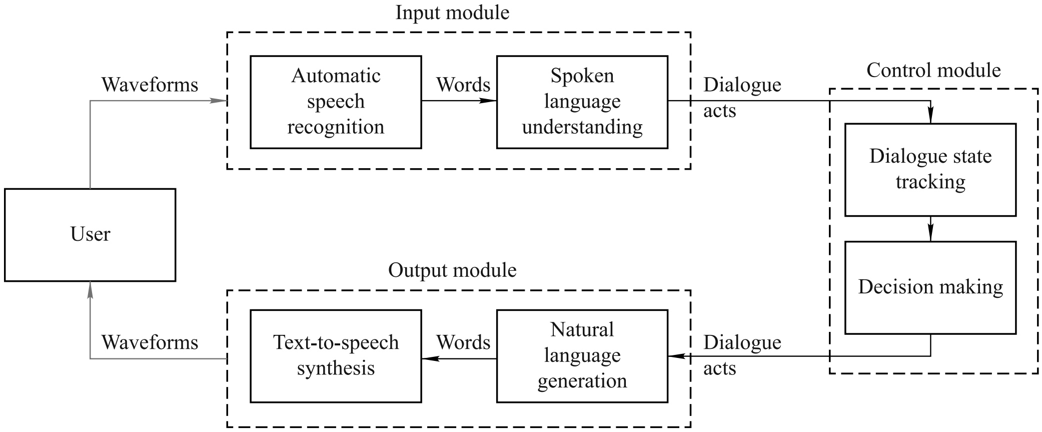


Fig. 1 Diagram of a spoken dialogue system (SDS)

Since dialogue state is observable, no tracking is needed. Dialogue decision is simply a set of mapping rules from state to machine action. This is referred to as rule-based dialogue management. However, unpredictable user behaviour, inevitable automatic speech recognition and spoken language understanding errors make it difficult to maintain the true dialogue state and make decision. Hence, in recent years, there is a research trend towards statistical dialogue management.

A well-founded theory for this is the partially observable Markov decision process (POMDP), which can provide robustness to errors from input module and automatic policy optimisation by reinforcement learning [1–4]. Under the POMDP framework, both dialogue state tracking and decision making are often modelled using statistical approaches. Recently, to advance the research of statistical dialogue management, researchers start to formulate dialogue state tracking as an independent problem so that a bunch of machine learning algorithms can be investigated. The dialogue state tracking challenge (DSTC) provides the first common testbed in a standard format, along with a suite of evaluation metrics for this purpose [5–9].

In most studies of POMDP SDS, domain and ontology, including slots, values and relation between slots, are assumed to be static and usually very simple. Although lab-scale statistical SDS of highly constrained domain has shown significant improvement over traditional rule-based SDS, domain ontology and semantics in real world tasks are usually open, and migrating SDS has become a frequent requirement. Hence, evolvable SDS becomes increasingly important in both research community and industry. To reflect this, the third DSTC provides a task of domain extension.

This paper reviews various DST approaches in the DSTCs, including rule-based and statistical approaches. In particular, how to build an evolvable tracker is the focus. Two issues need to be addressed here: extension of the semantic parser to include new slots/values and modification of the tracker for the new domain. Semantic enhancement using data augmentation provides an effectively way to extend statistical semantic parser without losing the ability of being tolerant to speech recognition errors. On the tracker side, a newly proposed framework to bridge the rule-based and statistical approaches is introduced. Two models within the framework, *constrained markov bayesian polynomial* (CMBP) [10] and *recurrent polynomial network* (RPN) [11], are discussed in detail. Since the new framework can effectively incorporate general knowledge of state transition, its initial performance is stable across different domains. On the other hand, both models can continuously improve with more data available. These characteristics make the framework particularly useful for evolvable DST.

The paper is organized as follows. Section 2 introduces evolvable dialogue state tracking. In Section 3, the evolvable parser is described. Section 4 reviews and compares the different methods for dialogue state tracking, Section 5 describes CMBP and RPN. Experiments are detailed in Section 6, followed by the conclusion in Section 7.

2 Evolvable dialogue state tracking

A dialogue can be regarded as a time sequence $\{\mathbf{a}_0, \mathbf{o}_1, \ldots, \mathbf{a}_{t-1}, \mathbf{o}_t\}$, where $\mathbf{a}_i$ is the system information at the ith dialogue turn, including the system response, and $\mathbf{o}_i$ denotes all information from the user's speech at the ith turn, e.g., the output of SLU. At each turn, dialogue state tracking (DST) is to estimate the probability distribution of the state given the whole dialogue history up to that turn, also referred to as belief state $b_t(\mathbf{s})$, or briefly b_t,

$$\begin{aligned} b_t(\mathbf{s}) &= \mathcal{T}(\mathbf{s}, b_0, \mathbf{a}_0, \mathbf{o}_1, \ldots, \mathbf{a}_{t-1}, \mathbf{o}_t) \\ &= P(\mathbf{s}|b_0, \mathbf{a}_0, \mathbf{o}_1, \ldots, \mathbf{a}_{t-1}, \mathbf{o}_t), \end{aligned} \quad (1)$$

where $\mathcal{T}(\cdot)$ denotes the tracker and b_0 is the initial belief state.

As shown in Eq. (1), the result of DST is affected not only by the functional form of tracker $\mathcal{T}(\cdot)$, but also by the sequence $\{\mathbf{a}_0, \mathbf{o}_1, \ldots, \mathbf{a}_{t-1}, \mathbf{o}_t\}$ and the initial belief state b_0. The initial state is usually assumed to be uniformly distributed. In the DST context in this paper, the information $\mathbf{o}_i$ from user's speech refers to the output SLU. It is worth noting that this output is not just a single hypothesis. Instead, it is an N-best list of semantic hypotheses, which describes the whole hypothesised semantic space.

For an end-to-end spoken dialogue system, a good dialogue state tracker should satisfy:

- **Accuracy** The tracker should be as accurately as possible to estimate the system state. It has been shown that the improvement of tracking accuracy can benefit for the task completion rates in the end-to-end spoken dialogue system [12].
- **Efficiency** As shown in Fig. 1, the tracker is only a small component in the whole system. In order to achieve real-time conversation, the tracker should compute as fast as possible.
- **Generalization** In practice, it is hard to collect enough dialogues for training before a system is employed, which is often the case whenever a new domain is encountered or the current domain is extended. Therefore, it is important that the tracker can work well in the new domain or extended domain.

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. In early works of POMDP, belief state is updated using Bayes' theorem with consideration of Markov and reasonable independence assumptions. This leads to the below update formula for b_t:

$$\begin{aligned} b_t(\mathbf{s}_t) &= P(\mathbf{s}_t|\mathbf{o}_t, a_{t-1}, b_{t-1}) \\ &= k \cdot P(\mathbf{o}_t|\mathbf{s}_t, a_{t-1}) \sum_{\mathbf{s}_{t-1} \in S} P(\mathbf{s}_t|\mathbf{s}_{t-1}, a_{t-1}) b_{t-1}(\mathbf{s}_{t-1}), \end{aligned} \quad (2)$$

where k is the normalization constant and a_{t-1} is the system response at the $(t-1)$th turn. Eq. (2) is a generative model for DST. Due to huge number of possible states, approximation is necessary for DST in real world tasks. State space partition (hidden information state (HIS)) [3] or further state independence assumption (Bayesian network update of dialogue state (BUDS)) [2] have been used. However, these generative methods neither accurately nor efficiently track the dialogue state.

To advance the statistical dialogue management research, the *dialogue state track challenges* (DSTCs) are organized to provide common testbeds for comparing different DST models. There have been three challenges, each with a different task. All challenges consider the tracking of users' goals and employ labelled dialogue corpus and simplified dialogue state representations. The first challenge, DSTC-1, investigates DST evaluation and suggests two primary metrics for evaluation: accuracy of joint goals of all slots and Brier score which is the L2 distance between the estimated state distribution from the tracker and the real state distribution (i.e., an indicator function) [7]. In DSTC-2 [8], the domain changes to restaurant search with eight slots, which is more complicated and realistic.

