The SJTU System for Dialog State Tracking Challenge 2

Kai Sun, Lu Chen, Su Zhu and Kai Yu
Department of Computer Science and Engineering, Shanghai Jiao Tong University
Shanghai, China
{accreator, chenlusz, paul2204, kai.yu}@sjtu.edu.cn

Abstract

Dialog state tracking challenge provides a common testbed for state tracking algorithms. This paper describes the SJTU system submitted to the second Dialogue State Tracking Challenge in detail. In the system, a statistical semantic parser is used to generate refined semantic hypotheses. A large number of features are then derived based on the semantic hypotheses and the dialogue log information. The final tracker is a combination of a rule-based model, a maximum entropy and a deep neural network model. The SJTU system significantly outperformed all the baselines and showed competitive performance in DSTC 2.

1 Introduction

Dialog state tracking is important because spoken dialog systems rely on it to choose proper actions as spoken dialog systems interact with users. However, due to automatic speech recognition (ASR) and spoken language understanding (SLU) errors, it is not easy for the dialog manager to maintain the true state of the dialog. In recent years, much research has been devoted to dialog state tracking. Many approaches have been applied to dialog state tracking, from rule-based to statistical models, from generative models to discriminative models (Wang and Lemon, 2013; Zilka et al., 2013; Henderson et al., 2013; Lee and Eskenazi, 2013). Recently, shared research tasks like the first Dialog State Tracking Challenge (DSTC 1) (Williams et al., 2013) have provided a common testbed and evaluation suite for dialog state tracking (Henderson et al., 2013).

Compared with DSTC 1 which is in the bus timetables domain, DSTC 2 introduces more complicated and dynamic dialog states, which may change through the dialog, in a new domain, i.e. restaurants domain (Henderson et al., 2014). For each turn, a tracker is supposed to output a set of distributions for each of the three components of the dialog state: *goals*, *method*, and *requested slots*. At a given turn, the goals consists of the user's true required value having been revealed for each slot in the dialog up until that turn; the method is the way the user is trying to interact with the system which may be *by name*, *by constraints*, *by alternatives* or *finished*; and the requested slots consist of the slots which have been requested by the user and not replied by the system. For evaluation in DSTC 2, 1-best quality measured by **accuracy**, probability calibration measured by **L2**, and discrimination measured by **ROC** are selected as featured metrics. Further details can be found in the DSTC 2 handbook (Henderson et al., 2013).

Previous research has demonstrated the effectiveness of rule-based (Zilka et al., 2013), maximum entropy (MaxEnt) (Lee and Eskenazi, 2013) and deep neural network (DNN) (Henderson et al., 2013) models separately. Motivated by this, the SJTU system employs a combination of a rule-based model, a MaxEnt and a DNN model. The three models were first trained (if necessary) on the training set and tested for each of the three components of the dialog state, i.e goals, method, and requested slots on the development set. Then, models with the best performance for each of the three components were selected to form a combined model. Finally, the combined model was retrained using both training set and development set. Additionally, as the live SLU was found not good enough with some information lost compared with the live ASR, a new semantic parser was implemented which took the live ASR as input and the SJTU system used the result from the new semantic parser instead of the live SLU.

The remainder of the paper is organized as follows. Section 2 describes the design of the new

Proceedings of the SIGDIAL 2014 Conference, pages 318–326,
Philadelphia, U.S.A., 18-20 June 2014. ©2014 Association for Computational Linguistics

semantic parser. Section 3 presents the rule-based model. Section 4 describes the statistical models including the maximum entropy model and the deep neural network model. Section 5 shows and discusses the performance of the SJTU system. Finally, section 6 concludes the paper.

2 Semantic Parser

It was found that the live SLU provided by the organisers has poor quality. Hence, a new statistical semantic parser is trained to parse the live ASR hypotheses.

2.1 Semantic Tuple Classifier

The semantics of an utterance is represented in functor form called *dialogue act* consisting of a dialogue act type and a list of slot-value pairs, for example:

request(name,food=chinese)

where "request" is the dialogue *act type*,"name" is a *slot* requested and "food=chinese" is a *slot-value pair* which provides some information to the system. In DSTC 2, there are many different dialogue act types (e.g. "request", "inform", "deny", etc) and different slot-value pairs (e.g. "food=chinese", "pricerange=cheap", "area=center", etc), which are all called *semantic items*.

