Constrained Markov Bayesian Polynomial for Efficient Dialogue State Tracking

Kai Yu, Kai Sun, Lu Chen, and Su Zhu

***Abstract*—Dialogue state tracking (DST) is a process to estimate the distribution of the dialogue states at each dialogue turn given the interaction history. Although data-driven statistical approaches are of most interest, there have been attempts of using rule-based methods for DST, due to their simplicity, efficiency and portability. However, the performance of these methods are usually not competitive to data-driven tracking approaches and it is not possible to improve the DST performance when training data are available. In this paper, a novel hybrid framework, *constrained Markov Bayesian polynomial* (CMBP), is proposed to formulate rule-based DST in a general way and allow data-driven rule generation. Here, a DST rule is defined as a polynomial function of a set of probabilities satisfying certain linear constraints. Prior knowledge is encoded in these constraints. Under reasonable assumptions, CMBP optimization can be converted to a constrained integer linear programming problem. The integer coefficient CMBP model is further extended to CMBP with real coefficients by applying grid search. CMBP was evaluated on the data corpora of the first, the second, and the third *Dialog State Tracking Challenge* (DSTC-1/2/3). Experiments showed that CMBP has good generalization ability and can significantly outperform both traditional rule-based approaches and data-driven statistical approaches with similar feature set. Compared with the state-of-the-art statistical DST approaches with much richer features, CMBP is also competitive.**

***Index Terms*—Data-driven rule, dialogue state tracking (DST), rule-based model, statistical dialogue management.**

I. Introduction

A TASK-ORIENTED *spoken dialogue system* (SDS) 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 semantic-level user dialogue acts are extracted from acoustic speech signals. With the input user dialogue acts, the control module, also called *dialogue management* accomplishes two functions. One is to maintain its internal state, an encoding of the machine's understanding about the conversation. As information is received from the input module, the state is updated, which is called *dialogue state tracking* (DST). Another is to choose a machine action, also at semantic-level, to direct the dialogue given the information of the dialogue state, referred to as *dialogue decision making*. The output consists of natural language generation (NLG) and text-to-speech (TTS) synthesis, with which machine dialogue acts are converted to audio.

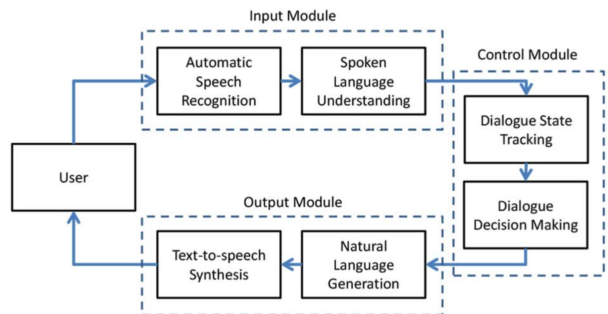

Fig. 1. Diagram of a spoken dialogue system (SDS).

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. Since the current dialogue state can be observed, dialogue state tracking is trivial and there is no need to estimate the state distribution. 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 behavior, inevitable automatic speech recognition and spoken language understanding errors make it difficult to maintain the true dialogue state and make decision [1]. 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) framework [1], [2], [3], [4]. In most studies of POMDP, both dialogue state tracking and decision making are modelled using data-driven 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 *dialog state tracking challenge* (DSTC) provides the first common testbed in a standard format, along with a suite of evaluation metrics for this purpose [5].

Since the DST problem is raised out of the statistical dialogue management framework, data-driven statistical approaches have been the natural focus. Most early works of POMDP concentrate on generative Bayesian models [4]. Fundamental

Manuscript received February 12, 2015; revised July 17, 2015; accepted August 12, 2015. Date of publication August 20, 2015; date of current version September 04, 2015. This work was supported by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning and the China NSFC project No. 61222208. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tatsuya Kawahara.

K. Yu, L. Chen, and S. Zhu are with the Computer Science and Engineering Department, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: kai.yu@sjtu.edu.cn; chenluhust@gmail.com; zhusu.china@gmail.com).

K. Sun is with the Department of Computer Science, Cornell University, Ithaca, NY 14853 USA (e-mail: ks985@cornell.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2015.2470597

weaknesses of generative model are revealed by the results of [6]. In contrast, discriminative state tracking models have been successfully used for spoken dialogue systems [7]. The results of the DSTC [5] further demonstrate the power of discriminative statistical models, such as Maximum Entropy [8], Conditional Random Field [9], Deep Neural Network (DNN) [10], and Recurrent Neural Network (RNN) [11]. However, data-driven statistical approaches have also shown large variation in performance and poor generalization ability due to the lack of data [6]. There also has been an attempt to employ rule-based methods for DST due to its simplicity, efficiency, portability and interpretability. For example, the baseline of DSTC just employs a simple rule of selecting the SLU result with the highest confidence score so far and discarding alternatives [5], [12]. More complex rules can also compute scores for multiple hypotheses [13], [14], [15]. These rule-based methods have shown amazingly good performance and generalization ability, even beating most data-driven statistical approaches with complex model structure and features. Since these rule based models directly incorporate prior knowledge or intuition into DST, they are easy to understand and interpret. Nevertheless, existing rule-based methods are usually not competitive to the best statistical approaches. Moreover, there is no general way to design rule based models with prior knowledge and it is not possible to improve their performance when training data are available.

In this paper, a novel hybrid framework, referred to as *constrained Markov Bayesian polynomial* (CMBP), is proposed to formulate rule-based DST in a general way and allow data-driven rule generation. It bridges the gap between rule-based and data-driven approaches. In the CMBP framework, a DST *rule* is defined as an integer-coefficient polynomial function of a set of probabilities, satisfying certain linear constraints. Prior knowledge is encoded into the constraints. Under reasonable assumptions, valid rule-based model candidates can be generated using integer linear programming. The valid model candidates, i.e. the generated polynomial functions, are then evaluated on training data to select the optimal one. As it is easy to generate multiple CMBPs and model combination is usually effective [8], [10], [11], [12], [16], score averaging based combination can also be easily implemented for CMBP. Furthermore, CMBP is extended to real-coefficient polynomial in this paper, where the real coefficients can be estimated by optimizing the DST performance on training data using grid search. This extension bridges rule-based methods and data-driven approaches.

The rest of the paper is organized as follows. Section II formulates the DST problem. The Constrained Markov Bayesian Polynomial framework and its optimization is discussed in Section III, followed by experiments in Section IV. Section V concludes the paper. Details of CMBP constraints and search space complexity are included in the Appendix.

II. Dialogue State Tracking

In statistical dialogue management, there are two key stages: dialogue state tracking (DST) and decision making. DST estimates the distribution over all possible dialogue states at a particular dialogue turn given the interaction history. The state distribution is usually referred to as *belief state* and can be regarded as *sufficient statistics* summarizing the initial belief state $b_0(\mathbf{s})$, all previously taken machine actions a and received observations $\mathbf{o}$, i.e.

$$b_t(\mathbf{s}) = P(\mathbf{s}_t | b_0(\mathbf{s}), a_0, \mathbf{o}_1, a_1, \mathbf{o}_2, \cdots, a_t, \mathbf{o}_t) \quad (1)$$

where the " *observation*" here is defined as the acoustic signal input to the SDS for the convenience of derivation.

A. Dialogue State Definition and Approximation

The state $\mathbf{s}$ in SDS is usually factorized into three distinct components [4]: the user's goal $\mathbf{g}$, the user's action $\mathbf{u}$ and the dialogue history $\mathbf{h}$, i.e.

$$\mathbf{s} = (\mathbf{g}, \mathbf{u}, \mathbf{h}) \quad (2)$$

Here, $\mathbf{g}$ and $\mathbf{u}$ can both be represented in the form of dialogue act [17]

$$d := t(l = v) \quad (3)$$

where d denotes a dialogue act, t is the type of the act such as *inform*, *deny*, etc., l denotes a semantic slot and v is the value of the slot. Although it is possible to have multiple slot-value pairs for one dialogue act, there is a preference to split it into several dialogue acts with the canonical form (3) [6]. The dialogue history component $\mathbf{h}$ is a discrete grounding state to denote the status of the slot-value pairs mentioned by the user and the system in various contexts. For example, the dialogue history state can be *user-inform*, *system-query*, *user-deny* etc.

Although the dialogue state factorization can reduce complexity, the real world SDS state space is still huge and intractable. Therefore, belief state approximation is necessary for real world SDS systems. The hidden information state (HIS) framework uses an N-best approximation, whereby the probabilities of all states are ranked and pruned to retain only the N most likely state partitions [4]. The Bayesian Update of Dialogue State (BUDS) framework further factorizes the state distribution by introducing slot independence assumptions, resulting in a Bayesian network representation for dialogue states [2]. The exact form of dialogue state approximation highly depends on tasks and naturally affects DST approaches.