Since transplanting SDS from a domain to another is of both theoretical and practical interest, DSTC-3 [9] designs tasks for building trackers for domain extension. Only a small set of labelled dialogues in a new domain (tourist information) are available, and all participants are asked to build a belief state tracker on the small data set plus the DSTC-2 data (restaurant domain). The new domain has 13 slots, which include all slots in DSTC-2 and five new slots. This is a typical domain extension scenario for testing the evolvement ability of the trackers. It is worth noting that there are two aspects to be considered during this process: parser extension and tracker extension. The below sections will detail algorithms for both aspects.

3 Evolvable statistical semantic parser

Semantic parser serves as an interface between automatic speech recognition (ASR) and dialogue state tracking (DST), which aims at understanding user's intentions of their current speech utterances. Typically, state tracking has assumed the output of a spoken language understanding (SLU) component in the form of a semantic decoder, which maps the hypotheses from automatic speech recognition (ASR) to a list of semantic hypotheses [13].

3.1 Dialogue acts

Once the system gets a set of ASR hypotheses with confidence scores, it needs to interpret the meaning of these word sequences. For a task-oriented SDS, the subtle and exact meaning is not important. It is only crucial to catch the user's goal related to the task. For example, whether the user says "I want to find a restaurant nearest." or "Could you show me the nearest restaurant?", the intention should be the same. The user is asking for a nearest restaurant.

Dialogue acts [14] are high-level meaning representation which can encapsulate a variety of the user's intentions. Dialogue act is a semantic functor consisting of a dialogue act type and slot-value pair, in form of

$$actType(slot = value),$$

where *actType* can be "inform" (for statement), "request" (for asking), etc., *slot* refers to one attribute that helps represent the user's intention and *value* is the corresponding value. Note that the semantic of one utterance may be made up of multiple dialogue acts. For example, "request(name) inform(food=Chinese)" refers to asking for the name of a Chinese restaurant.

3.2 Statistical semantic parsers

The task of spoken language understanding differs from usual semantic parsing in natural language processing in the sense that the input is erroneous ASR hypothesis and sometimes multiple hypotheses in the form of N-best list or lattices. The difficulty here is to achieve robustness to ASR errors.

There are a variety of approaches available for semantic parsing. Context-free grammar, or rule based method, is an example of hand-crafted approaches, which can be implemented easily but quite sensitive to ASR errors. To alleviate this issue, data-driven approaches are developed by learning the semantic representation from data statistically. In these methods, semantic parsing is regarded as a sequence labelling problem with aligned training data in which each word is tied with a semantic tag sequentially. Hidden vector state (HVS) model learns a probabilistic push-down automaton [15]. Machine translation techniques [16] consider semantic parsing as a task of translation from word sequence to semantic labelled sequence. Zettlemoyer & Collins present a grammar induction method that can learn a probabilistic combinatory categorial grammar (PCCG) from utterance-level annotations [17]. Conditional random fields (CRF) [18] shows good performance in semantic labelling. Weighted Finite State Transducers [19] have also been used. Recently, deep learning techniques are used in SLU, such as recurrent neural networks (RNN) [20], long short-term memory (LSTM) neural networks [21] and recursive neural networks (RecNNs) [22].

Different from the above methods which consider semantic parsing as a sequence labelling problem, the semantic tuple classifiers approach [23] builds classifiers on the whole sentence. This is critical for real world SLU, since it is usually hard to achieve word-semantics slot alignment due to ASR errors. In this method, a binary classifier is trained for each actType-slot and slot-value pair, and predicts the presence of this pair in the utterance. Similarly, a multi-class classifier is estimated for all the dialogue act types. Finally, all outputs of these classifiers are combined to be the predicted dialogue acts.

Beyond ASR top-hypothesis, N-best list of ASR hypotheses provides more words and information, which will improve the performance of semantic parsing significantly. Meanwhile, in a conversational dialogue system, not only the output of ASR, but also the history of dialogue can be used in semantic parsing [24].

3.3 Evolvable parser in dialogue domain extension

Statistical semantic parser trained on sufficient in-domain data has shown to be robust to errors of automatic speech recognition (ASR) [23–25], especially in end-to-end spoken dialogue systems.

In order to build practical systems, semantic parser should have portability when the domain of a dialogue system is extended. However, when the dialogue domain is changed or expanded, the performance of a semantic parser is usually significantly degraded as a result of the introduction of new semantic slots, values and unknown speech pattern (or ASR hypothesis pattern). Thus, the ability of SLU to cope well with the expanded domains and limited training data is very attractive to the deployment of commercial dialogue systems.

We proposed a practical semantic parser enhancement [26], which shows good scalability in dialogue domain extending with little labelled example data. It employs automatic pseudo-data generation for parser re-training and domain independent rescoring to further improve parsing performance.

3.3.1 ASR hypotheses simulation and parser re-training

ASR hypotheses simulation used as automatic pseudo-data generation helps to re-train semantic parsers for the extended dialogue domain. ASR hypotheses simulation is implemented on word level to generate additional training data adapted from the original domain to the extended domain. These new sentences are generated for the new slots and values, which contain new sentence patterns and new text contexts of the values, as shown in Fig. 2.

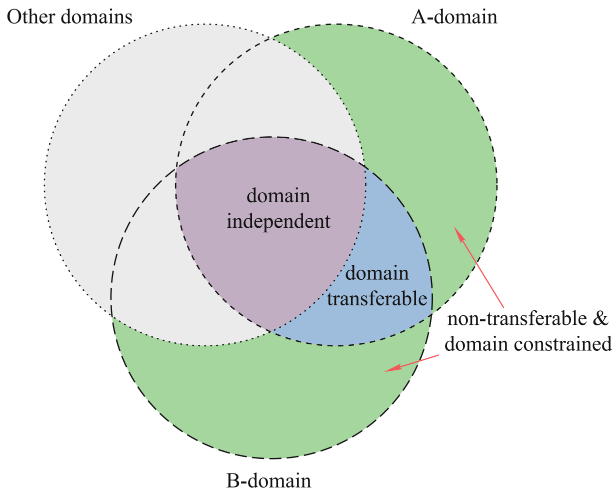


Fig. 2 A general comparison of different dialogue domains (A-domain and B-domain refer to two different dialogue domains)

In general, the semantic data of different dialogue domains can be classified as:

- **Domain independent** data samples independent of any specific domain, e.g., users say hello or thank you.
- **Domain transferable** data samples appearing in both the original and the extended domain.
- **Non-transferable & domain constrained** data samples specific for one domain, which can not be transferred to another domain.

ASR hypotheses simulation focuses on generation of the third part of data samples (non-transferable & domain constrained). With these transferred and generated data, the semantic parser for the extended domain is easy to be built. The main idea of data generation is based on combining old sentence patterns with new slot-values, and new sentence patterns with old slot-values. Sentence pattern refers to a sentence structure with slot-values replaced with some special label, for example, the pattern of "I need moderately priced Chinese food" is simply "I need [pricerange] priced [food] food".

In Ref. [26], ASR hypotheses simulation shows a significant improvement in contrast to the simple approach without data generation in DSTC-3 where F-score of dialogue act increases from 0.808 to 0.833.

3.3.2 Domain-independent SLU rescoring

In addition to parser re-training, we proposed a new view to make use of system act for dialogue domain extension. System act is the semantics that the machine feeds back to the user and has some relationship with what the user would say next. Henderson et al. trained semantic tuple classifiers by concatenating the text feature and the last system act feature [24]. But in the extended dialogue domain, it is more convenient to use the system act features independent of the dialogue domain. Hence, the last system act features independent of dialogue domain are exploited to train a SLU rescoring system in the original domain, and applied to the semantic parser enhancement in the extended domain.