A semantic tuple (e.g. act type, type-slot pair, slot-value pair) classifier (STC) approach developed by Mairesse et al. (2009) is used in the SJTU system. It requires a set of SVMs to be trained on n-gram features from a given utterance: a multiclass SVM is used to predict the dialogue act type, and a binary SVM is used to predict the existence of each slot-value pair. Henderson et al. (2012) improved this method with converting the SVM outputs to probabilities, and approximating the probability of a dialogue-act d of type d-type$_j$ with a set of slot-value pairs S by:

$$P(d|u) = P(\text{d-type}_j|u) \prod_{sv \in S} P(sv|u) \prod_{sv \notin S} (1 - P(sv|u)) \quad (1)$$

where u denotes an utterance and sv runs over all possible slot-value pairs.

2.2 Dialogue Context Features

In addition to the n-gram feature used in the original STC parser, the dialogue context can be exploited to constrain the semantic parser. In DSTC 2, the dialogue context available contains the history information of user's ASR hypotheses, the system act and the other output of system (e.g. whether there is a barge-in from the user or not, the turn-index) and so on. In the SJTU system, the context features from the last system act (indicators for all act types and slot-value pairs on whether they appear), an indicator for "barge-in" and the reciprocal of turn-index are combined with the original n-gram feature to be the final feature vector.

2.3 Generating Confidence Scores

For testing and predicting the dialogue act, the semantic parser is applied to each of the top N ASR hypotheses h_i, and the set of results D_i with m_i distinct dialogue act hypotheses would be merged in following way:

$$P(d|o) = \sum_{i=1}^{N} \begin{cases} p(h_i|o)p(d|h_i) & \text{if } d \in D_i \\ 0 & \text{otherwise} \end{cases}$$

where o is the acoustic observation, d runs over each different dialogue act in $D_i, i = 1, ..., N$, $p(h_i|o)$ denotes the ASR posterior probability of the i-th hypothesis, $p(d|h_i)$ denotes the semantic posterior probability given the i-th ASR hypothesis as in equation (1). Finally, a normalization should be done to guarantee the sum of $P(d|o)$ to be one.

2.4 Implementation

The STCs-based semantic parser is implemented with linear kernel SVMs trained using the LibSVM package (Chang and Lin, 2011). The SVM misclassification cost parameters are optimised individually for each SVM classifier by performing cross-validations on the training data.

3 Rule-based Model

In this section, the rule-based model which is slightly different from the focus tracker (Henderson et al., 2013) and HWU tracker (Wang, 2013) is described. The idea of the rule-based model is to maintain beliefs based on basic probability operations and a few heuristic rules that can be observed on the training set. In the following the rule-based model for joint goals, method and requested slots are described in detail.

3.1 Joint Goals

For slot s, the i-th turn and value v, $p^+_{s,i,v}$ ($p^-_{s,i,v}$) is used to denote the sum of all the confidence scores assigned by the SLU to the user informing or affirming (denying or negating) the value of slot s is v. The belief of "the value of slot s being v in the i-th turn" denoted by $b_{s,i,v}$ is defined as follows.

- If $v \neq$ "$None$",

$$\begin{aligned} b_{s,i,v} &= (b_{s,i-1,v} + p^+_{s,i,v}(1 - b_{s,i-1,v})) \\ &\quad (1 - p^-_{s,i,v} - \sum_{v' \neq v} p^+_{s,i,v'}) \end{aligned}$$

- Otherwise,

$$b_{s,i,v} = 1 - \sum_{v' \neq \text{“}None\text{”}} b_{s,i,v'}$$

In particular, when $i = 0$, $b_{s,0,v} = 1$ if $v =$ "$None$", otherwise 0. The motivation here comes from HWU tracker (Wang, 2013) that only $p^+_{s,\cdot,v}$ positively contributes to the belief of slot s being v, and both $p^+_{s,\cdot,v'}$ ($v' \neq v$) and $p^-_{s,\cdot,v}$ contribute to the belief negatively.

3.2 Method

For the i-th turn, $p_{i,m}$ is used to denote the sum of all the confidence scores assigned by the SLU to method m. Then the belief of "the method being m in the i-th turn" denoted by $b_{i,m}$ is defined as follows.

- If $m \neq$ "$none$",

$$b_{i,m} = b_{i-1,m}(1 - \sum_{m' \neq \text{“}none\text{”}} p_{i,m'}) + p_{i,m}$$

- Otherwise,

$$b_{i,m} = 1 - \sum_{m' \neq \text{“}none\text{”}} b_{i,m'}$$

In particular, $b_{0,m} = 0$ when $i = 0$ and $m \neq$ "$none$". An explanation of the above formula is given by Zilka et al. (2013). The idea is also adopted by the focus tracker (Henderson et al., 2013).