B. Dialog State Tracking Challenge

In early works of statistical DM research, dialogue state tracking and policy optimization are usually studied together. Although various DST approaches have been proposed, they cannot be directly compared due to task difference. To address this issue, *Dialog State Tracking Challenge* (DSTC) is organized to provide shared tasks for comparing DST algorithms [5]. Three DSTCs have been organized so far, all of which employed slot-filling tasks and static dialogue corpora. In these tasks, the dialogue state of interest is only the "user's goal" $\mathbf{g}$ in equation (2), which is represented by a canonical dialogue act with a fixed type "`find`". Hence, the target of DST here is to find the slot-value pairs which the user intend to pursue, from the dialogue interaction. Given the output of a state tracker in the form of joint distribution of the goals, a number of metrics have been developed for evaluation [5]. In DSTC-2 and DSTC-3, additional dialogue states of *method* and *requested slot* are also employed [18]. In this paper, for clarity, only *user's goal* is considered and the extension to *method* and *requested slot* is straightforward.

The "history state" $\mathbf{h}$ in the DSTC tasks is usually simplified as the grounding status of the received slot information. For a specific slot, $h \in \{-1, 0, 1\}$, where "0" denotes receiving neutral or no information, "-1" denotes receiving negative information (i.e. the user denies or the system negates a slot-value pair) and "1" denotes receiving positive information (i.e. the user informs or the system affirms a slot-value). With this "history state" definition, more flexible treatment of the observed information can be achieved.

In the DSTC tasks, multiple ASR and SLU hypotheses with normalized confidence scores are provided for each user turn. Hence, the direct input to a dialogue state tracker can be regarded as the distribution of the user act $p(\mathbf{u}|\mathbf{o})$. This input information is widely used in most DST approaches and also adopted in this paper. It is worth noting that the use of ASR hypotheses has been proved to be useful to improve the quality of $p(\mathbf{u}|\mathbf{o})$ and the DST performance [10], [11].

C. Generative Model for DST

In early work of POMDP, the belief state tracking is achieved by applying Bayesian rules as well as reasonable independence assumptions of the state components [4]:

$$b_{t+1}(\mathbf{s}) = \eta p(\mathbf{o}_{t+1}|\mathbf{u}_{t+1})P(\mathbf{u}_{t+1}|\mathbf{g}_{t+1}, a_t) \sum_{\mathbf{g}_t} P(\mathbf{g}_{t+1}|\mathbf{g}_t, a_t) \sum_{\mathbf{h}_t} P(\mathbf{h}_{t+1}|\mathbf{u}_{t+1}, \mathbf{g}_{t+1}, \mathbf{h}_t, a_t) b_t(\mathbf{g}, \mathbf{h}) \quad (4)$$

In the above equation, there are only two pieces of external input information: $p(\mathbf{o}_{t+1}|\mathbf{u}_{t+1})$ which is usually approximated by the estimated distribution (usually normalized confidence score) of the semantic hypotheses $q(\mathbf{u}_{t+1})$ and $b_t(\mathbf{g}, \mathbf{h}) = \sum_{\mathbf{u}_t} b_t(\mathbf{s})$ of turn t. The rest are all model parameters: $\eta = p(\mathbf{o}_{t+1}|a_t, b_t(\mathbf{s}))$ is a constant independent of $\mathbf{s}_t$, $P(\mathbf{g}_{t+1}|\mathbf{g}_t, a_t)$ is the *user goal model*, $P(\mathbf{u}_{t+1}|\mathbf{g}_{t+1}, a_t)$ is the *user action model* and $P(\mathbf{h}_{t+1}|\mathbf{u}_{t+1}, \mathbf{g}_{t+1}, \mathbf{h}_t, a_t)$ is the *dialogue history model*.

The dialogue history model is usually deterministic and simply measures the consistency between the updated dialogue state and the original dialogue state (e.g. if a goal is denied, h is set as -1). Parameters of the other two models need to be estimated using training data separately on static corpora or optimized jointly together with dialogue policy using reinforcement learning. Hence the generative Bayesian belief estimator is regarded as a data-driven statistical DST model. It has also been applied to DSTC by concentrating only on the goal component $\mathbf{g}$ [19], but did not yield competitive result due to inaccurate estimation of the parameters.

D. Discriminative Model for DST

Since the DST problem is inherently a classification task, various discriminative models have been proposed. They directly aim at optimizing the classification accuracy and can employ rich input features in addition to $p(\mathbf{o}_{t+1}|\mathbf{u}_{t+1})$ and $b_t(\mathbf{g}, \mathbf{h})$, which lead to the state-of-the-art performance. According to different statistical independence assumptions, there are four categories.

In *binary classifier*, all slots are assumed to be independent of each other, leading to an efficient goal state factorization:

$$b_t(\mathbf{g}) = \prod_i b_t(g_i) \quad (5)$$

where g_i is the dialogue state for slot i[1]. With this assumption, to get the joint goal, we just need to calculate the belief $b(g = v)$ for each slot g and candidate value v. This can be converted into a binary classification problem of determining whether $g = v$ is true or false. To reduce the number of required binary classifiers, value v can be encoded into input features, resulting in one binary classifier per slot. Various models have been used within this framework, such as MaxEnt [8], [10] and DNN [10], [20].

When binary classifiers are used, the relationship between values is not modelled. To address this issue, a *multi-classifier* is used to track the belief of all values simultaneously. Since it is usually not possible to get sufficient data samples for *every* candidate value, approximations must be used in multi-classifier. Recurrent Neural Networks (RNN) [11], [21] has been used within this framework.

Both of the above models hold the slot independence assumption (5). *Structured classifiers* are proposed to capture the relationship between slots *at a particular turn*. A typical example is conditional random field (CRF) with manually designed factored graph [9]. When the relational constrains are very complex, the design of the slot relationship structure is time-consuming and error-prone. In order to tackle this problem, decision forest model, which can automatically build conjunctions of raw features, has been used [12].

Though structured classifiers utilize the relational constrains between different slots, they only focus on information of a single turn. It is considered useful to also capture the relationship between multiple turns. *Sequence labelling* model is proposed for this purpose. Within this framework, linear-chain CRF has been investigated to model the temporal relationship between slots [22].

Discriminative models have defined the state-of-the-art performance in the DSTCs. However, feature design is ad-hoc and it is hard to incorporate prior knowledge of dialogue interaction. Furthermore, the performance is highly dependent on the available training data. In DSTC-3, where the task is to adapt a state tracker to a new domain using only 10 sample dialogues, many discriminative models performed worse than rule based models.

E. Rule-based Model for DST

The above generative Bayesian and discriminative belief estimators are all data-driven statistical approaches. As an alternative to the data-driven framework, there have been attempts to employ rule-based methods for dialogue state tracking. For example, the baseline of DSTC just employs a simple rule of selecting the SLU hypothesis with the highest confidence score so far and discarding the rest [5], [12]. In DSTC-1, the simple rule-based system outperformed many discriminative models and was ranked the 5th in the joint goal tracking task on Test3. More complex rules have also been proposed to enhance the power of rule-based models. In rule-based models, slot independence is usually assumed. In the below discussions, for clarity,

[1]In the rest of the paper, in case of no confusion, the index i is omitted.

$b_t(v)$ will be used to denote the probability of a specific goal slot taking value v at the tth turn, i.e.

$$b_t(v) = P(g_t = v | b_0(g), a_0, \mathbf{o}_1, a_1, \mathbf{o}_2, \cdots, a_t, \mathbf{o}_t)$$

In [14], $b_t(v)$ is set as the overall probability of the goal being positively stated from the first turn up to the current turn. To calculate this, probability operation of random events is used. The occurrence of positive (user informs or system affirms) or negative (user negates or system denies) "slot-value pair", denoted as $P_t^+(v) = P(g_t = v, h = 1)$ and $P_t^-(v) = P(g_t = v, h = -1)$ respectively, are regarded as random events and assumed to be independent across turns. Then, the goal belief can be updated as below [14]

$$\begin{aligned} b_{t+1}(v) &= \left(1 - (1 - b_t(v))\left(1 - P_{t+1}^+(v)\right)\right)\left(1 - P_{t+1}^-(v)\right) \\ &= P_{t+1}^+(v) + b_t(v) - P_{t+1}^+(v) b_t(v) - P_{t+1}^+(v) P_{t+1}^-(v) \\ &\quad - P_{t+1}^-(v) b_t(v) + P_{t+1}^+(v) P_{t+1}^-(v) b_t(v) \end{aligned} \quad (6)$$

From the above formula, state tracking is very efficient as it is only linear combination of the observations (the probabilities) and the belief of the previous turn. This improved rule-based model has outperformed most trackers in DSTC-1 (ranked the 5th/2nd/2nd/6th on Test1-4 respectively) and has been used as a strong baseline for DSTC-2 and DSTC-3.

Other rule-based models have also been proposed. In [13], the generative Bayesian DST model is employed but the parameters are set according to rules. In [23], system act is further introduced as a condition to determine rules under the Bayesian probability operation framework. These refined rule-based models have achieved very good tracking performance. However, most of them are still not competitive to data-driven statistical models. What's more, once the rule is set, they are not able to improve when more training data become available, hence lack the ability of evolution.

