A Simple and Generic Belief Tracking Mechanism for the Dialog State Tracking Challenge: On the believability of observed information

Zhuoran Wang and **Oliver Lemon**
Interaction Lab, MACS, Heriot-Watt University
Edinburgh, EH14 4AS, United Kingdom
{zhuoran.wang; o.lemon}@hw.ac.uk

Abstract

This paper presents a generic dialogue state tracker that maintains beliefs over user goals based on a few simple domain-independent rules, using basic probability operations. The rules apply to observed system actions and partially observable user acts, without using any knowledge obtained from external resources (i.e. without requiring training data). The core insight is to maximise the amount of information directly gainable from an error-prone dialogue itself, so as to better lower-bound one's expectations on the performance of more advanced statistical techniques for the task. The proposed method is evaluated in the Dialog State Tracking Challenge, where it achieves comparable performance in hypothesis accuracy to machine learning based systems. Consequently, with respect to different scenarios for the belief tracking problem, the potential superiority and weakness of machine learning approaches in general are investigated.

1 Introduction

Spoken dialogue system (SDS) can be modelled as a decision process, in which one of the main problems researchers try to overcome is the uncertainty in tracking dialogue states due to error-prone outputs from automatic speech recognition (ASR) and spoken language understanding (SLU) components (Williams, 2012). Recent advances in SDS have demonstrated that maintaining a distribution over a set of possible (hidden) dialogue states and optimising dialogue policies with respect to long term expected rewards can significantly improve the interaction performance (Roy et al., 2000; Williams and Young, 2007a). Such methods are usually developed under a partially observable Markov decision process (POMDP) framework (Young et al., 2010; Thomson and Young, 2010; Williams, 2010), where the distribution over dialogue states is called a 'belief' and is modelled as a posterior updated every turn given an observation. Furthermore, instead of simply taking the most probable (or highest confidence score) hypothesis of the user act as in 'traditional' handcrafted systems, the observation here may consist of an n-best list of the SLU hypotheses (dialogue acts) with (normalised) confidence scores. See (Henderson and Lemon, 2008; Williams and Young, 2007b; Thomson et al., 2010; Young et al., 2013) for more details of POMDP-based SDS.

It is understandable that beliefs more accurately estimating the true dialogue states will ease the tuning of dialogue policies, and hence can result in better overall system performance. The accuracy of belief tracking has been studied in depth by Williams (2012) based on two SDS in public use. Here the effects of several mechanisms are analysed, which can alter the 'most-believed' dialogue state hypothesis (computed using a generative POMDP model) from the one derived directly from an observed top SLU hypothesis. Williams's work comprehensively explores how and why a machine learning approach (more specifically the generative model proposed in (Williams, 2010)) functions in comparison with a naive baseline. However, we target a missing intermediate analysis in this work: how much information one can gain purely from the SLU n-best lists (and the corresponding confidence scores), without any prior knowledge either being externally learned (using data-driven methods) or designed (based on domain-specific strategies), but beyond only considering the top SLU hypotheses. We explain this idea in greater detail as follows.

Firstly, we can view the belief update procedure in previous models as re-constructing the hidden

Proceedings of the SIGDIAL 2013 Conference, pages 423–432,
Metz, France, 22-24 August 2013.

dialogue states (or user goals) based on the previous belief, a current observation (normally an SLU n-best list), and some prior knowledge. The prior knowledge can be observation probabilities given a hidden state, the previous system action and/or dialogue histories (Young et al., 2010; Thomson and Young, 2010; Williams, 2010), or probabilistic domain-specific ontologies (Mehta et al., 2010), where the probabilities can be either trained on a collection of dialogue examples or manually assigned by human experts. In such models, a common strategy is to use the confidence scores in the observed n-best list as immediate information substituted into the model for belief computation, which implies that the performance of such belief tracking methods to a large extent depends on the reliability of the confidence scores. On the other hand, since the confidence scores may reflect the probabilities of the occurrences of corresponding user acts (SLU hypotheses), a belief can also be maintained based on basic probability operations on those events (as introduced in this paper). Such a belief will advance the estimation obtained from top SLU hypotheses only, and can serve as a baseline to justify how much further improvement is actually contributed by the use of prior knowledge. Note that the fundamental method in this paper relies on the assumption that confidence scores carry some useful information, and their informativeness will affect the performance of the proposed method as will be seen in our experiments (Section 5).

Therefore, this paper presents a generic belief tracker that maintains beliefs over user goals only using information directly observable from the dialogue itself, including SLU n-best list confidence scores and user and system behaviours, such as a user not disconfirming an implicit confirmation of the system, or the system explicitly rejecting a query (since no matching item exists), etc. The belief update is based on simple probability operations and a few very general domain-independent rules. The proposed method was evaluated in the Dialog State Tracking Challenge (DSTC) (Williams et al., 2013). A systematic analysis is then conducted to investigate the extent to which machine learning can advance this naive strategy. Moreover, the results show the performance of the proposed method to be comparable to other machine learning based approaches, which, in consideration of the simplicity of its implementation, suggests that another practical use of the proposed method could be as a module in an initial system installation to collect training data for machine learning techniques, in addition to functioning as a baseline for further analysing them.