3.3 Requested Slots

For the i-th turn and slot r, $p_{i,r}$ is used to denote the sum of all the confidence scores assigned by the SLU to r is one of the requested slots. Then the belief of "r being one of the requested slots in the i-th turn" denoted by $b_{i,r}$ is defined as follows.

- If $i = 1$, or system has at least one of the following actions: "canthelp", "offer", "reqmore", "confirm-domain", "expl-conf", "bye", "request",

$$b_{i,r} = p_{i,r}$$

- Otherwise,

$$b_{i,r} = b_{i-1,r}(1 - p_{i,r}) + p_{i,r}$$

This rule is a combination of the idea of HWU tracker (Wang, 2013) and an observation from the labelled data that once system has some certain actions, the statistics of requested slots from the past turn should be reset.

4 Statistical Model

In this section, two statistical models, one is the MaxEnt model, the other is the DNN model, are described.

4.1 Features

The performance of statistical models is highly dependent on the feature functions.

Joint Goals

For slot s, the i-th turn and value v, the feature functions designed for joint goals are listed below.

- $f_1 \triangleq inform(s, i, v)$ = the sum of all the scores assigned by the SLU to the user informing the value of slot s is v.
- $f_2 \triangleq affirm(s, i, v)$ = the sum of all the scores assigned by the SLU to the user affirming the value of slot s is v.
- $f_3 \triangleq pos(s, i, v) = inform(s, i, v) + affirm(s, i, v)$.
- $f_4 \triangleq deny(s, i, v)$ = the sum of all the scores assigned by the SLU to the user denying the value of slot s is v.
- $f_5 \triangleq negate(s, i, v)$ = the sum of all the scores assigned by the SLU to the user negating the value of slot s is v.

- $f_6 \triangleq neg(s,i,v) = deny(s,i,v) + negate(s,i,v)$.

- $f_7 \triangleq acc(s,i,v) = pos(s,i,v) - neg(s,i,v)$.

- $f_8 \triangleq rule(s,i,v)$ = the confidence score given by the rule-based model.

- $f_9 \triangleq rank\_inform(s,i,v)$ = the sum of all the reciprocal rank of the scores assigned by the SLU to the user informing the value of slot s is v, or 0 if informing v cannot be found in the SLU n-best list.

- $f_{10} \triangleq rank\_affirm(s,i,v)$ = the sum of all the reciprocal rank of the scores assigned by the SLU to the user affirming the value of slot s is v, or 0 if affirming v cannot be found in the SLU n-best list.

- $f_{11} \triangleq rank^{+}(s,i,v) = rank\_inform(s,i,v) + rank\_affirm(s,i,v)$.

- $f_{12} \triangleq rank\_deny(s,i,v)$ = the sum of all the reciprocal rank of the scores assigned by the SLU to the user denying the value of slot s is v, or 0 if denying v cannot be found in the SLU n-best list.

- $f_{13} \triangleq rank\_negate(s,i,v)$ = the sum of all the reciprocal rank of the scores assigned by the SLU to the user negating the value of slot s is v, or 0 if negating v cannot be found in the SLU n-best list.

- $f_{14} \triangleq rank^{-}(s,i,v) = rank\_deny(s,i,v) + rank\_negate(s,i,v)$.

- $f_{15} \triangleq rank(s,i,v) = rank^{+}(s,i,v) - rank^{-}(s,i,v)$.

- $f_{16} \triangleq max(s,i,v)$ = the largest score given by SLU to the user informing, affirming, denying, or negating the value of slot s is v from the 1-st turn.

- $f_{17} \triangleq rest(s,i,v)$ = 1 if $v =$ "$None$", otherwise 0.

- $f_{18} \triangleq \overline{pos}(s,i,v) = \frac{\sum_{k=1\leq i} pos(s,k,v)}{i}$, which is the arithmetic mean of $pos(s,\cdot,v)$ from the 1-st turn to the i-th turn. Similarly, $f_{19} \triangleq \overline{neg}(s,i,v)$, $f_{20} \triangleq \overline{rank^{+}}(s,i,v)$ and $f_{21} \triangleq \overline{rank^{-}}(s,i,v)$ are defined.