III. Constrained Markov Bayesian Polynomial

Rule-based models [13], [14], and Bayesian generative models [4] 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. The polynomial still describes the relationship between the belief state and relevant probabilities, hence it is an analogy of Bayes' theorem. The difference is that the polynomial coefficients may not be strictly derived from Bayes' theorem. To simplify the DST process, *Markov* assumption is also introduced. The belief state is assumed to be only dependent on the observations of the current turn and the belief state of the previous turn, rather than the entire history of observations and belief states. Hence, the general form is referred to as *Markov Bayesian Polynomial* (MBP):

$$b_{t+1}(\mathbf{s}) = \mathcal{P}(b_t(\mathbf{g}, \mathbf{h}), q(\mathbf{u}_{t+1})) \quad (7)$$

where $b_{t+1}(\mathbf{s})$ is the belief state of $\mathbf{s}$ at the $(t+1)$th turn, $q(\mathbf{u}_{t+1})$ is the estimated confidence distribution of the user act $\mathbf{u}_{t+1}$ and $\mathcal{P}(\cdot)$ is a multivariate polynomial function

$$\mathcal{P}(x_1, \cdots, x_D) = \sum_{0 \le k_1 \le \cdots \le k_n \le D} w_{k_1, \cdots, k_n} \prod_{1 \le i \le n} x_{k_i} \quad (8)$$

where $k \in \{0, 1, \cdots, D\}$, D is the number of input variables, $x_0 = 1$, n is the order of the polynomial. The scalar coefficient

$$w_{k_1, \cdots, k_n} = f_{k_1, \cdots, k_n}(\mathbf{s}_t, a_t, \mathbf{s}_{t+1})$$

is the *parameter* of MBP. In general, they can be viewed as a function of the interaction history.

It is noted that, the Bayesian generative model, equation (4), is a special case of MBP. Here, the input information is the semantic hypothesis distribution $q(\mathbf{u}_t)$ (the approximation of $p(\mathbf{o}_{t+1} | \mathbf{u}_{t+1})$) and a set of goal/history belief states $b_t(\mathbf{g}, \mathbf{h})$. The total dimension of the input is then $D = N_\mathbf{g} N_\mathbf{h} + 1$, where $N_\mathbf{g}$ and $N_\mathbf{h}$ are the numbers of all possible values of goal and history respectively.

Let l be the index for a goal-history pair, $l = (\mathbf{g}_t, \mathbf{h}_t)$, equation (4) can be rewritten as

$$b_{t+1}(\mathbf{s}) = \sum_l f_l(\mathbf{s}_t, a_t, \mathbf{s}_{t+1}) q(\mathbf{u}_{t+1}) b_t(\mathbf{g}, \mathbf{h}) \quad (9)$$

where the MBP parameters, $f_l(\mathbf{s}_t, a_t, \mathbf{s}_{t+1})$, correspond to the combination of various models described in Section II-C:

$$\begin{aligned} f_l(\mathbf{s}_t, a_t, \mathbf{s}_{t+1}) = \eta P(\mathbf{u}_{t+1} | \mathbf{g}_{t+1}, a_t) P(\mathbf{g}_{t+1} | \mathbf{g}_t, a_t) \\ P(\mathbf{h}_{t+1} | \mathbf{u}_{t+1}, \mathbf{g}_{t+1}, \mathbf{h}_t, a_t) \end{aligned} \quad (10)$$

It can be seen that the rewritten form, equation (9), is a second order MBP ($n = 2$) with $D = N_\mathbf{g} N_\mathbf{h} + 1$ input variables and the polynomial coefficients are defined by the Bayesian generative model parameters. Since these parameters can be estimated from data[2], the generative Bayesian belief estimator is regarded as a data-driven statistical DST model. It is worth noting that it is usually hard to get sufficient annotated data to estimate the parameters, hence, heuristics are usually used to directly optimize dialogue state tracking performance or the parameter update is performed together with the dialogue policy update within reinforcement learning framework [4].

Assuming that all slots are independent and only goal tracking is of interest, for a specific slot, rule-based models, e.g. equation (6), can also be written in a similar form of MBP

$$b_{t+1}(v) = \mathcal{P}\left(b_t(v), P_{t+1}^+(v), P_{t+1}^-(v)\right) \quad (11)$$

In contrast to the generative Bayesian model, all coefficients in (11) are manually set to be integers. Therefore, rule-based model can be viewed as an MBP with features of $b_t(v), P_t^+(v), P_t^-(v)$ and prior knowledge (i.e. rule) is incorporated by manually setting the integral polynomial coefficients.

A. Generalized Rule-based Model: Constrained MBP

MBP is a hybrid framework of rule-based and data-driven models. However, it does not provide a roadmap to bridge the two types of models. The key issue is how to use data-driven method to modify the coefficients of a rule-based model without losing the ability to incorporate prior knowledge. Here, a novel framework, constrained Markov Bayesian polynomial (CMBP) is proposed to address issue. The basic idea is to construct a constrained optimization problem for DST model training, where

[2]Except that the dialogue history model is usually manually set.

the model takes the form of MBP and the constraints encode all necessary probabilistic conditions as well as prior knowledge or intuition. In this paper, CMBP is derived as an extension of the rule-based model (6), hence slot and value independence are also assumed, though CMBP is not limited to the assumptions. To enhance the power of rule-based model, more probabilistic features are introduced into CMBP as below

- $P_t^+(v)$: 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^r: probability of the value being '`None`' (the value not mentioned) at turn t
- $b_t(v)$: belief of "the value being v at turn t"

With the above probabilistic features, a *Constrained Markov Bayesian Polynomial* (CMBP) model is defined as

$$b_{t+1}(v) = \mathcal{P}\left(P_{t+1}^+(v), P_{t+1}^-(v), \tilde{P}_{t+1}^+(v), \tilde{P}_{t+1}^-(v), b_t^r, b_t(v)\right) \quad \texttt{s.t. constraints} \tag{12}$$

The *constraints* in equation (12) can be classified into three categories.

- **Probabilistic constraints** enforce the probabilistic requirement by definition. These constraints can be directly written as a set of linear equality or inequalities, e.g.

$$b_t^r = 1 - \sum_{v' \neq \text{None}} b_t(v') \tag{13}$$

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

$$\frac{\partial \mathcal{P}(P_{t+1}^+(v), P_{t+1}^-(v), \tilde{P}_{t+1}^+(v), \tilde{P}_{t+1}^-(v), b_t^r, b_t(v))}{\partial P_{t+1}^+(v)} \geq 0 \tag{14}$$

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

Although constraints can be represented in mathematics forms, to construct a feasible constrained optimization problem, it is necessary to further approximate the constraints using linear equalities or inequalities. For example, inequality (14) can be approximated by the below linear constraint

$$\left\{ \mathcal{P}(\mathbf{a}) \geq \mathcal{P}(\mathbf{b}) \,\middle|\, \begin{array}{l} \mathbf{a}, \mathbf{b} \in \boldsymbol{\chi}, a_i \in T_5 \\ a_1 = b_1 + 0.1,\ a_i = b_i\ \forall i \neq 1 \end{array} \right\} \tag{15}$$

where $\mathbf{a}$ and $\mathbf{b}$ are the 6-dimensional input vectors of equation (12), $\boldsymbol{\chi}$ denotes all possible input vectors and $T_5 = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ denotes quantized interval of $[0, 1]$. Details of CMBP constraints and their corresponding linear approximations can be found in the appendix.

B. Data-driven Coefficient Optimization for CMBP

Once a rule-based model is formulated as CMBP, intuitive knowledge becomes *soft* constraints and there usually exist multiple feasible solutions. It is then possible to employ data-driven method to optimize CMBP. In CMBP, polynomial order n, as shown in equation (8), determines model complexity as well as the size of search space during optimization. Model complexity affects the modelling power and generalization ability of CMBP, while search space affects the computation time of optimization. Order $n = 1$ or $n = 2$ is too small to model complex situations, while $n \geq 4$ is too large to efficiently optimize. Hence, in this paper, polynomial order $n = 3$ is used to construct a suitable search space.[3] By using the overall goal tracking accuracy on the training data as the optimization criterion, the data-driven CMBP can be written as the below optimization problem

$$\begin{aligned} \max \quad & \mathcal{L}(\mathbf{w}) = \sum_{m=1}^{M} \texttt{Acc}\left(\mathcal{P}(x_1^{(m)}, \cdots, x_6^{(m)}; \mathbf{w})\right) \\ \texttt{s.t.} \quad & \texttt{approximated linear constraints} \end{aligned} \tag{16}$$

where $\mathbf{w} = \{w_{000}, w_{001}, \cdots, w_{666}\}$ and $w \in \mathbb{Z}$ are the CMBP parameters, M is the total number of turns of the training data, $\texttt{Acc}(\cdot)$ is the goal state tracking accuracy evaluation function. CMBP can then be optimized as below:

1) Generate a superset of all feasible CMBP solutions satisfying the approximated linear constraints.
   This generation can be regarded as a special kind of *integer linear programming* problem whose objective function is dummy. Existing integer linear programming solver can be used for this purpose. In this paper, SCIP [24] is used. By setting additional constraints or adding sparsity penalty term to the criterion, the size of this superset can be controlled so that it is neither too small, nor too large.
2) $\mathcal{L}(\mathbf{w})$ is exhaustively calculated for each feasible solution from step 1.
   During the optimization, due to relaxation of constraints, it is possible to get some $b_t(v)$ or b_t^r out of $[0, 1]$. To get legal track output, out-of-range $b_t(v)$ is always clipped to be 0 or 1 and b_t^r is re-calculated accordingly.
3) Find the optimal CMBP solution by selecting the one with highest accuracy. Additional regularization term, such as sparsity regularization, can also apply here.