The proposed rescoring approach is to train a classifier for each semantic item, like actType, actType-slot pair and slot-value pair. The domain-independent features used in rescoring are listed below:

- **SLU score** the output probability of the original semantic parser, e.g., semantic tuple classifiers.
- **System act type** an indication feature for each dialogue act type whether it exists in the last system act.
- **Acttype-slot** a feature giving the indication of whether each (acttype, slot) pair exists in the last system act.
- **Slot-value** a feature giving the indication of whether each (slot, value) pair exists in the last system act.

Moreover, more domain-independent features can be extracted and exploited. In Ref. [26], it is shown that SLU rescoring yields slight improvements in addition to the ASR hypotheses simulation method.

4 Evolvable tracker

Given multiple SLU hypotheses with confidence scores, the aim of a tracker is to estimate the distribution of user's goal. Since tracker needs to take into account the interaction history and the output is a distribution, the tracking task is not a typical classification tasks. In general, there are two types of approaches for dialogue state tracking — statistical approach and rule-based approach.

4.1 Statistical tracker

Before the DSTCs, most statistical DST approaches are Bayesian generative models. Although they are mathematically sound, it is hard to incorporate rich features for DST, and sometimes they are intractable [27]. Hence, statistical discriminative models, such as maximum entropy model (MaxEnt) [27, 28], conditional random field (CRF) [29, 30], deep neural networks (DNN) [28, 31], recurrent neural networks (RNN) [9, 32] and decision forest [33], have been used as the statistical DST models and achieved great success since DSTC-1. With different assumptions on slot and value independence, these approaches fall into four main categories: binary classification model, multi-classification model, structured discriminative model and labelling model.

- **Binary classification models** Here all slots are assumed to be *independent* of each other, leading to an efficient state factorization:

$$b(s_1 = v_1, \ldots, s_n = v_n) = \prod_j b(s_j = v_j). \qquad (3)$$

  In addition, for a slot, all candidate values which have not been observed up to the current turn are clustered together as a special value "*None*". This significantly reduces the computational cost. With these assumptions, the joint goal can be easily obtained by calculating the belief $b(s = v)$ for each slot s and candidate value v. This can be converted into a binary classification problem of determining whether $s = v$ is true or false. Various models have been used within this framework, such as MaxEnt [27, 28] and DNN [28, 31].

- **Multi-classification models** In the binary classification models, the belief of every candidate value v is evaluated separately, which may degrade the performance. To address this issue, multi-classifier is used to track the belief of all values simultaneously. Same as the binary classification models, different slots are assumed to be independent of each other, thus the belief state of each slot can be updated separately, and the belief of joint goal is calculated by Eq. (3). A typical example is RNN [9, 32].

- **Structured discriminative models** In both binary and multi-classification models, slots are assumed to be independent of each other. Considering relational constraints may result in potential improvement of the DST performance. *Structured discriminative models* are proposed to capture the relationship between slots at a particular turn. A typical example is CRF with manually designed factored graph [29]. Another example is a web-style ranking model (decision forest model) [33], which can automatically build conjunctions of raw features to track the belief state of joint slots.

- **Labelling models** Although structured discriminative models utilize the relational constraints between different slots, they only focus on information of a single turn. In DSTC-2, a sequential labelling model is proposed to capture the relationship between multiple turns [30]. In this approach, the output of the model includes labels of the dialogue state from multiple turns. To model the temporal relationship, a linear-chain CRF is used [30].

It is worth noting that features play an important role here. In the DSTCs, the available information includes speech recognition and semantic parsing results as well as the system response history. Since N-best results are also available, various features, such as confidence scores, ranks, and statistics of confidence scores, are commonly used as features. For the speech recognition results, the most common feature is the n-gram feature weighted by confidence scores [32]. The system dialogue acts can provide useful information for state estimation [28, 32]. Besides these features, the *turn-id* of the dialogue, whether the user has interrupted the system etc., can also be used as features [27, 28, 31]. The main advantage of statistical discriminative model is the ability of incorporating various forms of features, and the performance is very good with sufficient training data.

4.2 Rule-based tracker

Since the DST problem is raised out of the statistical dialogue management framework, statistical approaches have been the natural focus. However, statistical approaches has also shown large variation in performance and poor generalisation ability due to the lack of data. There has been also an attempt to employ rule-based methods due to its simplicity, efficiency, portability and interpretability. For example, the standard POMDP belief update can be seen as a rule-based model, when all parameters are set according to prior knowledge without data-driven estimation [34]. During DSTCs, a couple of more interesting rule-based models [35] have also been proposed, one of which is explained in detail in the CMBP section. The advantage of rule-based model is being domain independent, and hence no evolvement issue to be considered.

4.3 Mixed trackers

Although statistical models can improve with more data and achieve good performance, it is relatively hard to use it for domain extension due to lack of data. Rule-based approaches are not bothered by the domain change issue, however, the performance of rule-based model is usually poor and they lack the ability of evolvement with data.

To address these limitations, approaches bridging rule-based and statistical models are proposed: constrained Markov Bayesian polynomial (CMBP) and recurrent polynomial network (RPN). They have the advantages of both rule-based models and statistical models and will be reviewed in detail in the next section. The comparison of different DST models is shown in Table 1.

Table 1 Comparison of different models

Type	Tracker	Accuracy	Efficiency	Generalization
Rule	Rule-based	×	√	√
Statistical	Generative	×	×	×
	Discriminative	√	√	×
Mixed	CMBP/RPN	√	√	√

Note: '√' means that most of trackers have the corresponding merit. '×' means that most of trackers do not have the corresponding merit

5 Bridging rule-based and statistical approaches

Taken into consideration of benefits and weaknesses of statistical models and rule-based models, mixed trackers are able to transcend their limitations by combining the advantages of rule-based models and statistical models. This makes the mixed tracker particularly useful for evolvable DST.

Broadly, there are two ways to bridge rule-based models and statistical models: one is to find a good rule using prior knowledge and data-driven approaches, while the other starts from statistical models and takes advantage of prior knowledge. Constrained Markov Bayesian polynomial (CMBP) takes the first way [10, 36], while recurrent polynomial network (RPN) takes the second way [11].

5.1 Constrained Markov Bayesian polynomial

5.1.1 Motivation

In DSTC-1, a simple rule-based model achieving good results was proposed by Wang et al. [35]. In the model, for slot s and value v, the belief of "the value of slot s being v in the tth turn", denoted as $b_t(v)$, is calculated as follows:

$$b_t(v) = \begin{cases} (1-(1-b_{t-1}(v))(1-P_t^+(v)))\cdot & \\ \quad (1-P_t^-(v)), & if\ v \neq \text{“}None\text{”}; \\ 1 - \displaystyle\sum_{v' \neq \text{“}None\text{”}} b_t(v'), & \text{otherwise}, \end{cases} \tag{4}$$

where $P_t^+(v)$ and $P_t^-(v)$ are used to denote the sum of SLU confidence scores that user informs or affirms v, and sum of SLU confidence scores that user negates or denies v, respectively.

Rule-based models [34, 35], and Bayesian generative models [3] are all based on Bayes' theorem. Since Bayes' theorem is essentially summation and multiplication of probabilities, they can be rewritten in a general polynomial form, referred to as Markov Bayesian polynomial (MBP) [10, 36]:

$$b_{t+1}(\mathbf{s}) = \mathcal{P}(b_t, \mathbf{q}_t), \tag{5}$$

where $b_{t+1}(\mathbf{s})$ is the belief state of $\mathbf{s}$ at the tth turn, $\mathbf{q}_t$ is the probability features about current user acts and machine acts, and $\mathcal{P}(\cdot)$ is a multivariate polynomial function

$$\mathcal{P}(x_1, x_2, \ldots, x_D) = \sum_{0 \leqslant k_1 \leqslant \cdots \leqslant k_n \leqslant D} g_{k_1,k_2,\ldots,k_n} \prod_{1 \leqslant i \leqslant n} x_{k_i} \tag{6}$$

where D is the number of input variables, $x_0 = 1$, n called order of MBP is the order of the polynomial. The coefficient $g_{k_1,k_2,\ldots,k_n}$ is the parameter of MBP.