- $\boldsymbol{f}_{22} \triangleq (f_{22,1}, f_{22,2}, \cdots, f_{22,10})$, where $f_{22,j} \triangleq bin\_pos(s,i,v,j) = \frac{tot_{pos}(s,i,v,j)}{Z}$, where $tot_{pos}(s,i,v,j)$ = the total number of slot-value pairs from the 1-st turn to the i-th turn with slot s and value v which will fall in the j-th bin if the range of confidence scores is divided into 10 bins, and $Z = \sum_{k\leq i, 1\leq j'\leq 10, v'} tot_{pos}(s,k,v',j')$, which is the normalization factor. Similarly, $\boldsymbol{f}_{23} \triangleq (f_{23,1}, f_{23,2}, \cdots, f_{23,10})$ where $f_{23,j} \triangleq bin\_neg(s,i,v,j)$ is defined.

- $\boldsymbol{f}_{24} \triangleq (f_{24,1}, f_{24,2}, \cdots, f_{24,10})$. Where $f_{24,j} \triangleq bin\_rule(s,i,v,j) = \frac{tot_{rule}(s,i,v,j)}{Z}$, where $tot_{rule}(s,i,v,j)$ = the total number of $rule(s,\cdot,v)$ from the 1-st turn to the i-th turn which will fall in the j-th bin if the range of $rule(\cdot,\cdot,\cdot)$ is divided into 10 bins, and $Z = \sum_{k\leq i, 1\leq j'\leq 10, v'} tot_{rule}(s,k,v',j')$, which is the normalization factor. Similarly, $\boldsymbol{f}_{25} \triangleq (f_{25,1}, f_{25,2}, \cdots, f_{25,10})$ where $f_{25,j} \triangleq bin\_rank(s,i,v,j)$, and $\boldsymbol{f}_{26} \triangleq (f_{26,1}, f_{26,2}, \cdots, f_{26,10})$ where $f_{26,j} \triangleq bin\_acc(s,i,v,j)$ are defined.

- $\boldsymbol{f}_{27} \triangleq (f_{27,1}, f_{27,2}, \cdots, f_{27,10})$. Where $f_{27,j} \triangleq bin\_max(s,i,v,j) = 1$ if $max(s,i,v)$ will fall in the j-th bin if the range of confidence scores is divided into 10 bins, otherwise 0.

- $\boldsymbol{f}_{28} \triangleq (f_{28,1}, f_{28,2}, \cdots, f_{28,17})$. Where $f_{28,j} \triangleq user\_acttype(s,i,v,u_j)$ = the sum of all the scores assigned by the SLU to the user act type being $u_j (1 \leq j \leq 17)$. There are a total of 17 different user act types described in the handbook of DSTC 2 (Henderson et al., 2013).

- $\boldsymbol{f}_{29} \triangleq (f_{29,1}, f_{29,2}, \cdots, f_{29,17})$. Where $f_{29,j} \triangleq machine\_acttype(s,i,v,m_j)$ = the number of occurrences of act type $m_j (1 \leq j \leq 17)$ in machine act. There are a total of 17 different machine act types described in the handbook of DSTC 2 (Henderson et al., 2013).

- $f_{30} \triangleq canthelp(s,i,v)$ = 1 if the system cannot offer a venue with the constrain $s = v$, otherwise 0.

- $f_{31} \triangleq slot\_confirmed(s,i,v)$ = 1 if the system has confirmed $s = v$, otherwise 0.

- $f_{32} \triangleq slot\_requested(s, i, v)$ = 1 if the system has requested the slot s, otherwise 0.
- $f_{33} \triangleq slot\_informed(s, i, v)$ = 1 if the system has informed $s = v$, otherwise 0.
- $f_{34} \triangleq bias(s, i, v)$ = 1.

In particular, all above feature function are 0 when $i \leq 0$.

Method

For the i-th turn and method m, the feature functions designed for method are listed below.

- $f_1 \triangleq slu(i, m)$ = the sum of all the scores assigned by the SLU to the method being m.
- $f_2 \triangleq rank(i, m)$ = the sum of all the reciprocal rank of the scores assigned by the SLU to the method being m.
- $f_3 \triangleq rule(i, m)$ = the confidence score given by the rule-based model.
- $f_4 \triangleq \overline{slu}(i, m)$ = $\frac{\sum_{k=1}^{i} slu(k,m)}{i}$, which is the arithmetic mean of $slu(\cdot, m)$ from the 1-st turn to the i-th turn. Similarly, $f_5 \triangleq \overline{rank}(i, m)$ and $f_6 \triangleq \overline{rule}(i, m)$ are defined.
- $f_7 \triangleq score\_name(i)$ = the sum of all the scores assigned by the SLU to the user informing the value of slot $name$ is some value.
- $f_8 \triangleq venue\_offered(i)$ = 1 if at least one venue has been offered to the user by the system from the 1-st turn to the i-th turn, otherwise 0.
- $\boldsymbol{f}_9 \triangleq (f_{9,1}, f_{9,2}, \cdots, f_{9,17})$. Where $f_{9,j} \triangleq user\_acttype(i, u_j)$ = the sum of all the scores assigned by the SLU to the user act type being $u_j (1 \leq j \leq 17)$.
- $f_{10} \triangleq bias(i)$ = 1.