With the above optimization, the best integer coefficient CMBP can be found and this optimal solution can be refined when more training data is available.

Although CMBP was originally motivated from Bayesian probability operation which leads to the natural use of integer polynomial coefficient $w \in \mathbb{Z}$, CMBP can also be viewed as a data-driven model. Hence, the CMBP framework, equation (16), can be extended to *real* coefficient polynomials. The optimization of real-coefficient CMBP is done by first getting an integer solution and then performing hill climbing search as shown in algorithm 1. Note that the choice of hill climbing search here is just an initial simple solution, and alternative

[3]Basic analysis of the trivial search space of MBP with different orders can be found in Appendix C.

optimization approaches such as Nelder-Mead simplex search [25] can also be used here.

Algorithm 1: Hill climbing algorithm for real coefficient CMBP solution

Let $\mathbf{w} = \{w_{000}, \cdots, w_{666}\}$ be an integer coefficient solution set for equation (16);
Let $I \leftarrow \{i | w_i \in \mathbf{w}, w_i \neq 0\}$;
Let $D \leftarrow \{-0.4, -0.2, -0.1, 0.1, 0.2, 0.4\}$;
Let $\mathbf{w}^r \leftarrow \mathbf{w}$, $done \leftarrow$ **false**;
Let T be the maximum number of iterations[4];
while $T > 0$ **and** $done$ **is false do**
  $T \leftarrow T - 1$;
  $done \leftarrow \mathbf{true}$;
  foreach *index* i *in* I **do**
    foreach *step* d *in* D **do**
      Let $\Delta\mathbf{w} = \{\cdots, w_i + d, \cdots\}$;
      Let $\mathbf{w}^r = \mathbf{w}^r + \Delta\mathbf{w}$;
      if $\mathcal{L}(\mathbf{w}^r) > \mathcal{L}(\mathbf{w}^r)$ **then**
        $\mathbf{w}^r \leftarrow \mathbf{w}^r$, $done \leftarrow$ **false**;
      end
    end
  end
end

The above hybrid CMBP framework effectively bridges rule-based approaches and data-driven approaches. The constraints reflect intuitive prior knowledge and can be set manually, while the general Bayesian polynomial representation allows data-driven optimization of model parameters. An additional advantage of the integer-programming based optimization approach is that it is straightforward to find multiple feasible solutions satisfying constraints. It is then possible to perform system combination on multiple solutions of equation (16) with similar performance to obtain a more robust CMBP system. In this paper, *belief score averaging* is investigated as a simple system combination approach.

IV. Experiment

As introduced in Section II-B, the DSTCs have provided the first common testbed in a standard format, along with a suite of evaluation metrics for dialogue state tracking [5]. In this paper, DSTC-1, DSTC-2 and DSTC-3 tasks are used to evaluate the proposed approach. All tasks provide training dialogues with turn-level ASR hypotheses and SLU hypotheses with confidence scores, as well as human-annotated user goal labels for training. DSTC-1 is a bus timetables domain task [5] and DSTC-2 is a restaurant domain task [16]. Both tasks provide sufficient training and development dialogues [5], [16]. While in DSTC-3, a tourist domain task, only 10 in-domain training dialogues are provided. The DSTC-3 task is to adapt the tracker trained on DSTC-2 data to the new domain with the 10 seed dialogues.

[4]In all our experiments, T is set to 1000. In practice, most of our experiments take several or several dozen iterations. In all our experiments, the number of iterations does not reach T.

TABLE I
Summary of Data Corpora of DSTC-1/2/3. (`dstc1trn*` is the Unlabeled Training Data in DSTC-1 While `dstc1trn` is Labeled

Task	Dataset	#Dialogues	Usage
DSTC-1	`dstc1trn`	2344	Training
	`dstc1trn*`	10619	Not used
	`dstc1eval`	2923	Test
DSTC-2	`dstc2trn`	1612	Training
	`dstc2dev`	506	Training
	`dstc2eval`	1117	Test
DSTC-3	`dstc3seed`	10	Not used
	`dstc3eval`	2265	Test

Table I is a summary of the data usage in this paper. In particular, `dstc1trn` stands for the combination of all the labeled training data, i.e. Train1a, Train2, Train3; `dstc1trn*` stands for the combination of all the unlabeled training data, i.e. Train1b, Train1c; `dstc1eval` stands for the combination of all the test data, i.e. Test1-4 [5]. Note that, for DSTC-2, `dstc2trn` and `dstc2dev` were combined, i.e. 2118 dialogues, to increase the amount of training data for CMBP training. Here, a single CMBP is used for all slots. This makes CMBP domain independent. Therefore, to show the generalization ability of CMBP, for DSTC-3 the 10 seed dialogues were not used at all and the CMBP trained on DSTC-2 was directly used in this paper.

In this section, only joint goal tracking is of interest. The DST evaluation criteria are the *joint goal accuracy* and the *L2* [16], [26]. *Accuracy* is defined as the fraction of turns in which the tracker's 1-best joint goal hypothesis is correct, the larger the better. *L2* is the L2 norm between the distribution of all hypotheses output by the tracker and the correct goal distribution (a delta function), the smaller the better. Moreover, schedule 2 and labelling scheme A defined in [18] are used in both tasks. Specifically, schedule 2 only counts the turns where new information about some slots either in a system confirmation action or in the SLU list is observed. Labelling scheme A is that the labelled state is accumulated forwards through the whole dialogue. For example, the goal for slot s is "`None`" until it is informed as $s = v$ by the user, from then on, it is labelled as v until it is again informed otherwise.

Since the features of CMBP are all probability features, the performance of CMBP is strongly correlated to the quality of confidence scores from SLU. It has been shown that the organizer-provided live SLU confidence was not good enough in DSTC-2/3 tasks [10], [27]. 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 implicitly refine it by inputing the ASR hypotheses into the trackers [10], [11], [12], [21], [23], [28]). In accordance to this, except for the results directly taken from other papers (shown in Table VIII and IX), all experiments of DSTC-2/3 tasks in this paper used the output from a refined semantic parser [10], [27], instead of the live SLU provided by the organizer.

A. Investigation on CMBP Configurations

This section describes the experiments comparing different configurations of CMBP. All experiments were performed on the DSTC-2 tasks. As indicated in Section III-B, multiple

TABLE II
PERFORMANCE OF CMBP WITH DIFFERENT CONSTRAINT SETS ON (`dstc2eval`

Constraint set	#Solutions	Acc	L2
{(40) - (51)}	7926	0.756	0.372
{(31),(40) - (51)}	461	0.756	0.370
{(31),(32),(40) - (51)}	132	0.756	0.375

TABLE III
THE NUMBER OF ALIGNED NON-ZERO COEFFICIENTS BETWEEN EACH PAIR OF THE TOP 5 CMBP SOLUTIONS. BOLD NUMBERS ARE THE NUMBERS OF NON-ZERO COEFFICIENTS OF THE TOP 5 CMBP SOLUTIONS

	1	2	3	4	5
1	**12**				
2	9	**14**			
3	8	13	**13**		
4	10	14	13	**16**	
5	8	10	9	11	**11**

TABLE IV
PERFORMANCE OF THE TOP 5 CMBPS ON `dstc2eval`

Performance	N-Best CMBP Solution				
	1	2	3	4	5
Acc	0.756	0.756	0.756	0.756	0.756
L2	0.370	0.375	0.375	0.375	0.371

feasible solutions can be generated using integer programming, and the feasible solution space is controlled by the number of constraints so that it is neither too big nor too small. Table II compares the integer CMBP performance with different constraint sets[5]. For each constraint set, the detailed description can be found in the appendix. The number of feasible solutions is shown in column #Solutions, and the best integer CMBP is obtained by exhaustively checking the overall joint goal accuracy on the training data set. The performance of the best CMBP is then evaluated on (`dstc2eval`, shown in columns Acc and L2.

It can be seen that larger solution space does not necessarily yield significantly better results. By applying more constraints, the feasible CMBP space can be effectively controlled without losing much performance of the best CMBP contained in the space. In the following experiment, the constraint set is fixed to be {(31), (40)-(51)}.