CMBP and RPN, which both bridge rule-based and statistical approaches, are based on MBP.

5.1.2 Definition of CMBP

MBP gives a common form for rule-based and statistical generative Bayesian models. CMBP we proposed in the work of [10, 36] is a data-driven approach using constraints and prior knowledge to find a good model of this common form.

Similar to work proposed in Ref. [35], slot and value are assumed to be independent, though CMBP is not limited to the assumptions.

More probabilistic features are used in CMBP as below to allow CMBP to model complex cases:

- $P_t^+(v)$: 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;
- $\tilde{P}_t^+(v) = \sum_{v' \notin \{v, \text{None}\}} P_t^+(v')$;
- $\tilde{P}_t^-(v) = \sum_{v' \notin \{v, \text{None}\}} P_t^-(v')$;
- $b_t(v)$: 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.

With the above probabilistic features, a CMBP model is defined as

$$b_{t+1}(v) = \mathcal{P}\left(P_{t+1}^+(v), P_{t+1}^-(v), \tilde{P}_{t+1}^+(v), \tilde{P}_{t+1}^-(v), b_t^r, b_t(v)\right) \\ \text{s.t. constraints.} \tag{7}$$

where $\mathcal{P}(\cdot)$ is a multivariate polynomial function defined by Eq. (6).

CMBP uses intuition or prior knowledge by *constraints* in Eq. (7). They can be classified into three categories.

- **Probabilistic constraints** are used to restrict probabilistic features by definition, which can be directly written as linear equalities or inequalities. For example, according to the definition of b_t^r,

$$b_t^r = 1 - \sum_{v' \neq \text{None}} b_t(v'). \tag{8}$$

- **Intuition constraints** encode intuitive prior knowledge (i.e., rules). For example, the rule *"the belief should be unchanged or negatively correlated with the negative scores from SLU"* is represented by

$$\frac{\partial \mathcal{P}(P_{t+1}^{+}(v), P_{t+1}^{-}(v), \tilde{P}_{t+1}^{+}(v), \tilde{P}_{t+1}^{-}(v), b_t^r, b_t(v))}{\partial P_{t+1}^{-}(v)} \leqslant 0. \tag{9}$$

- **Regularization constraints** attempt to regularise the solution to prevent overfitting in the data-driven rule generation. For example, the coefficients of $\mathcal{P}(\cdot)$ may be limited to be in $[-1, 1]$.

To build an easily solvable model, these constraints in mathematical forms should further be relaxed to linear forms to get a constrained optimisation problem.

For example, Eq. (9) can be approximated by the linear constraint shown below

$$\left\{ \mathcal{P}(\mathbf{a}) \leqslant \mathcal{P}(\mathbf{b}) \middle| \begin{array}{l} \mathbf{a}, \mathbf{b} \in \chi, a_i \in T_5 \\ a_2 = b_2 + 0.1, \quad a_i = b_i \quad \forall i \neq 2 \end{array} \right\}, \tag{10}$$

where $\mathbf{a}$ and $\mathbf{b}$ are the 6-dimensional input vectors of Eq. (7), χ denotes all possible input vectors and $T_5 = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ is quantised interval of $[0, 1]$.

5.1.3 Data-driven rule generation for CMBP

Once rule-based model is formulated as CMBP, good rules are defined by constraints that are indicated by prior knowledge. Data-driven approach is further used to refine CMBP. By utilizing data, CMBP is an evolvable tracker which overcomes the inferior performance weakness of normal rule-based model. The model complexity is indicated by the order of CMBP. Order $n = 3$ is used in Refs. [10, 36] which yields state-of-the-art results. Then the Eq. (6) is

$$\mathcal{P}(x_1, x_2, \ldots, x_6) = \sum_{0 \leqslant k_1 \leqslant k_2 \leqslant k_3 \leqslant 6} g_{k_1,k_2,k_3} \prod_{1 \leqslant i \leqslant 3} x_{k_i}, \tag{11}$$

where $x_0 = 1$ and $g \in \mathbb{Z}$ are the CMBP parameters.

Aiming at improving the overall goal tracking accuracy of training data, the data-driven CMBP can be formulated as the following optimization problem:

$$\max \mathcal{L}(\mathbf{w}) = \sum_{s=1}^{S} \mathrm{Acc}\left(\mathcal{P}(x_1^{(s)}, x_2^{(s)}, \ldots, x_6^{(s)}; \mathbf{g})\right)$$
$$\text{s.t. approximated linear constraints.} \tag{12}$$

where $\mathbf{g} = \{g_{000}, g_{001}, \ldots, g_{666}\}$, S is the total number of turns of the training samples, $\mathrm{Acc}(\cdot)$ is the accuracy of state tracking.

The steps of finding the optimal CMBP are shown below given the formulated optimisation problem (Eq. (12)):

1) A superset of feasible CMBPs satisfying the approximated linear constraints is generated. This generation can be done by an algorithm of *integer linear programming* with objective function being dummy. Existing solver, such as SCIP [37], can be used for this purpose. The size of the superset can be neither too small, nor too big by setting additional constraints.
2) Each feasible solution from step 1 is enumerated and the corresponding $\mathcal{L}(\mathbf{g})$ is calculated. It is possible that some $b_t(v)$ or b_t^r are out of $[0, 1]$ because of constraints approximated. To get legal track output, $b_t(v)$ that is less than 0 is set to 0, and $b_t(v)$ that is larger than 1 is set to 1. b_t^r is re-calculated with legal $b_t(v)$.
3) The best CMBP is found by the one with highest accuracy. Regularization terms and other additional selection criteria, can also be used here.

The optimal integer-coefficient CMBP can be found with the above process. With more training data, the optimal CMBP can be further refined.

Bayesian probability operation only involves integer coefficient. Although CMBP is inspired by Bayes' theorem originally, CMBP as a statistical approach can be extended to *real* coefficient. An integer solution is obtained first, then hill climbing is used to extend the integer-coefficient solution to real-coefficient solution [10].

CMBP bridges rule-based approaches and statistical approaches effectively. The intuition prior knowledge is encoded by constraints which can be set manually, while data-driven optimisation of model parameters is possible in optimising general Bayesian polynomial representation.

5.2 Recurrent polynomial network

As another approach to bridge rule-based and statistical models, the basic idea of RPN [11] is to enable a kind of statistical model to utilize prior knowledge or intuition by using the parameters of rule-based models to initialize the parameters of statistical models. By taking into account of prior knowledge, RPN can achieve good performances even if training data is insufficient, which are shown in experiments.

An RPN, as a computational network, contains multiple edges and loops. Like an RNN, each node at time t in RPN takes values from nodes at time t and nodes at time $t-1$. Formally, there are two types of edges, $type-1$ and $type-2$ edges. A $type-1$ edge indicates that a node at time t takes value from a node at time $t-1$, while a node at time t takes the value of a value at time t when they are connected by a $type-2$ edge.

Let I_x be the set of nodes which are connected to node x using $type-1$ edges, and $\hat{I}_x$ be the set of nodes which are connected to node x using $type-2$ edges. Similarly, let w_{xy} and $\hat{w}_{xy}$ denote the weight of $type-1$ edge $\overrightarrow{yx}$ and $type-2$ edge $\overrightarrow{yx}$ respectively.