In particular, all above feature function are 0 when $i \leq 0$.

Requested Slots

For the i-th turn and slot r, the feature functions designed for requested slots are listed below.

- $f_1 \triangleq slu(i, r)$ = the sum of all the scores assigned by the SLU to r being one of the requested slots.
- $f_2 \triangleq rank(i, r)$ = the sum of all the reciprocal rank of the scores assigned by the SLU to r being one of the requested slots.
- $f_3 \triangleq rule(i, r)$ = the confidence score given by the rule-based model.
- $f_4 \triangleq bias(i, r)$ = 1

In particular, all above feature function are 0 when $i \leq 0$.

4.2 Maximum Entropy Model

Total 6 MaxEnt models (Bohus and Rudnicky, 2006) are employed, four models for the joint goals, one for the method and one for the requested slots. The Maximum Entropy model is an efficient means that models the posterior of class y given the observations $\boldsymbol{x}$:

$$P(y|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\boldsymbol{\lambda}^T \boldsymbol{f}(y, \boldsymbol{x})\right)$$

Where $Z(\boldsymbol{x})$ is the normalization constant. $\boldsymbol{\lambda}$ is the parameter vector and $\boldsymbol{f}(y, \boldsymbol{x})$ is the feature vector.

The models for the joint goals are implemented for four informable slots (i.e. area, food, name and pricerange) separately. In the k-th turn, for every informable slot s and its value v, i.e. slot-value pair in SLU, the MaxEnt model for the corresponding slot is used to determine whether the value v for the slot s in the user goals is right or not. The input consists of 160 features [1] which are selected from the feature functions described in section 4.1 Joint Goals:

$$\{f_{34}\}_{i=k} \cup \bigcup_{k-2 \leq i \leq k} \{f_1, \cdots, f_{15}, \boldsymbol{f}_{28}, \cdots, f_{33}\}$$

Where i is the turn index . The output of the model is the confidence score that the value v for the slot s is right.

In the k-th turn, the model for the method is used to determine which way the user is trying to interact with the system. The input consists of 97 features which are selected from the feature func-

[1] For the feature function whose range is not 1 dimension, the number of features defined by the feature function is counted as the number of dimensions rather than 1. For example, the number of features defined by $\boldsymbol{f}_{28}$ is 17.

tions described in section 4.1 Method:

$$\{f_{10}\}_{i=k} \cup \bigcup_{k-3\leq i\leq k} \{f_7, f_8, \boldsymbol{f}_9\}$$
$$\cup \bigcup_{\substack{m \\ k-3 \leq i \leq k}} \{f_3\}$$

and the output consists of five confidence scores that the method belongs to every one of the five ways (i.e. *by name*, *by constraints*, *by alternatives*, *finished* and *none*).

The model for the requested slots is used to determine whether the requestable slot r in the SLU "request(slot)" is truly requested by the user or not in the k-th turn. The input consists of 10 features which are selected from the feature functions described in section 4.1 Requested Slots:

$$\{f_4\}_{i=k} \cup \bigcup_{k-2\leq i\leq k} \{f_1, f_2, f_3\}$$

and the output is the confidence score that r is truly requested by the user in this turn.

The parameters of the 6 MaxEnt models are optimised separately through maximizing the likelihood of the training data. The training process is stopped when the likelihood change is less than 10^{-4}.

4.3 Deep Neural Network Model

4 DNNs for joint goals (one for each slot), 1 for method and 1 for requested slots are employed. All of them have similar structure with Sigmoid for hidden layer activation and Softmax for output layer activation. As shown in figure 1, each DNN has 3 hidden layers and each layer has 64 nodes. DNNs take the feature set (which will be described in detail later) of a certain value of goal, method, or requested slots as the input, then output two values (donated by X and Y), through the hidden layer processing, and finally the confidence of the value can be got by $\frac{e^X}{e^X+e^Y}$.