Since multiple CMBP solutions can be generated, it is interesting to investigate how consistent are the non-zero coefficients across the N-best list of solutions. It can be seen from Table III that the non-zero coefficients of 5-best CMBP solutions are consistent in general. It is also interesting to investigate the performance of multiple feasible CMBP solutions. Table IV shows that the top 5 integer coefficient CMBP models have similar performance. This demonstrates the robustness of CMBP and implies that system combination is likely to be safe.

The top-5 solutions in Table IV were obtained by purely optimizing the overall goal accuracy of the training data (`dstc2trn` + `dstc2dev`). They usually have large complexity (i.e. the number of non-zero integer coefficients), for example, the 1-best solution has 12 parameters. As indicated in

[5]The solution set is calculated by SCIP version 3.1.0 with 8 byte precision. Due to the limited numerical precision, the calculated solution may not be exactly the same as the real solution set.

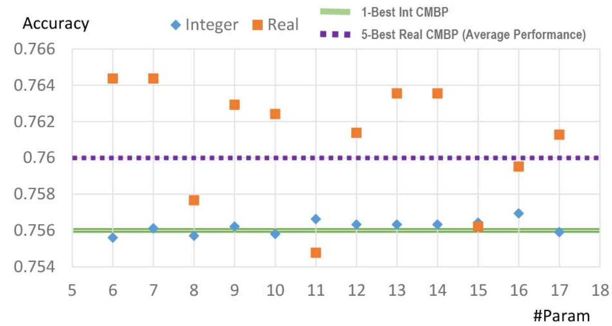

Fig. 2. Joint goal accuracy of real-coefficient CMBPs compared with the integer-coefficient CMBPs with different complexity on `dstc2eval`. The solid line shows the average performance of the top 5 integer-coefficient CMBPs in Table IV. The dotted line shows the average performance of the real-coefficient CMBPs optimized from the top-5 solutions in Table IV.

Section III-B, sparsity penalty can be imposed on CMBP to control complexity and avoid over-fitting. To investigate the effect of sparsity penalty, 12 CMBP models with different number of non-zero integer coefficients were randomly selected (there may be multiple solutions with the same complexity) from the feasible solution space. The results are shown in Fig. 2. Furthermore, the hill climbing algorithm is applied to each of them to generate corresponding optimal real-coefficient CMBPs.

From Fig. 2, the CMBPs with small complexity have similar performance compared with the top 5 integer CMBPs in Table IV. This shows that by enforcing sparsity, the model size can be effectively reduced without hurting the performance. In addition, most of the real-coefficient CMBPs outperform the corresponding integer coefficient CMBPs, demonstrating the importance of extending integer coefficients to real numbers. Another observation is that there is no obvious correlation between the performance of the optimized real-coefficient CMBPs and the corresponding integer-coefficient CMBPs. In practice, the time algorithm 1 takes is positively correlated to the complexity of CMBP. Therefore, to efficiently optimize real-coefficient CMBPs, in the following experiments, for the 1-best CMBP, we only run algorithm 1 on integer-coefficient CMBPs with the smallest number of non-zero coefficients.

It can be seen from Fig. 2 that although the real-coefficient CMBPs have better performance in general, sometimes they do not outperform the corresponding integer-coefficient CMBPs. System combination is then used to make the performance of the real-coefficient CMBPs stable. The combined model, which applies belief score averaging on the CMBP solutions with less than 8 non-zero parameters, achieved an accuracy of 0.762 on (`dstc2eval`. Though not the best possible result, it is competitive and believed to be stable. This system combination setup is also used in the following experiments.

To investigate how much data is needed to reliably train the tracker, the joint goal accuracies of CMBPs trained on different amount of data are compared. It can be seen from Fig. 3 that the performance is generally positively correlated with the amount of data. For the DSTC-2 (in-domain) task, the performance consistently improves with the increase of the training data amount (except for an outlier when the training data is very limited). While for the DSTC-3 (extended domain) task, a

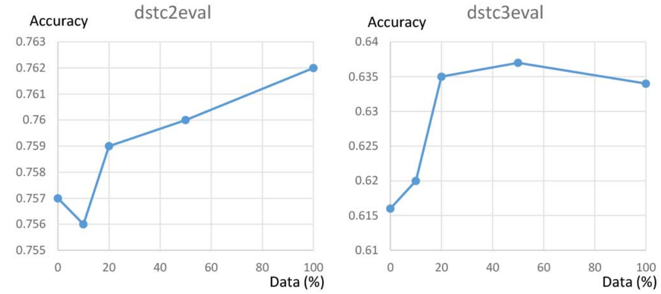

Fig. 3. Performance comparison of CMBPs trained on different amount of data (0%, 10%, 20%, 50%, 100%).

TABLE V
PERFORMANCE COMPARISON BETWEEN HILL CLIMBING SEARCH AND NELDER-MEAD METHOD

Algorithm	dstc2eval		dstc3eval	
	Acc	L2	Acc	L2
Hill Climbing	0.762	0.436	0.634	0.579
Nelder-Mead	0.756	0.372	0.624	0.551

relatively small amount of the DSTC-2 training data (around 20%) is sufficient to yield satisfactory performance and more data may not yield further significant improvements.

Besides hill climbing search, for comparison, Nelder-Mead method was also applied to optimize real-coefficient CMBP. Here, the initial points of Nelder-Mead method were set as randomly selected integer-coefficient CMBP solutions. It can be seen from Table V that hill climbing performs better than Nelder-Mead method.

B. Comparison with Other DST Approaches

The previous section investigates how to get the best integer-coefficient CMBP and the optimized real-coefficient CMBP. In this section, the performance of CMBP is compared to both rule-based and data-driven statistical approaches. As indicated before, CMBP is a hybrid approach, which incorporates data-driven update with the probability features defined in III-A, to make fair comparison, all data-driven statistical models in this section also use similar feature set. Altogether, 1 rule-based trackers and 2 statistical trackers were built for performance comparison:

- **MaxConf** is a *rule-based* model commonly used in spoken dialogue systems which always selects the value with the highest confidence score from the 1st turn to the current turn. It was used as one of the primary baselines in DSTC-1, DSTC-2 and DSTC-3.
- **HWU** is a *rule-based* model based on equation (6), proposed by [14], [15]. It is regarded as a simple, yet competitive baseline of DSTC-2 and DSTC-3.
- **DNN** is a *data-driven statistical model* with probability features similar as CMBP. Since CMBP at turn t uses the belief of the previous turn $b_{t-1}(v)$, to fairly take this into account, the DNN feature set at the tth turn is defined as

$$\bigcup_{i\in\{t-9,\cdots,t\}} \left\{P_i^+(v), P_i^-(v), \tilde{P}_i^+(v), \tilde{P}_i^-(v)\right\} \cup \left\{\hat{P}_t(v)\right\}$$

  where $\hat{P}_t(v)$ is the highest confidence score from the 1st turn to the tth turn. The DNN has 3 hidden layers with 64 nodes per layer.

TABLE VI
PERFORMANCE COMPARISON BETWEEN CMBPS, RULE-BASED AND STATISTICAL APPROACHES WITH A SIMILAR FEATURE SET ON (dstc2eval) AND (dstc3eval)

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
Data-driven	DNN	0.719	0.469	0.628	0.556
	MaxEnt	0.710	0.431	0.607	0.563
CMBP	Int	0.756	0.370	0.623	0.552
	Real	0.764	0.428	0.632	0.591
	Sys. Comb.	0.762	0.436	0.634	0.579

TABLE VII
PERFORMANCE COMPARISON BETWEEN CMBP AND BEST TRACKERS OF DSTC-1 ON dstc1eval. BASELINE* IS THE BEST RESULTS FROM THE 2 ORGANIZER-PROVIDED BASELINE IN DSTC-1. PBM REFERS TO THE PARTITION-BASED MODEL DESCRIBED IN [29]

System	Approach	Rank	Acc	L2
Baseline*	Rule	9	0.231	1.087
Lee et al. [8]	MaxEnt+PBM	1	0.438	0.922
Kim et al. [19]	HIS-POMDP	2	0.345	1.039
Wang et al. [14]	Rule	3	0.341	1.066
Henderson et al. [20]	DNN	4	0.339	1.072
CMBP	Sys. Comb.	1.5	0.402	0.969

- **MaxEnt** is another *data-driven statistical model* using Maximum Entropy with the same input feature as DNN.

The 1-best integer-coefficient and the optimized real-coefficient CMBPs as well as the CMBP system combination are compared with the above DST approaches and the results are shown in Table VI.

It can be observed that, with a similar feature set, CMBPs, especial real-coefficient CMBP, can outperform both rule-based and data-driven statistical approaches in terms of joint goal accuracy. Statistical significance tests were performed assuming a binomial distribution for each turn. CMBP with system combination was shown to significantly outperform both rule-based and data-driven statistical approaches at 95% confidence level. For L2, CMBP system combination is competitive to both rule-based and data-driven statistical approaches. As indicated before, for DSTC-3, the CMBP model trained for DSTC-2 was directly used without modification.[6] From Table VI, the CMBP with system combination performance is stably good: slightly worse than the real-coefficient CMBP on dstc2eval, but better on dstc3eval. This demonstrates that the CMBP with system combination has good generalization ability.