Generally, two types of nodes are used in RPN, *input nodes* and *computation nodes*. Input nodes are used to input features at each time. Computation nodes are further classified into three categories: sum node, product node and activation node. For node x at time t, the evaluation of nodes differ according to their types. Let $u_x^{(t)}$ denote the value of node x at time t.

- **Sum node** For a sum node x at time t, its value is the weighted sum of nodes linked to it.

$$u_x^{(t)} = \sum_{y \in I_x} w_{x,y} u_y^{(t-1)} + \sum_{y \in \hat{I}_x} \hat{w}_{x,y} u_y^{(t)}. \tag{13}$$

- **Product node** For a product node x at time t, its value is the product of its inputs. However, there may be some nodes that are linked to node x multiple times. Then these values should be multiplied to $u_x^{(t)}$ multiple times. Let M_{xy} and $\hat{M}_{xy}$ denote the multiplicity of the $type-1$ and $type-2$ edge $\overrightarrow{yx}$ respectively. Then $u_x^{(t)}$ is evaluated by

$$u_x^{(t)} = \prod_{y \in I_x} u_y^{(t-1) M_{x,y}} \prod_{y \in \hat{I}_x} u_y^{(t) \hat{M}_{x,y}}. \tag{14}$$

- **Activation node** Activation nodes only take one input and only has one input edge of type-2, i.e $|\hat{I}_x| = 1$ and $I_x = \emptyset$. The value of an activation node x is calculated as

$$u_x^{(t)} = softclip\left(u_{J_x}^{(t)}\right), \tag{15}$$

where J_x denotes the input node of node x. i.e. $\hat{I}_x = \{J_x\}$. *Softclip* will be detailed later.

Note that $M_{x,y}$ and $\hat{M}_{x,y}$ are indicated by the structure of RPN. And parameters of RPN only include $w, \hat{w}$. Each computation node can be regarded as an output node.

5.2.1 Basic structure for DST

With the sum nodes and product nodes introduced above, a polynomial can be easily computed. Because CMBP uses a homogeneous polynomial, it can be computed by a simple layered RPN. An RPN expressing a 3-order CMBP is shown in Fig. 3.

Generally, there is a product layer and a sum layer besides the input layer, which corresponds to monomials and polynomial in CMBP. Specifically, Let (l, i) denote the index of ith node in the lth layer.

- **First layer / Input layer** Features used in CMBP at turn t are used to assign input nodes' value.

$$-u_{(1,0)}^{(t)} = b_{t-1};$$

$$-u_{(1,1)}^{(t)} = P_t^+;$$

$$-u_{(1,2)}^{(t)} = P_t^-;$$

$$-u_{(1,3)}^{(t)} = \tilde{P}_t^+;$$

$$-u_{(1,4)}^{(t)} = \tilde{P}_t^-;$$

$$-u_{(1,5)}^{(t)} = 1.$$

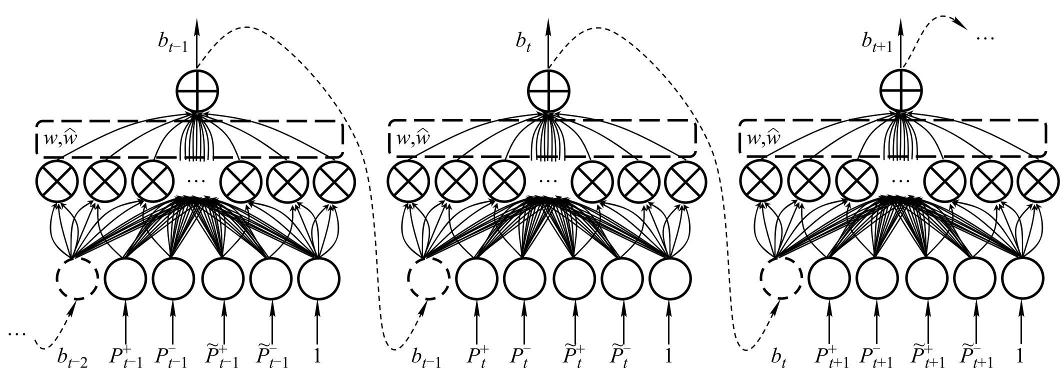


Fig. 3 RPN for DST.

Since preliminary experiments show the performance of CMBP would not degrade without feature b^r_{t-1}, to make the structure clearer and more compact, b^r_{t-1} is not used for RPN and CMBP in our work [11].

- **Second layer** Each monomial in CMBP is a value of some product node in the second layer. Since the order of CMBP is 3, every monomial in CMBP is the product of three repeatable features. Similarly, the value of every product node in second layer is the product of values of three repeatable nodes in the first layer. A product node $x = (2, i)$ is created by exhaustively enumerating every triple $(k_1, k_2, k_3)(0 \leqslant k_1 \leqslant k_2 \leqslant k_3 \leqslant 5)$. Nodes $(1, k_1), (1, k_2), (1, k_3)$ are linked to node x. And thus $u^{(t)}_x = u^{(t)}_{(1,k_1)} u^{(t)}_{(1,k_2)} u^{(t)}_{(1,k_3)}$. Different nodes in the second layer are created by distinct triples. To simplify the notation, a bijection to map nodes to their corresponding monomials is defined as:

$$\mathcal{F} : \{x | x \text{ is the index of a node in the second layer}\} \rightarrow \{(k_1, k_2, k_3) \mid 0 \leqslant k_1 \leqslant k_2 \leqslant k_3 \leqslant 5\}, \tag{16}$$

$$\mathcal{F}(x) = (k_1, k_2, k_3) \Longleftrightarrow u^{(t)}_{2,i} = u^{(t)}_{1,k_1} u^{(t)}_{1,k_2} u^{(t)}_{1,k_3}. \tag{17}$$

- **Third layer / Output layer** The value of sum node $x = (3, 0)$ in the third layer is the value of CMBP, i.e., b_t. Every product nodes in the second layer are linked to it. Therefore, its value is a weighted sum of values of product node $u^{(t)}_{2,i}$ where the weights correspond to g_{k_1,k_2,k_3} in Eq. (11).

5.2.2 Activation function

The values of product nodes and sum nodes are not guaranteed to lie in certain interval without constraints of input values and weights.

However, a belief is a possibility value which should lie in $[0, 1]$. Then the output value in RPN should be in $[0, 1]$, too. Besides, experiments show that if weights are not properly set in RPN, b_t may grow to a very large number. Activation functions are introduced to map the output value to a legal belief value in $[0, 1]$.

With several functions investigated [11], an activation function $softclip(\cdot)$, which is a combination of logistic function and clip function, is introduced. Let ϵ denote a small value such as 0.01, δ denote the offset of sigmoid function s.t. $sigmoid(\epsilon - 0.5 + \delta) = \epsilon$, where

$$sigmoid(x) = \frac{1}{1 + e^{-x}}. \tag{18}$$

Formally, the softclip function is defined as

$$softclip(x) \triangleq \begin{cases} sigmoid(x - 0.5 + \delta), & \text{if } x \leqslant \epsilon; \\ x, & \text{if } \epsilon < x < 1 - \epsilon; \\ sigmoid(x - 0.5 - \delta), & \text{if } x \geqslant 1 - \epsilon. \end{cases} \tag{19}$$

$softclip : \mathbb{R} \rightarrow (0, 1)$, which is linear on $[\epsilon, 1 - \epsilon]$, is a non-decreasing and continuous function.

5.2.3 Complex structure

Since RPN is a statistical model, it is easy to add new features and use complex structures. More importantly, by letting part of RPN expressing a CMBP, RPN can always achieve the performance of CMBP, no matter what new features and complex structure are implemented in RPN. Both new features and complex structure are explored in the work of [11].