For slot s, the k-th turn and value v, the feature set of goal consisting of 108 features is defined as:

$$\bigcup_{k-5\leq i\leq k} \{f_3, f_6, f_7, f_8, f_{11}, f_{14}, f_{15}\}$$
$$\cup \{f_{18}, \cdots, f_{21}\}_{i=k-6}$$
$$\cup \{f_{16}, f_{17}, \boldsymbol{f}_{22}, \cdots, \boldsymbol{f}_{27}\}_{i=k}$$

For the k-th turn and method m, the feature set of method consisting of 15 features is defined as:

$$\bigcup_{k-3\leq i\leq k} \{f_1, f_2, f_3\} \cup \{f_4, f_5, f_6\}_{i=k-4}$$

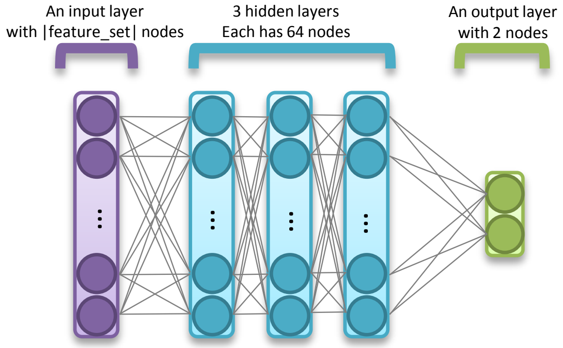

Figure 1: Structure of the DNN Model

For the k-th turn and slot r, the feature set of requested slots consisting of 12 features is defined as:

$$\bigcup_{k-3\leq i\leq k} \{f_1, f_2, f_3\}$$

Bernoulli-Bernoulli RBM was applied to pretrain DNNs and Stochastic Gradient Descent with cross-entropy criterion to fine-tune DNNs. For the fine-tuning process, $3/4$ of the data was used for training and $1/4$ for validation.

5 Experiments

DSTC 2 provides a training dataset of 1612 dialogues (11677 utterances) and a development set of 506 dialogues (3934 utterances). The training data was first used to train the semantic parser and the MaxEnt and the DNN models for internal system development as shown in section 5.1 and 5.2. These systems were tested on the development data. Once the system setup and parameters were determined, the training and development set were combined together to train the final submitted system. The final system was then tested on the final evaluation data as shown in section 5.3.

5.1 Effect of the STC Semantic Parser

In DSTC 2, as the live semantic information was found to be poor, two new semantic parsers were then trained as described in section 2. One used the top ASR hypothesis n-gram features and the other one employed additional system feedback features (the last system act, "barge-in" and turn-index).

Table 1 shows the performance of two new semantic parser in terms of the precision, recall,

System	Precision	Recall	F-score	ICE
baseline	0.6659	0.8827	0.7591	2.1850
1-best	0.7265	0.8894	0.7997	1.4529
+ sys_fb	0.7327	0.8969	0.8065	1.3449

Table 1: Performance of semantic parsers with different features on the development set.

F-score of top dialogue act hypothesis and the Item Cross Entropy (ICE) (Thomson et al., 2008) which measures the overall quality of the confidences distribution of semantic items (the less the better). The baseline is the original live semantic hypotheses, "1-best" (row 3) represents the semantic parser trained on the top ASR hypothesis with n-gram feature, and "sys_fb" (row 4) represents the semantic parser added with the system feedback features. The STC semantic parsers significantly improve the quality of semantic hypotheses compared with baseline in the score of precision, recall, F-score and ICE. And the parser using context features (row 4) scored better than the other one (row 3).

The improved semantic parsers are expected to also yield better performance in dialogue state tracking. Hence, the parsers were used in focus baseline provided by the organiser. As shown in table 2, the new parsers achieved consistent improvement on the accuracy of joint goals, method and requested slots. So the semantic hypotheses of parser using the system feedback features was used for later development.

	Joint Goals	Method	Requested
baseline	0.6121	0.8303	0.8936
1-best	0.6613	0.8764	0.8987
+ sys_fb	0.6765	0.8764	0.9297

Table 2: Results for focus baseline tracker with different parsers

5.2 Internal System Development

Table 3 shows the the results of rule-based model, the MaxEnt model and the DNN model on the development set. From the table we can see that the DNN model has the best performance for joint goals, the MaxEnt model has the best performance for method and the rule-based model has the best performance for requested slots. So the combined model is a combination of those three models, one for one of the three components where it has the best performance, that is, the rule-based model for requested slots, the MaxEnt model for method, and the DNN model for joint goals.