C. Comparison with the State-of-the-art DST Trackers

In the DSTCs, the state-of-the-art trackers mostly employed data-driven statistical approaches. Usually, a richer feature set and more complicated model structures than the statistical models in Section IV-B are used. In this section, the proposed CMBP approach is compared to the best submitted trackers in DSTC-1/2/3, regardless of fairness of feature selection and the SLU refinement approach. The real-coefficient CMBP with system combination is compared and the results are shown in Table VII, VIII and IX.

From Table VII, it can be seen that CMBP ranks second to the best tracker in accuracy and L2. It is worth mentioning that the best tracker in DSTC-1 consists of three discriminative models

TABLE VIII
PERFORMANCE COMPARISON BETWEEN CMBP AND BEST TRACKERS OF DSTC-2 ON dstc2eval. BASELINE* IS THE BEST RESULTS FROM THE 4 ORGANIZER-PROVIDED BASELINES IN DSTC-2

System	Approach	Rank	Acc	L2
Baseline*	Rule	5	0.719	0.464
Williams [12]	LambdaMART	1	0.784	0.735
Henderson et al. [11]	RNN	2	0.768	0.346
Sun et al. [10]	DNN	3	0.750	0.416
CMBP	Sys. Comb.	2.5	0.762	0.436

TABLE IX
PERFORMANCE COMPARISON BETWEEN CMBP AND BEST TRACKERS OF DSTC-3 ON dstc3eval. BASELINE* IS THE BEST RESULTS FROM THE 4 ORGANIZER-PROVIDED BASELINES IN DSTC-3

System	Approach	Rank	Acc	L2
Baseline*	Rule	6	0.575	0.691
Henderson et al. [21]	RNN	1	0.646	0.538
Kadlec et al. [23]	Rule	2	0.630	0.627
Sun et al. [28]	Int CMBP	3	0.610	0.556
CMBP	Sys. Comb.	1.5	0.634	0.579

and one generative model [8] which is fairly complex, CMBP achieves competitive performance with much smaller model complexity.

Note that, in DSTC-2, the Williams's system [12] employed batch ASR hypothesis information (i.e. off-line ASR re-decoded results) and cannot be used in the normal on-line mode in practice. Hence, the practically best tracker is Henderson *et al.* [11]. It can be observed from Table VIII, CMBP ranks only second to the best practical tracker in accuracy, although the L2 performance is slightly worse. The difference between CMBP and Henderson's system is not statistically significant. Since accuracy is the most important criterion, considering that CMBP used much simpler feature set and can operate very efficiently, it is still competitive.

It can be seen from Table IX, CMBP trained on DSTC-2 can achieve state-of-the-art performance on DSTC-3 without modification[6]. This demonstrates that CMBP successfully inherits the advantage of good generalization ability of rule-based model. Actually, the third best tracker in DSTC-3 is an integer coefficient CMBP with system combination. Real-coefficient CMBP can obtain further gains and become the most competitive rule-based model so far. It is also worth noting that the second best tracker in DSTC-3 also used rule-based approach whose original formulation [23] is quoted as follows[7]:

$$P(S_{t+1}) = \sum_{S_t} \sum_{O_{t+1}} P(S_{t+1}|S_t, O_{t+1}, a_t) P(S_t) P(O_{t+1}) \tag{17}$$

where $P(S_t)$ is the *goal* belief state at time t which corresponds to $b_t(\mathbf{g})$, $P(O_t)$ is the semantic hypothesis confidence which corresponds to $q(\mathbf{u}_t)$ in this paper. It can be seen that the above equation is similar as equation (9), the MBP rewritten form of Bayesian generative model. Compared with the general CMBP formulation defined by equation (12), the above equation models multiple slots simultaneously and more importantly, the Bayesian polynomial coefficients are dependent on the *system action* a_t via the dialogue model $P(S_t|S_{t-1}, O_t, a_t)$. Specifically, the dialogue model is a piece-wise function [23], which has different values when a_t is *CantHelp*, *Select*, or neither of them. Hence, equation (17) can be regarded as a piece-wise polynomial extension of CMBP, equation (12), but without slot independence assumption. Since the CMBP framework is easy to extend, future work will investigate piece-wise polynomials and richer features.

[6]The parser is refined for DSTC-3 [27]

[7]We made slight modification on the original notation so that the comparison is easy to understand.

V. CONCLUSION

This paper proposes a novel hybrid framework, referred to as *constrained Markov Bayesian polynomial* (CMBP), to bridge the rule-based model and statistical, or data-driven, approaches. It uses a polynomial function of SLU probabilities and previously estimated belief states as the estimator and employs constraints to incorporate prior knowledge. By approximating descriptive constraints using linear constraints, the CMBP training is formulated as a standard problem of optimization with linear constraints. Furthermore, the integer coefficient CMBP is extended to real-coefficient CMBP. With the ability of incorporating prior knowledge and being data-driven, CMBP, as a hybrid model, has the advantages of both rule-based and data-driven approaches. Experiments on three dialog state tracking challenge (DSTC) tasks showed that the proposed approach not only is more stable than many major statistical approaches, but also has competitive performance, outperforming many state-of-the-art trackers. There are two limitations of the current CMBP approaches. First, it employs a non-differentiable criterion for training and hence is not easy to optimize. Second, it still assumes slot independence due to computational cost. The second issue is also applicable to many existing statistical DST approaches as described in Section II-D. Future work will address these two issues to obtain further improvements. Moreover, the hill climbing search used in this paper is just a straightforward and preliminary choice, future work will also investigate better optimization approaches.

APPENDIX
CMBP CONSTRAINTS FORMULATION

In order to be consistent with Section III-B and introduce the constraints clearly, the constraints formulation of order-3 CMBP is the focus in the following content. The constraints formulation of CMBP of other order can be obtained with just slight modifications of the constraints formulation of order-3 CMBP. As definition (8), the coefficients of CMBP of order 3 is denoted by w_{ijk}:

$$\mathcal{P}(x_1, x_2, x_3, x_4, x_5, x_6) = \sum_{0 \le i \le j \le k \le 6} w_{ijk} x_i x_j x_k \tag{18}$$

where $x_0 = 1$, and $w \in \mathbb{Z}$.

A. Constraints Formulation

The probabilistic constraints, intuition constraints, and regularization constrains investigated in this paper are described below respectively.

Probabilistic constraints:

$$0 \le b_t(v) \le 1 \tag{19}$$
$$0 \le b_t^r \le 1 \tag{20}$$
$$b_t^r = 1 - \sum_{v'} b_t(v') \tag{21}$$

Intuition constraints:

- If neither positive nor negative information is collected, the belief should not change.

$$P_{t+1}^+(v) = 0 \wedge P_{t+1}^-(v) = 0 \wedge \tilde{P}_{t+1}^+(v) = 0 \wedge \tilde{P}_{t+1}^-(v) = 0 \Rightarrow b_{t+1}(v) = b_t(v) \tag{22}$$

where here "∧" and "⇒" are used to denote logical conjunction and material implication respectively.

- If both ASR and SLU is perfectly correct, that is, 1 is assigned to all correct values and 0 to all incorrect values, then the model should always give the correct result. Considering the special case that there is only one value which is not "None", the following 3 constraints can be obtained.

$$P_t^+(v) = 1 \Rightarrow b_t(v) \ge 0.5 \tag{23}$$
$$P_t^-(v) = 1 \Rightarrow b_t(v) \le 0.5 \tag{24}$$
$$\tilde{P}_t^+(v) = 1 \Rightarrow b_t(v) \le 0.5 \tag{25}$$

- The belief should be unchanged or positively correlated with the positive scores from SLU.

$$\frac{\partial \mathcal{P}(P_{t+1}^+(v), P_{t+1}^-(v), \tilde{P}_{t+1}^+(v), \tilde{P}_{t+1}^-(v), b_t^r, b_t(v))}{\partial P_{t+1}^+(v)} \ge 0 \tag{26}$$

- The belief should be unchanged or negatively correlated with the negative scores from SLU.

$$\frac{\partial \mathcal{P}(P_{t+1}^+(v), P_{t+1}^-(v), \tilde{P}_{t+1}^+(v), \tilde{P}_{t+1}^-(v), b_t^r, b_t(v))}{\partial P_{t+1}^-(v)} \le 0 \tag{27}$$

- The belief should be unchanged or negatively correlated with the sum of the positive scores of the other values.

$$\frac{\partial \mathcal{P}(P_{t+1}^+(v), P_{t+1}^-(v), \tilde{P}_{t+1}^+(v), \tilde{P}_{t+1}^-(v), b_t^r, b_t(v))}{\partial \tilde{P}_{t+1}^+(v)} \le 0 \tag{28}$$

- The belief should be unchanged or positively correlated with the sum of the negative scores of the other values.