The remainder of this paper is organised as follows. Section 2 reviews some basic mathematical background, based on which Section 3 introduces the proposed belief tracker. Section 4 briefly describes the DSTC task. The evaluation results and detailed analysis are illustrated in Section 5. Finally, we further discuss in Section 6 and conclude in Section 7.

2 Basic Mathematics

We first review some basic mathematics, which provide the fundamental principles for our belief tracker. Let $P(X)$ denote the probability of the occurrence of an event X, then the probability of X not occurring is simply $P(\neg X) = 1 - P(X)$. Accordingly, if X occurs at a time with probability $P_1(X)$, and at a second time, it occurs with probability $P_2(X)$ independently of the first time, then the overall probability of its occurrence is $P(X) = 1 - P_1(\neg X)P_2(\neg X) = 1 - (1 - P_1(X))(1 - P_2(X))$. To generalise, we can say that in a sequence of k independent events, if the probability of X occurring at the ith time is $P_i(X)$, the overall probability of X having occurred at least once among the k chances is $P(X) = 1 - \prod_{i=1}^{k} P_i(\neg X) = 1 - \prod_{i=1}^{k}(1 - P_i(X))$. This quantity can also be computed recursively as:

$$P^t(X) = 1 - (1 - P^{t-1}(X))(1 - P_t(X)) \quad (1)$$

where $P^t(X)$ denotes the value of $P(X)$ after t event occurring chances, and we let $P^0(X) = 0$.

Now we consider another situation. Let A be a binary random variable. Suppose that we know the prior probability of A being true is $Pr(A)$. If there is a chance where with probability $P(B)$ we will observe an event B independent of A, and we assume that if B happens, we must set A to false, then after this, the probability of A still being true will become $P(A = \text{true}) = Pr(A) * P(\neg B) = Pr(A)(1 - P(B))$.

3 A Generic Belief Tracker

In this section, we will take the semantics defined in the bus information systems of DSTC as

examples to explain our belief tracker. Without losing generality, the principle applies to other domains and/or semantic representations. The SDS we are interested in here is a turn-based slot-filling task. In each turn, the system executes an action and receives an observation. The observation is an SLU n-best list, in which each element could be either a dialogue act without taking any slot-value arguments (e.g. `affirm()` or `negate()`) or an act presenting one or more slot-value pairs (e.g. `deny(route=64a)` or `inform(date.day=today, time.ampm=am)`), and normalised confidence scores are assigned to those dialogue act hypotheses. In addition, we follow a commonly used assumption that the user's goal does not change during a dialogue unless an explicit `restart` action is performed.

3.1 Tracking Marginal Beliefs

Since a confidence score reflects the probability of the corresponding dialogue act occurring in the current turn, we can apply the probability operations described in Section 2 plus some 'common sense' rules to track the marginal probability of a certain goal being stated by the user during a dialogue trajectory, which is then used to construct our beliefs over user goals. Concretely, we start from an initial belief b_0 with zero probabilities for all the slot-value hypotheses and track the beliefs over individual slot-value pairs as follows.

3.1.1 Splitting-Merging Hypotheses

Firstly, in each turn, we split those dialogue acts with more than one slot-value pairs into single slot-value statements and merge those identical statements among the n-best list by summing over their confidence scores, to yield marginal confidence scores for individual slot-value representations. For example, an n-best list observation:

```
inform(date.day=today, time.ampm=am) 0.7
                inform(date.day=today) 0.3
```

after the splitting-merging procedure will become:

```
inform(date.day=today)     1
  inform(time.ampm=am)     0.7
```

3.1.2 Applying Rules

Let $P_t(u, s, v)$ denote the marginal confidence score for a user dialogue act $u(s = v)$ at turn t. Then the belief $b_t(s, v)$ for the slot-value pair (s, v) is updated as:

- **Rule 1:** If $u =$ `inform`, then $b_t(s, v) = 1 - (1 - b_{t-1}(s, v))(1 - P_t(u, s, v))$.
- **Rule 2:** If $u =$ `deny`, then $b_t(s, v) = b_{t-1}(s, v)(1 - P_t(u, s, v))$.

In addition, motivated by some strategies commonly used in rule-based systems (Bohus and Rudnicky, 2005), we consider the effects of certain system actions on the beliefs as well. Let $a(h)$ be one of the system actions performed in turn t, where h stands for a set of n slot-value arguments taken by a, i.e. $h = \{(s_1, v_1), \ldots, (s_n, v_n)\}$. We check:

- **Rule 3:** If a is an implicit or explicit confirmation action (denoted by `impl-conf` and `expl-conf`, respectively) and an `affirm` or `negate` user act u is observed with confidence score $P_t(u)$:
  - **Rule 3.1:** If $u =$ `affirm`, then $b_t(s_i, v_i) = 1 - (1 - b_{t-1}(s_i, v_i))(1 - P_t(u))$, $\forall (s_i, v_i) \in h$.
  - **Rule 3.2:** If $u =$ `negate`, then $b_t(s_i, v_i) = b_{t-1}(s_i, v_i)(1 - P_t(u))$, $\forall (s_i, v_i) \in h$.
- **Rule 4:** Otherwise, if a is an `impl-conf` action, and there are no `affirm`/`negate` user acts observed, and no information presented in a is re-informed or denied in the current turn, then we take all $(s_i, v_i) \in h$ as being affirmed by the user with probability 1.