	Joint Goals	Method	Requested
Rule-based	0.6890	0.8955	0.9668
MaxEnt	0.6741	0.9079	0.9665
DNN	0.6906	0.8991	0.9661

Table 3: Performance of three tracking models

5.3 Evaluation Performance

The official results of the challenge are publicly available and the SJTU team is team 7. Entry 0, 1, 2, 3 of team 7 is the combined model, the rule-based model, the DNN model and the MaxEnt model respectively. They all used the new semantic parser based on live ASR hypotheses. Entry 4 of team 7 is also a combined model but it does not use the new semantic parser and takes the live SLU as input.

Table 4 shows the results on the final evaluation test set. As expected, the semantic parser does work, and the combined model has the best performance for joint goals and method, however, that is not true for requested slots. Notice that on the development set, the difference of the accuracy of requested slots among the 3 models is significantly smaller than that of joint goals and method. One reasonable explanation is that one cannot claim that the rule-based model has better performance for requested slots than the MaxEnt model and the DNN model only with an accuracy advantage less than 0.1%.

	Joint Goals	Method	Requested
Baseline	0.6191	0.8788	0.8842
Focus	0.7193	0.8670	0.8786
HWU	0.7108	0.8971	0.8844
HWU+	0.6662	0.8846	0.8830
Rule-based	0.7387	0.9207	0.9701
MaxEnt	0.7252	0.9357	0.9717
DNN	0.7503	0.9287	0.9710
Combined+	0.7503	0.9357	0.9701
Combined-	0.7346	0.9102	0.9458

Table 4: Accuracy of the combined model (Combined+) compared with the rule-based model, the MaxEnt model, the DNN model, the combined model without the new semantic parser (Combined-) and four baselines on the test set. Four baselines are the baseline tracker (Baseline), the focus tracker (Focus), the HWU tracker (HWU) and the HWU tracker with "original" flag set to (HWU+) respectively.

Figure 2 summaries the performance of the approach relative to all 31 entries in the DSTC 2.

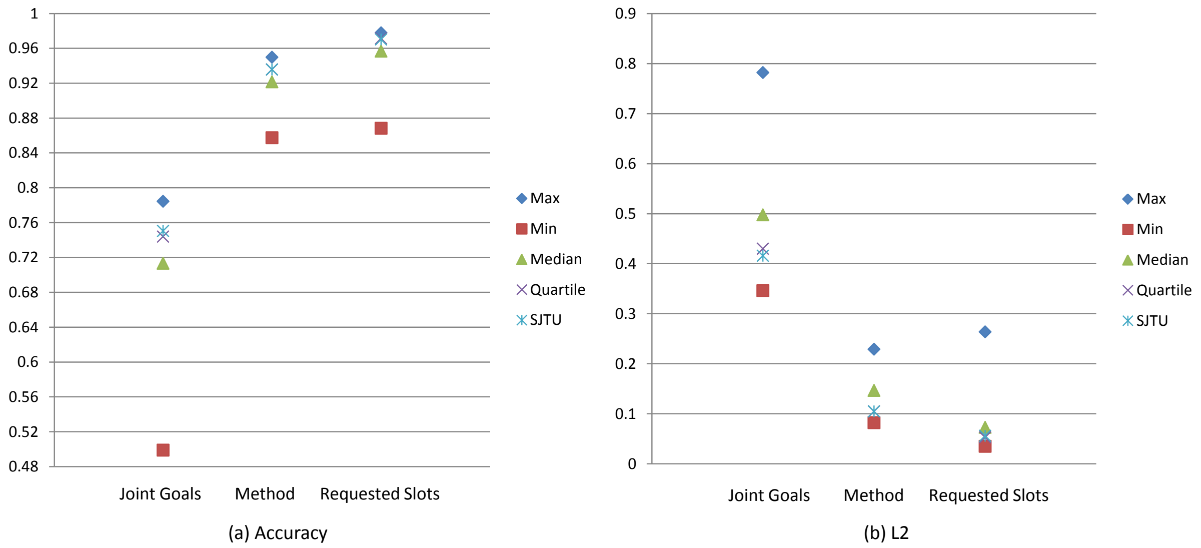

Figure 2: Performance of the combined model among 31 trackers. SJTU is the combined model (entry 0 of team 7).

As ROC metric is only comparable between systems of similar accuracy, only accuracy and L2 are compared. The results of the combined model is competitive for all the three components, especially for joint goals.