$$\frac{\partial \mathcal{P}(P_{t+1}^+(v), P_{t+1}^-(v), \tilde{P}_{t+1}^+(v), \tilde{P}_{t+1}^-(v), b_t^r, b_t(v))}{\partial \tilde{P}_{t+1}^-(v)} \ge 0 \tag{29}$$

- The belief of the current turn should be unchanged or positively correlated with the belief of the previous turn.

$$\frac{\partial \mathcal{P}(P_{t+1}^+(v), P_{t+1}^-(v), \tilde{P}_{t+1}^+(v), \tilde{P}_{t+1}^-(v), b_t^r, b_t(v))}{\partial b_t(v)} \ge 0 \tag{30}$$

Regularization constraints:

- The coefficients of $\mathcal{P}(\cdot)$ is limited to be in $[-1, 1]$. This constraint comes from the observation that all coefficients of rule-based model (6) are in $[-1, 1]$.

$$-1 \le w_{ijk} \le 1 \tag{31}$$

- The sum of the coefficients of $\mathcal{P}(\cdot)$ is limited to be 0. This constraint comes from the observation that the sum of the coefficients of rule-based model (6) is 0.

$$\sum_{0 \le i \le j \le k \le 6} w_{ijk} = 0 \tag{32}$$

B. Constraints Approximation

To simplify the presentation, the set consisting of all possible input vectors $((P_t^+(v), P_t^-(v), \tilde{P}_t^+(v), \tilde{P}_t^-(v), b_t^r, b_t(v)))$ is denoted by $\boldsymbol{\chi}$. By definition, the following relations and (19), (20), (21) are true:

$$0 \le P_t^+(v) \le 1 \tag{33}$$
$$0 \le P_t^-(v) \le 1 \tag{34}$$
$$0 \le \tilde{P}_t^+(v) \le 1 \tag{35}$$
$$0 \le \tilde{P}_t^-(v) \le 1 \tag{36}$$
$$0 \le P_t^+(v) + \tilde{P}_t^+(v) \le 1 \tag{37}$$
$$0 \le P_t^-(v) + \tilde{P}_t^-(v) \le 1 \tag{38}$$

Therefore,

$$\boldsymbol{\chi} = \{(x_1, x_2, x_3, x_4, x_5, x_6) | 0 \le x_1 \le 1 \wedge 0 \le x_2 \le 1 \wedge 0 \le x_3 \le 1 \wedge 0 \le x_4 \le 1 \wedge x_1 + x_3 \le 1 \wedge x_2 + x_4 \le 1 \wedge 0 \le x_5 \le 1 \wedge 0 \le x_6 \le 1 \wedge x_5 + x_6 \le 1\} \tag{39}$$

The conversion from the exact constraints to the relaxed linear constraints is discussed in detail as below. For approximation purpose, two quantized interval of $[0, 1]$, T_5 and T_{10}, need to be defined first:

$$T_5 = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$$
$$T_{10} = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$$

A number of theorems are then proved for the constraints approximation.

Theorem 1: If a rule satisfies constraints (19), (20), (21), then the rule satisfies the following sets of linear constraints:

$$\{0 \le \mathcal{P}(\boldsymbol{a}) \le 1 | \boldsymbol{a} \in \boldsymbol{\chi}, a_i \in T_5, i = 1, \cdots, 6\} \tag{40}$$
$$\{0 \le \mathcal{P}(\boldsymbol{a}) + \mathcal{P}(\boldsymbol{b}) \le 1 | \boldsymbol{a}, \boldsymbol{b} \in \boldsymbol{\chi}, a_1 + a_3 = b_1 + b_3, a_2 + a_4 = b_2 + b_4, a_1 \le b_3, b_1 \le a_3, a_2 \le b_4, b_2 \le a_4, a_5 = b_5, a_5 + a_6 + b_6 = 1, a_i, b_i \in T_5, i = 1, \cdots, 6\} \tag{41}$$

Proof: The set of linear constraints (40) can be obtained by constraint (19). By combining constraint (20) and (21), it can be proved that $0 \le \sum_{v'} b_t(v') \le 1$. Thus the set of linear constraints (41) can be obtained by considering the special case that there are at least 2 values which are not "None". ■

Theorem 2: A rule satisfies constraint (22) if and only if

$$w_{000} = w_{005} = w_{055} = w_{056} = w_{066} = w_{555} = w_{556} = w_{566} = w_{666} = 0 \tag{42}$$

and

$$w_{006} = 1 \tag{43}$$

Proof: Suppose constraints (42) and (43) hold. Under the condition $P^+_{t+1}(v) = P^-_{t+1}(v) = \tilde{P}^+_{t+1}(v) = \tilde{P}^-_{t+1}(v) = 0$, then for all v, $(P^+_{t+1}(v) = P^-_{t+1}(v) = 0)$ by the definition of $P^+_{t+1}(v), P^-_{t+1}(v), \tilde{P}^+_{t+1}(v), \tilde{P}^-_{t+1}(v)$ and constraints (33), (34). Thus by definition (12) and equation (18)

$$\begin{aligned} b_{t+1}(v) &= w_{000} + w_{005}b^r_t + w_{055}(b^r_t)^2 + w_{056}b^r_t b_t(v) \\ &\quad + w_{066}(b_t(v))^2 + w_{555}(b^r_t)^3 + w_{556}(b^r_t)^2 b_t(v) \\ &\quad + w_{566}b^r_t(b_t(v))^2 + w_{666}(b_t(v))^3 + w_{006}b_t(v) \\ &= b_t(v) \end{aligned}$$

Therefore, constraint (22) holds. Reversely suppose constraint (22) holds, it is easy to check that under the condition that $P^+_{t+1}(v) = 0 \wedge P^-_{t+1}(v) = 0 \wedge \tilde{P}^+_{t+1}(v) = 0 \wedge \tilde{P}^-_{t+1}(v) = 0$, if at least one of constraint (42) or (43) does not hold, the equality "$b_{t+1}(v) \equiv b_t(v)$" does not hold. ■

Theorem 3: If a rule satisfies constraints (23), (24), (25), then the rule satisfies the following set of linear constraints:

$$\{\mathcal{P}(1,0,0,0,a_5,0) \geq 0.5 | a_5 \in T_{10}\} \tag{44}$$

$$\{\mathcal{P}(0,1,0,0,a_5,a_6) \leq 0.5 | a_5, a_6 \in T_{10}, a_5 + a_6 = 1\} \tag{45}$$

$$\{\mathcal{P}(0,0,1,0,a_5,a_6) \leq 0.5 | a_5, a_6 \in T_{10}, a_5 + a_6 = 1\} \tag{46}$$

Proof: The set of linear constraints (44) can be obtained by simply combining constraint (23) and definition (12). The derivations for the sets of linear constraints (45) and (46) are similar. ■

Theorem 4: If a rule satisfies constraints (26), (27), (28), (29), (30), then the rule satisfies the following sets of linear constraints:

$$\left\{ \mathcal{P}(\mathbf{a}) \geq \mathcal{P}(\mathbf{b}) \,\middle|\, \begin{matrix} \mathbf{a}, \mathbf{b} \in \boldsymbol{\chi}, a_i \in T_5, i = 1, \cdots, 6 \\ a_1 = b_1 + 0.1,\ a_i = b_i\ \forall i \neq 1 \end{matrix} \right\} \tag{47}$$

$$\left\{ \mathcal{P}(\mathbf{a}) \geq \mathcal{P}(\mathbf{b}) \,\middle|\, \begin{matrix} \mathbf{a}, \mathbf{b} \in \boldsymbol{\chi}, a_i \in T_5, i = 1, \cdots, 6 \\ a_2 = b_2 + 0.1,\ a_i = b_i\ \forall i \neq 2 \end{matrix} \right\} \tag{48}$$

$$\left\{ \mathcal{P}(\mathbf{a}) \geq \mathcal{P}(\mathbf{b}) \,\middle|\, \begin{matrix} \mathbf{a}, \mathbf{b} \in \boldsymbol{\chi}, a_i \in T_5, i = 1, \cdots, 6 \\ a_3 = b_3 + 0.1,\ a_i = b_i\ \forall i \neq 3 \end{matrix} \right\} \tag{49}$$

$$\left\{ \mathcal{P}(\mathbf{a}) \geq \mathcal{P}(\mathbf{b}) \,\middle|\, \begin{matrix} \mathbf{a}, \mathbf{b} \in \boldsymbol{\chi}, a_i \in T_5,\ i = 1, \cdots, 6 \\ a_4 = b_4 + 0.1,\ a_i = b_i\ \forall i \neq 4 \end{matrix} \right\} \tag{50}$$

$$\left\{ \mathcal{P}(\mathbf{a}) \geq \mathcal{P}(\mathbf{b}) \,\middle|\, \begin{matrix} \mathbf{a}, \mathbf{b} \in \boldsymbol{\chi}, a_i \in T_5, i = 1, \cdots, 6 \\ a_6 = b_6 + 0.1,\ a_i = b_i\ \forall i \neq 6 \end{matrix} \right\} \tag{51}$$

Proof: The rule satisfies the set of linear constraints (47) is because constraint (26) indicates $\mathcal{P}(x_1, x_2, x_3, x_4, x_5, x_6)$ is monotonically increasing with respect to x_1. The derivations for the other sets of linear constraints are similar. ■

By theorem 1, 2, 3 and 4, it can be seen that the linear constraints (40)–(51) relax the constraints (19)–(21), (22)–(30).