When adding new features, new input nodes and product nodes should be added, which correspond to new features and new monomials. For slot s, value v at turn t, six features $f_0 \sim f_5$ used in previous structure are $b_{t-1}(v)$, $P^+_t(v)$, $P^-_t(v)$, $\tilde{P}^+_t(v)$, $\tilde{P}^-_t(v)$ and 1, respectively. Four new features have been investigated [11]. f_6 and f_7 are features of system acts at the last turn:

- $f_6 \triangleq canthelp(s, t, v)$ =1 if the constraints including $s = v$ make the system unable to find a venue, otherwise 0.
- $f_7 \triangleq select(s, t, v)$ =1 if the user is asked to choose a value for slot s with option v, otherwise 0.

f_8 and f_9 are features of user acts at the current turn:

- $f_8 \triangleq inform(s, t, v) = 1$ if user informs slot s is v in some SLU hypotheses, otherwise 0.
- $f_9 \triangleq deny(s, t, v)$ =1 if user denies slot s is v in some SLU hypotheses, otherwise 0.

To characterize some properties across turns, a new sum node $x = (3, 1)$ in the third layer is introduced. This node is also used as an input in the next time. The content of the node is unknown, but it is the only value which is not supervised by the label and may help reduce the effect of inaccurate label.

The structure of the RPN with new activation nodes, one new sum node and four new features is shown in Fig. 4.

5.2.4 RPN initialization

By using CMBP to initialize RPN, it takes advantage of prior knowledge or constraints effectively. More specifically, given

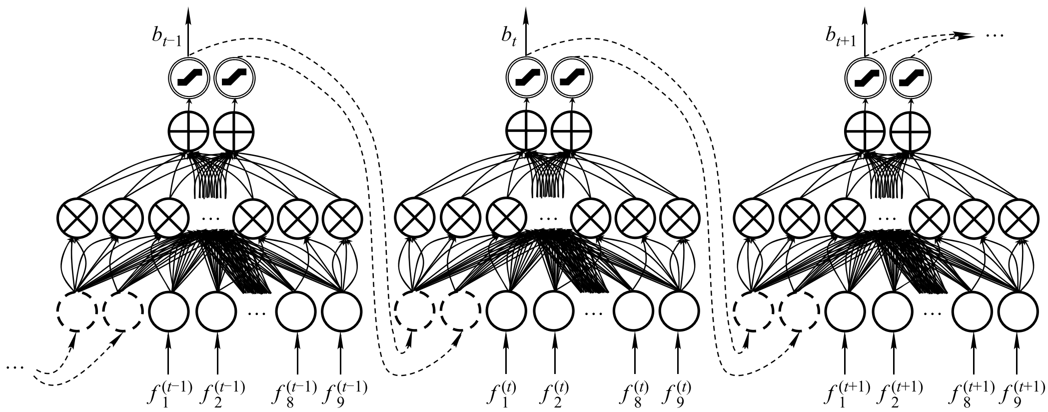


Fig. 4 RPN with new features and more complex structure for DST

a CMBP, part of RPN can be made identical to CMBP.

CMBP only utilizes $f_0 \sim f_5$. In structure shown in Fig. 4, if a product node is a monomial in CMBP, its weights should be set according to coefficient of this monomial in CMBP. Formally, for product node y, $\mathcal{F}(y) = (k_1, k_2, k_3)$ is defined in Eq. (16). If $0 \leqslant k_1 \leqslant k_2 \leqslant k_3 \leqslant 5$ is satisfied, weights $\hat{w}_{xy}$ should be initialized as g_{k_1,k_2,k_3} which is the coefficient of $f_{k_1} f_{k_2} f_{k_3}$ in CMBP. Other weights should be initialized as 0.

$$w_{x,y} = \begin{cases} g_{k_1,k_2,k_3}, & \text{if } x = (2,0) \text{ and } \mathcal{F}(x) = (k_1, k_2, k_3); \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

For RPN of other structures, prior knowledge and constraints in CMBP are used to find a suboptimum point in RPN's parameter space as the initial parameters.

5.3 Limitation of CMBP and RPN

Although the CMBP and the RPN approaches can effectively bridge rule-based and data-driven models, they both bear the assumption of goal and slot independence in this paper. It is possible to extend both CMBP and RPN to model multiple goals and slots simultaneously. However, straightforward joint goal/slot modelling will significantly increase the complexity of the models. Compared to state-of-the-art statistical tracker, such as RNN [32], it is not trivial to simultaneously model multiple goals for CMBP and RPN. This issue is to be addressed in the future.

6 Experiment

6.1 Evolvable SLU

The experiments of evolvable SLU in case of dialogue domain extension are conducted in DSTC-2/3 task, where DSTC-2 is the original domain (i.e., finding restaurant) with total 3235 dialogues and DSTC-3 is the extended domain (i.e., tourist information) with only 11 example dialogues available. In these experiments, the DSTC-3 evaluation corpus with 2264 dialogues was split into a train and test set again for method comparison. These datasets are:

- Seed: 11 labelled example dialogues.
- Real-train: *Seed* and 1144 dialogues of the DSTC-3.
- Real-test: other 1120 dialogues used for evaluation.

In Table 2, *Real-train* is implemented on the Real-train training set, which is an ideal in-domain parser with the best F-score and ICE[1] of semantic dialogue act. *Seed-train* is carried out in all DSTC-2 data and Seed dataset without data generation. It gets the worst F-score and ICE. In contrast, *Simul-train* exploits ASR hypotheses simulation to generate more data of DSTC-3 than Seed-train, as well as higher F-score and lower ICE. SLU rescoring further improves Simul-train slightly, although the gain is not significant.

Table 2 Performances of different SLU parsers in *Real-test*

method	Prec.	Recall	F-score	ICE
Real-train	0.914	0.818	0.863	1.060
Seed-train	0.858	0.764	0.808	1.734
Simul-train	0.870	0.799	0.833	1.483
+ rescore	0.871	0.801	0.834	1.483

6.2 Evolvable tracker

Evolvable mixed trackers taking advantage of prior knowledge and data-driven methods demonstrate their effectiveness and good performance under all situations in this section.

[1] Item cross entropy (ICE) between the N-best semantic hypotheses and the semantic label, assesses the overall quality of the semantic items distribution [24]. The lower, the better

Four trackers are investigated, including a baseline system using the HWU rule-based model [38], a statistical model using Max Entropy (MaxEnt) [28] and two mixed-type trackers: CMBP [10, 36] and RPN [11].

6.2.1 Effect of training data amount

Large amount of training data is crucial for building statistical trackers. However, as a matter of fact, cases with insufficient training data frequently occur because it is hard to get data and label them.

In this section, performances of different trackers with different amount of training data are investigated. Both the DSTC-3 sample training data (10 dialogues) and the DSTC-3 development data are used to train RPN and MaxEnt. When using only part of the data in these two dataset, the tracker's accuracy drop. The results are show in shown in Fig. 5, with X-axis being the fraction of data used, Y-axis being trackers' accuracy.

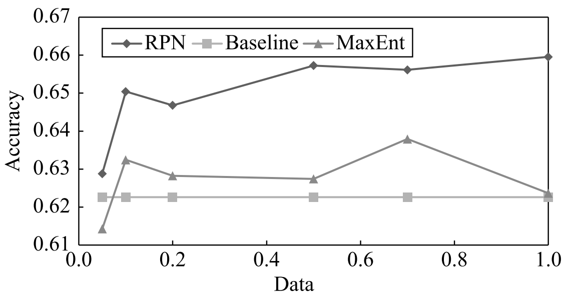


Fig. 5 Tracking with different amount of training data

It can be seen that RPN outperforms both rule-based (baseline HWU rule) and statistical model (MaxEnt) with all kinds of data amounts. This shows the advantage of incorporating prior knowledge when data is small, and the ability of evolvement when data amount is increasing.