5.4 Post Evaluation Analysis

Two strategies and two kinds of features were added to the MaxEnt model for the requested slots after DSTC 2 based on some observations on the training set and development set. The first strategy is that the output for the requested slots of the first turn is set to empty by force. The second strategy is that the output of the confidence is additionally multiplied by $(1-C_f)$, where C_f denotes the confidence given by the MaxEnt model to the method of current turn being *finished*. As for the two kinds of features, one is the slot indicator and the other is the acttype-slot tuple. They are defined as [2]:

- $\boldsymbol{f}_5 \triangleq (f_{5,1}, f_{5,2}, \cdots, f_{5,8})$, where $f_{5,j} \triangleq slot\_indicator(i, r, s_j) = 1$ if the index of the slot r is j, i.e. $s_j = r$, otherwise 0.

- $\boldsymbol{f}_6 \triangleq (f_{6,1}, f_{6,2}, \cdots, f_{6,33})$, where $f_{6,j} \triangleq user\_act\_slot(i, r, t_j)$ = the sum of all the scores assigned by the SLU to the j-th user acttype-slot tuple t_j. The acttype-slot tuple is the combination of dialog act type and possible slot, e.g. $inform\text{-}food$, $confirm\text{-}area$. There are 33 user acttype-slot tuples.

- $\boldsymbol{f}_7 \triangleq (f_{7,1}, f_{7,2}, \cdots, f_{7,46})$, where $f_{7,j} \triangleq sys\_act\_slot(i, r, t_j)$ = the number of occurrences of the j-th machine acttype-slot tuple t_j in the dialog acts. There are 46 machine acttype-slot tuples.

With those strategies and features, the MaxEnt model achieved an accuracy of 0.9769 for the requested slots, which is significantly improved compared with the submitted system.

6 Conclusion

This paper describes the SJTU submission for DSTC 2 in detail. It is a combined system consisting of a rule-based model, a maximum entropy model and a deep neural network model with a STC semantic parser. The results show that the SJTU system is competitive and outperforms most of the other systems in DSTC 2 on test datasets. Post evaluation analysis reveal that there is still room for improvement by refining the features.

Acknowledgments

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.

References

Blaise Thomson, Kai Yu, Milica Gasic, Simon Keizer, Francois Mairesse, Jost Schatzmann and Steve

[2] The feature number is consistent with that in section 4.1.

Young. 2008. Evaluating semantic-level confidence scores with multiple hypotheses. In *INTERSPEECH*, pp. 1153-1156.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), 27.

Dan Bohus and Alex Rudnicky. 2006. A K-hypotheses + Other Belief Updating Model. In *Proc. of AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*.

François Mairesse, Milica Gasic, Filip Jurcícek, Simon Keizer, Blaise Thomson, Kai Yu and Steve Young. 2009. Spoken language understanding from unaligned data using discriminative classification models. *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 4749-4752. IEEE.

Jason Williams, Antoine Raux, Deepak Ramachandran and Alan Black. 2013. The Dialog State Tracking Challenge. In *SIGDIAL*.

Lukas Zilka, David Marek, Matej Korvas and Filip Jurcicek. 2013. Comparison of Bayesian Discriminative and Generative Models for Dialogue State Tracking. In *SIGDIAL*.

Matthew Henderson, Milica Gasic, Blaise Thomson, Pirros Tsiakoulis, Kai Yu and Steve Young. 2012. Discriminative spoken language understanding using word confusion networks. *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pp. 176-181. IEEE.

Matthew Henderson, Blaise Thomson and Steve Young. 2013. Deep Neural Network Approach for the Dialog State Tracking Challenge. In *SIGDIAL*.

Matthew Henderson, Blaise Thomson and Jason Williams. 2013. Dialog State Tracking Challenge 2 & 3. Technical report, University of Cambridge.

Matthew Henderson, Blaise Thomson and Jason Williams. 2014. The Second Dialog State Tracking Challenge. In *SIGDIAL*.

Sungjin Lee and Maxine Eskenazi. 2013. Recipe For Building Robust Spoken Dialog State Trackers: Dialog State Tracking Challenge System Description. In *SIGDIAL*.

Zhuoran Wang and Oliver Lemon. 2013. A Simple and Generic Belief Tracking Mechanism for the Dialog State Tracking Challenge: On the believability of observed information. In *SIGDIAL*.

Zhuoran Wang. 2013. HWU Baseline Belief Tracker for DSTC 2 & 3. Technical report, Heriot-Watt University.