C. Estimation of Trivial Search Space

Theorem 5: The trivial search space of integer-coefficient CMBP of order k with constraint (31) only is

$$3^{(k+1)(k+2)(k+3)(k+4)(k+5)(k+6)/720} \tag{52}$$

TABLE X
THE TRIVIAL SEARCH SPACE OF INTEGER-COEFFICIENT CMBP OF ORDER 2, 3, 4

Order	2	3	4
Search Space	3^{28}	3^{84}	3^{210}

Proof: The number of different monomials in CMBP of order k is the number of multisubset of size k from the set $\{P^+_{t+1}(v), P^-_{t+1}(v), \tilde{P}^+_{t+1}(v), \tilde{P}^-_{t+1}(v), b^r_t, b_t(v), 1\}$ which can be calculated by

$$\begin{pmatrix} 7 + k - 1 \\ k \end{pmatrix} = \frac{(k+1)(k+2)(k+3)(k+4)(k+5)(k+6)}{720} \tag{53}$$

Because constraint (31) limits the coefficients of each monomial to be in $[-1, 1]$, the coefficient of each monomial in the integer-coefficient CMBP can only be in

$$\{-1, 0, 1\} \tag{54}$$

Formula (52) can be obtained by simply combining (53) and (54). ■

The trivial search space of integer-coefficient CMBP of order 2, 3, 4 calculated by theorem 5 is shown in Table X.

ACKNOWLEDGMENT

Thanks to Tobias Achterberg for making SCIP [24] available for research purposes.

REFERENCES

[1] J. D. Williams and S. Young, "Partially observable Markov decision processes for spoken dialog systems," *Comput. Speech Lang.*, vol. 21, no. 2, pp. 393–422, 2007.

[2] B. Thomson and S. Young, "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems," *Comput. Speech Lang.*, vol. 24, no. 4, pp. 562–588, 2010.

[3] M. Gašić and S. Young, "Effective handling of dialogue state in the hidden information state POMDP-based dialogue manager," *ACM Trans. Speech Lang. Process. (TSLP)*, vol. 7, no. 3, p. 4, 2011.

[4] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The hidden information state model: A practical framework for POMDP-based spoken dialogue management," *Comput. Speech Lang.*, vol. 24, no. 2, pp. 150–174, 2010.

[5] J. Williams, A. Raux, D. Ramachandran, and A. Black, "The dialog state tracking challenge," in *Proc. SIGDIAL Conf.*, Aug. 2013, pp. 404–413, Assoc. Comput. Linguist..

[6] J. D. Williams, "Challenges and opportunities for state tracking in statistical spoken dialog systems: Results from two public deployments," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 8, pp. 959–970, Dec. 2012.

[7] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, and J. Williams *et al.*, "Recent advances in deep learning for speech research at Microsoft," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 8604–8608.

[8] S. Lee and M. Eskenazi, "Recipe for building robust spoken dialog state trackers: Dialog state tracking challenge system description," in *Proc. SIGDIAL Conf.*, Metz, France, Aug. 2013, pp. 414–422, Assoc. Comput. Linguist..

[9] S. Lee, "Structured discriminative model for dialog state tracking," in *Proc. SIGDIAL Conf.*, Metz, France, Aug. 2013, pp. 442–451, Assoc. Comput. Linguist..

[10] K. Sun, L. Chen, S. Zhu, and K. Yu, "The SJTU system for dialog state tracking challenge 2," in *Proc. 15th Annu. Meeting Special Interest Group Discourse Dialogue (SIGDIAL)*, Philadelphia, PA, USA, Jun. 2014, pp. 318–326, Assoc. Comput. Linguist..

[11] M. Henderson, B. Thomson, and S. Young, "Word-based dialog state tracking with recurrent neural networks," in *Proc. 15th Annu. Meeting Special Interest Group Discourse Dialogue (SIGDIAL)*, Philadelphia, PA, USA, Jun. 2014, pp. 292–299, Assoc. Comput. Linguist..

[12] J. D. Williams, "Web-style ranking and SLU combination for dialog state tracking," in *Proc. 15th Annu. Meeting Special Interest Group Discourse Dialogue (SIGDIAL)*, Philadelphia, PA, USA, Jun. 2014, pp. 282–291, Assoc. Comput. Linguist..

[13] L. Zilka, D. Marek, M. Korvas, and F. Jurcicek, "Comparison of Bayesian discriminative and generative models for dialogue state tracking," in *Proc. SIGDIAL Conf.*, Metz, France, Aug. 2013, pp. 452–456, Assoc. Comput. Linguist..

[14] Z. Wang and O. Lemon, "A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information," in *Proc. SIGDIAL Conf.*, Metz, France, Aug. 2013, pp. 423–432, Assoc. Comput. Linguist..

[15] Z. Wang, "HWU baseline belief tracker for dstc 2 & 3," Tech. Rep., Oct. 2013.

[16] M. Henderson, B. Thomson, and J. D. Williams, "The second dialog state tracking challenge," in *Proc. 15th Annu. Meeting Special Interest Group Discourse Dialogue (SIGDIAL)*, Philadelphia, PA, USA, Jun. 2014, pp. 263–272, Assoc. Comput. Linguist..

[17] B. Thomson, *Statistical Methods for Spoken Dialogue Management*. New York, NY, USA: Springer Science & Business Media, 2013.

[18] M. Henderson, B. Thomson, and J. Williams, "Dialog state tracking challenge 2 & 3," 2013. [Online]. Available: http://camdial.org/~mh521/dstc/downloads/handbook.pdf

[19] D. Kim, J. Choi, K. Kim, J. Lee, and J. Sohn, "Engineering statistical dialog state trackers: A case study on DSTC," in *Proc. SIGDIAL Conf.*, Metz, France, Aug. 2013.

[20] M. Henderson, B. Thomson, and S. Young, "Deep neural network approach for the dialog state tracking challenge," in *Proc. SIGDIAL Conf.*, Metz, France, Aug. 2013, pp. 467–471, Assoc. Comput. Linguist..

[21] M. Henderson, B. Thomson, and S. Young, "Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2014, pp. 360–365.

[22] S. Kim and R. Banchs, "Sequential labeling for tracking dynamic dialog states," in *Proc. SIGDIAL Conf.*, Jun. 2014.

[23] R. Kadlec, M. Vodoln, J. Libovick, J. Macek, and J. Kleindienst, "Knowledge-based dialog state tracking," in *Proc. IEEE Spoken Language Technology Workshop*, South Lake Tahoe, CA, USA, Dec. 2014, pp. 348–353.

[24] T. Achterberg, "SCIP: Solving constraint integer programs," *Math. Program. Comput.*, vol. 1, no. 1, pp. 1–41, 2009.

[25] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965.

[26] M. Henderson, B. Thomson, and J. D. Williams, "The third dialog state tracking challenge," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2014, pp. 324–329.

[27] S. Zhu, L. Chen, K. Sun, D. Zheng, and K. Yu, "Semantic parser enhancement for dialogue domain extension with little data," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2014, pp. 336–341.

[28] K. Sun, L. Chen, S. Zhu, and K. Yu, "A generalized rule based tracker for dialogue state tracking," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2014, pp. 330–335.

[29] J. D. Williams, "Incremental partition recombination for efficient tracking of multiple dialog states," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2010, pp. 5382–5385.

Kai Yu (M'06–SM'11) received the B.Eng. degree in automation and the M.Sc. degree from Tsinghua University, China in 1999 and 2002, respectively. He then joined the Machine Intelligence Lab at the Engineering Department at Cambridge University, U.K., where he obtained the Ph.D. degree in 2006. He is currently a Research Professor at Shanghai Jiao Tong University, China. His main research interests lie in the area of speech-based human–machine interaction including speech recognition, synthesis, language understanding and dialogue management. He was selected into the 1000 Overseas Talent Plan (Young Talent) by Chinese government and the Excellent Young Scientists Project of NSFC China. He is a member of the Technical Committee of the Speech, Language, Music and Auditory Perception Branch of the Acoustic Society of China.

Kai Sun received his B.S. in computer science from Shanghai Jiao Tong University in 2015. Then he started his Ph.D. in the Department of Computer Science at Cornell University. His current research interests include speech and language processing, machine learning, and AI in games.

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

Su Zhu received a B.Sc. in computer science from Xi'an Jiao Tong University, in 2013. He is currently a Ph.D. candidate student in computer science at Shanghai Jiao Tong University in China. His main research interests focus on natural language processing, machine learning and deep neural networks, creating solutions for large scale problems in spoken language understanding.