6.2.2 SLU robustness

In an end-to-end system, performances of training SLU and testing SLU are often different, because it is very hard to get a reliable SLU applicable to all utterance patterns with limited training data and extended domain.

However, SLU confidence scores are usually used as direct inputs for DST and affect performance of state tracking greatly. The organiser-provided live SLU confidence have been shown to be poor [26, 28]. So most of the good trackers reported in DSTC-2 and DSTC-3 use their own SLU [28, 32, 33, 36, 39, 40]. Accuracy of Kadlec et al. [40]'s tracker increases 7.6% when organiser-provided SLU is replaced by their own refined SLU. Hence, mismatched SLU is a main problem in dialogue state tracking.

Four types of SLUs with different levels of performance are used to investigate this effect:

- Original: the rule-based parser provided by the DSTC-3 organizer.
- Train: a statistical parser trained with k percent error simulated data. SLU results are given by parsing on ASR-hypotheses.
- Combined: averaging the SLU results of the Original and the Train SLU.
- Transcript: similar to the Train type parser, the parser is trained with k percent error simulated data. However, to get an oracle setting for comparison, SLU results are parsed on transcription instead of ASR hypothesis.

The six SLUs used for comparison are shown in Table 3.

Table 3 Performance of the six different SLUs

SLU type	ASR error/%	ICE	Fscore	Precision	Recall
Original	0	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

To simulate the condition with gap between training SLU and testing SLU, in experiments shown in Fig. 6, the SLU for all trackers is fixed during training and the above six SLUs are used for testing respectively. In Fig. 6, the X-axis is the SLU ICE (the lower the better), and the Y-axis is the tracking accuracy on DSTC3-test. RPN achieves best results with all SLUs, which demonstrate the power of evolvable trackers.

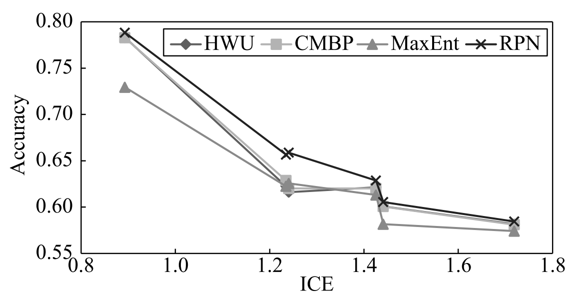


Fig. 6 Tracking with mismatched SLU

Trackers with consistent SLUs are also tested. In Fig. 7, trackers are trained and tested on the SLU output from the same parser. RPN also outperforms all other approaches. This shows that the best mixed-type tracker is robust to SLUs.

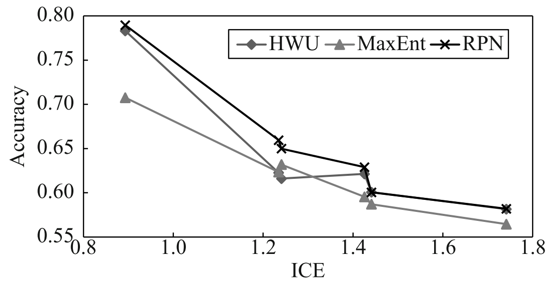


Fig. 7 Tracking with matched SLU

6.2.3 Comparison between state-of-the-art trackers

In this section, the mixed-type trackers with the enhanced semantic parser are compared to state-of-the-art trackers on the DSTC-3 challenge task. Note that, in DSTC-3, only ten dialogues of the new domain are provided for training. Both rule-based and mixed-type trackers do not need to use the ten training dialogues, since they can both be regarded as domain independent rules. On the other hand, mixed-type trackers are refined using the data of DSTC-2, i.e., another related domain with sufficient data.

Firstly, the performances of rule-based, statistical and mixed-type trackers using similar feature set (six features introduced before), i.e., the performances of CMBP, RPN, two rule-based trackers and two statistical trackers are compared in Table 4. Altogether, two rule-based trackers and two statistical trackers were built for performance comparison. Both accuracy and Brier score (L2) are employed as evaluation metrics.

Table 4 Performance comparison among RPN, CMBP and other models

Type	System	dstc2eval		dstc3eval	
		Acc	L2	Acc	L2
Rule	MaxConf	0.668	0.647	0.548	0.861
	HWU	0.720	0.445	0.594	0.570
Statistical	DNN	0.719	0.469	0.628	0.556
	MaxEnt	0.710	0.431	0.607	0.563
Mixed	CMBP	0.756	0.370	0.628	0.546
	RPN	0.758	0.370	0.644	0.542

Note that the performance of CMBP in the table is the initialisation parameter for RPN. It can be seen that the mixed-type trackers outperform both rule-based and statistical trackers with the same feature set. It may be argued that the performance of the statistical models with the limited feature set is not the state-of-the-art. To address this issue, we compare the mixed-type trackers to the best trackers in DSTCs.

Table 5 shows the results of DSTC-2, where sufficient training data is available. It can be observed that, although statistical model performs best in this case, RPN and CMBP's performance is still competitive compared to the best submitted trackers in DSTC-2. Note that Baseline* is the best one from the four baselines in DSTC-2 and in DSTC-3, Williams [33]'s system employed batch ASR hypothesis (i.e., off-line ASR re-decoded results) and cannot be used in the normal online model in practice. Hence, Henderson et al. [32] achieve the the best practical result. It can be seen that CMBP and RPN rank only second to the best practical tracker. Considering that only probabilistic features and the very limited added features are used, they are quite competitive in performance and can operate very efficiently.

Table 5 Performance comparison among trackers of DSTC-2 on dstc2eval

System	Approach	Rank	Acc	L2
Baseline*	Rule	5	0.719	0.464
Williams (2014) [33]	LambdaMART	1	0.784	0.735
Henderson et al. (2014d) [32]	RNN	2	0.768	0.346
Sun et al. (2014b) [28]	DNN	3	0.750	0.416
Yu et al. (2015) [10]	Real CMBP	2.5	0.762	0.436
RPN	RPN	2.5	0.756	0.372

The above models are using the same feature set. It is also of interest to compare the evolvable mixed-type trackers to the state-of-the-art submissions in DSTC-3.

Table 6 Performance comparison among trackers of DSTC-3 on dstc3eval

System	Approach	Rank	Acc	L2
Baseline*	Rule	6	0.575	0.691
Henderson et al. [39]	RNN	1	0.646	0.538
Kadlec et al. [40]	Rule	2	0.630	0.627
Sun et al. [36]	Int CMBP	3	0.610	0.556
Yu et al. [10]	Real CMBP	1.5	0.634	0.579
RPN	RPN	0.5	0.650	0.538

It can be observed from Table 6, state-of-the-art performance on DSTC-3 is achieved by CMBP and RPN trained on DSTC-2 without modifying tracking method[2)]. RPN with enhanced semantic parser outperforms all the submitted trackers in DSTC-3 including the best submission (RNN system). This effectively demonstrates the evolution ability of mixed-type parsers.

7 Conclusions

This paper reviews dialogue state tracking for domain extension. We have reviewed evolvable methods for dialogue state tracking from two aspects: parser and tracker. Two novel trackers, CMBP and RPN, which bridge rule-based and sta-

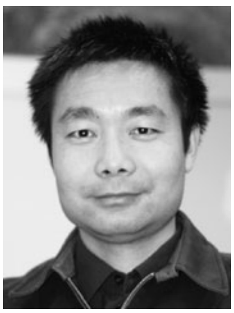
2) The parser is enhanced for DSTC-3 using the approach in Section 3 [26]

tistical approaches, are reviewed in detail. The experiments show that these methods can achieve state-of-the-art performance with sufficient data, and are particularly robust and effective for domain extension.

References

1. Williams J D, Young S. Partially observable Markov decision processes for spoken dialog systems. Computer Speech & Language, 2007, 21(2): 393–422
2. Thomson B, Young S. Bayesian update of dialogue state: a POMDP framework for spoken dialogue systems. Computer Speech & Language, 2010, 24(4): 562–588
3. Young S, Gašić M, Keizer S, Mairesse F, Schatzmann J, Thomson B, Yu K. The hidden information state model: a practical framework for POMDP-based spoken dialogue management. Computer Speech & Language, 2010, 24(2): 150–174
4. Young S, Gasic M, Thomson B, Williams J D. POMDP-based statistical spoken dialog systems: a review. Proceedings of the IEEE, 2013, 101(5): 1160–1179
5. Williams J D. Challenges and opportunities for state tracking in statistical spoken dialog systems: results from two public deployments. IEEE Journal of Selected Topics in Signal Processing, 2012, 6(8): 959–970
6. Williams J D. A belief tracking challenge task for spoken dialog systems. In: Proceedings of NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data. 2012, 23–24
7. Williams J, Raux A, Ramachandran D, Black A. The dialog state tracking challenge. In: Proceedings of the SIGDIAL 2013 Conference. 2013, 404–413
8. Henderson M, Thomson B, Williams J D. The second dialog state tracking challenge. In: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL). 2014, 263–272
9. Henderson M, Thomson B, Williams J D. The third dialog state tracking challenge. In: Proceedings of IEEE Spoken Language Technology Workshop (SLT). 2014
10. Yu K, Sun K, Chen L, Zhu S. Constrained markov bayesian polynomial for efficient dialogue state tracking. IEEE/ACM Transactions on Audio, Speech and Language Processing, 2015, 23(12): 2177–2188
11. Sun K, Xie Q, Yu K. Recurrent polynomial network for dialogue state tracking. 2015, arXiv preprint arXiv: 1507.03934
12. Kim D, Henderson M, Gasic M, Tsiakoulis P, Young S J. The use of discriminative belief tracking in pomdp-based dialogue systems. In: Proceedings of IEEE Spoken Language Technology Workshop (SLT). 2014, 354–359
13. Henderson M, Thomson B, Williams J. The second dialog state tracking challenge. In: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discowrse and Dialogue. 2014, 263
14. Young S. CUED standard dialogue acts. Report, Cambridge University, Engineering Department, 2007
15. He Y, Young S. Spoken language understanding using the hidden vector state model. Speech Communication, 2006, 48(3): 262–275
16. Wong Y W, Mooney R J. Learning synchronous grammars for semantic parsing with lambda calculus. Annual Meeting-Association for Computational Linguistics. 2007, 45(1): 960
17. Zettlemoyer L S, Collins M. Online learning of relaxed CCG grammars for parsing to logical form. In: Proceedings of the Joint Conference on Emperical Methods in Natural Language Processing and Computational Natural Language Learning. 2007, 878–887
18. Sha F, Pereira F. Shallow parsing with conditional random fields. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. 2003, 134–141
19. Jurcıcek F, Mairesse F, Gašic M, Keizer S, Thomson B, Yu K, Young S. Transformation-based learning for semantic parsing. In: Proceedings of INTERSPEECH. 2009, 2719–2722
20. Yao K, Zweig G, Hwang M Y, Shi Y, Yu D. Recurrent neural networks for language understanding. In: Proceedings of INTERSPEECH. 2013, 2524–2528
21. Yao K, Peng B, Zhang Y, Yu D, Zweig G, Shi Y. Spoken language understanding using long short-term memory neural networks. In: Proceedings of IEEE Spoken Language Technology Workshop. 2014, 189–194
22. Guo D Z, Tur G, Yih W T, Zweig G. Joint semantic utterance classification and slot filling with recursive neural networks. In: Proceedings of IEEE Spoken Language Technology Workshop. 2014, 554–559
23. Mairesse F, Gasic M, Jurcícek F, Keizer S, Thomson B, Yu K, Young S. Spoken language understanding from unaligned data using discriminative classification models. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing. 2009
24. Henderson M, Gasic M, Thomson B, Tsiakoulis P, Yu K, Young S. Discriminative spoken language understanding using word confusion networks. In: Proceedings of IEEE Spoken Language Technology Workshop. 2012, 176–181
25. Raymond C, Riccardi G. Generative and discriminative algorithms for spoken language understanding. In: Proceedings of INTERSPEECH. 2007, 1605–1608
26. Zhu S, Chen L, Sun K, Zheng D, Yu K. Semantic parser enhancement for dialogue domain extension with little data. In: Proceedings of IEEE Spoken Language Technology Workshop (SLT). 2014
27. Lee S, Eskenazi M. Recipe for building robust spoken dialog state trackers: dialog state tracking challenge system description. In: Proceedings of the SIGDIAL 2013 Conference. 2013, 414–422
28. Sun K, Chen L, Zhu S, Yu K. 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). 2014, 318–326
29. Lee S. Structured discriminative model for dialog state tracking. In: Proceedings of the SIGDIAL 2013 Conference. 2013, 442–451
30. Kim S, Banchs R. Sequential labeling for tracking dynamic dialog states. In: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue. 2014, 332
31. Henderson M, Thomson B, Young S. Deep neural network approach for the dialog state tracking challenge. In: Proceedings of the SIGDIAL 2013 Conference. 2013, 467–471
32. Henderson M, Thomson B, Young S. Word-based dialog state tracking with recurrent neural networks. In: Proceedings of the 15th An-

nual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL). 2014, 292–299
33. Williams J D. 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). 2014, 282–291
34. Zilka L, Marek D, Korvas M, Jurcicek F. Comparison of Bayesian discriminative and generative models for dialogue state tracking. In: Proceedings of the SIGDIAL 2013 Conference. 2013, 452–456
35. Wang Z, Lemon O. 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. 2013, 423–432
36. Sun K, Chen L, Zhu S, Yu K. A generalized rule based tracker for dialogue state tracking. In: Proceedings of IEEE Spoken Language Technology Workshop (SLT). 2014, 330–335
37. Achterberg T. SCIP: Solving constraint integer programs. Mathematical Programming Computation, 2009, 1(1): 1–41
38. Wang Z. HWU baseline belief tracker for dstc 2 & 3. Technical Report. 2013
39. Henderson M, Thomson B, Young S. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In: Proceedings of IEEE Spoken Language Technology Workshop (SLT). 2014, 360–365
40. Kadlec R, Vodolan M, Libovicky J, Macek J, Kleindienst J. Knowledge-based dialog state tracking. In: Proceedings of IEEE Spoken Language Technology Workshop. 2014, 348–353



Kai Yu received his BE and MS from Tsinghua University, China in 1999 and 2002, respectively. He then obtained his PhD from Cambridge University, UK in 2006. He is currently a research professor at Shanghai Jiao Tong University, China. His main research interests include speech recognition, synthesis, language understanding and dialogue management. He was selected into the 1000 Overseas Talent Plan (Young Talent) by Chinese government and is the Awardee of the NSFC Excellent Young Scholars Program in 2012. He is a member of the Technical Committee of the Speech, Language, Music and Auditory Perception Branch of the Acoustic Society of China.

Lu Chen received his BS from the School of Computer Science and Technology, Huazhong University of Science & Technology, China in 2013. He is currently a PhD student in the SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include dialogue management and machine learning.

Kai Sun received his BS in computer science from Shanghai Jiao Tong University, China in 2015. Then he worked for his PhD in the Department of Computer Science at Cornell University, USA. His current research interests include speech and language processing, machine learning, and AI in games.

Qizhe Xie is currently a BS student in ACM class in Zhiyuan College, Shanghai Jiao Tong University, China. His research interests include dialogue state tracking, machine learning and natural language processing.

Su Zhu received his BS in computer science from Xi'an Jiao Tong University, China in 2013. He is currently a PhD student in computer science at Shanghai Jiao Tong University, China. His main research interests include language understanding and machine learning.