However, note that, the marginal probabilities $b(s, v)$ computed using the above rules do not necessarily yield valid beliefs, because sometimes we may have $\sum_v b(s, v) > 1$ for a given slot s. When this occurs, a reasonable solution is to seek a multinomial vector $\bar{b}(s, \cdot)$ that minimises the symmetrised Kullback-Leibler (KL) divergence between $b(s, \cdot)$ and itself. It can be checked that solving such an optimisation problem is actually equivalent to simply normalising $b(s, \cdot)$, for which the proof is omitted here but can be found in Appendix B.

Finally, we consider an extra fact that normally a user will not insist on a goal if he/she has been notified by the system that it is impossible to satisfy. (In the DSTC case, such notifications correspond to those `canthelp.*` system actions.) Therefore, we have:

- **Rule 5:** If the system has explicitly disabled a hypothesis h, we will block the generation of any hypotheses containing h in the belief tracking procedure, until the dialogue finishes.

Note here, if h is a marginal hypothesis, eliminating it from our marginal belief will result in joint hypotheses (see Section 3.2) containing h also being blocked, but if h is a joint representation, we will only block the generation of those joint hypothesis containing h, without affecting any marginal belief.

3.2 Constructing Joint Representations

Beliefs over joint hypotheses can then be constructed by probabilistic disjunctions of those marginal representations. For example, given two marginal hypotheses (s_1, v_1) and (s_2, v_2) ($s_1 \neq s_2$) with beliefs $b(s_1, v_1)$ and $b(s_2, v_2)$ respectively, one can compute the beliefs of their joint representations as:

$$b^{\text{joint}}(s_1 = v_1, s_2 = v_2) = b(s_1, v_1)b(s_2, v_2)$$
$$b^{\text{joint}}(s_1 = v_1, s_2 = \texttt{null}) = b(s_1, v_1)b(s_2, \texttt{null})$$
$$b^{\text{joint}}(s_1 = \texttt{null}, s_2 = v_2) = b(s_1, \texttt{null})b(s_2, v_2)$$

where `null` represents that none of the current hypotheses for the corresponding slot is correct, i.e. $b(s, \texttt{null})$ stands for the belief that the information for slot s has never been presented by the user, and can be computed as $b(s, \texttt{null}) = 1 - \sum_v b(s, v)$.

3.3 Limitations

The insight of the proposed approach is to explore the upper limit of the observability one can expect from an error-prone dialogue itself. Nevertheless, this method has two obvious deficiencies. Firstly, the dialogue acts in an SLU n-best list are assumed to be independent events, hence error correlations cannot be handled in this method (which is also a common drawback of most existing models as discussed by Williams (2012)). Modelling error correlations requires statistics on a certain amount of data, which implies a potential space of improvement left for machine learning techniques. Secondly, the model is designed to be biased on the accuracy of marginal beliefs rather than that of joint beliefs. The beliefs for joint hypotheses in this method can only lower-bound the true probability, as the observable dependencies among some slot-value pairs are eliminated by the splitting-merging and rejoining procedures described above. For example, in the worst case, a multi-slot SLU hypothesis `inform`$(s_1 = v_1, s_2 = v_2)$ with a confidence score $p < 1$ may yield two marginal beliefs $b(s_1, v_1) = p$ and $b(s_2, v_2) = p$,[1] then the re-constructed joint hypothesis will have its belief $b^{\text{joint}}(s_1 = v_1, s_2 = v_2) = p^2$, which is exponentially reduced compared to the originally observed confidence score. However, the priority between the marginal hypotheses and the joint representations to a greater extent depends on the action selection strategy employed by the system.

4 Description of DSTC

DSTC (Williams et al., 2013) is a public evaluation of belief tracking (a.k.a. dialogue state tracking) models based on the data collected from different dialogue systems that provide bus timetables for Pittsburgh, Pennsylvania, USA. The dialogue systems here were fielded by three anonymised groups (denoted as Group A, B, and C).

There are 4 training sets (`train1a`, `train1b`, `train2` and `train3`) and 4 test sets (`test1...4`) provided, where all the data logs are transcribed and labelled, except `train1b` which is transcribed but not labelled (and contains a much larger number of dialogues than others). It is known in advance to participants that `test1` was collected using the same dialogue system from Group A as `train1`$\star$ and `train2`, `test2` was collected using a different version of Group A's dialogue manager but is to a certain extent similar to the previous ones, `train3` and `test3` were collected using the same dialogue system from Group B (but the training set for this scenario is relatively smaller than that for `test1`), and `test4` was collected using Group C's system totally different from any of the training sets.

The evaluation is based on several different metrics[2], but considering the nature of our system, we will mainly focus on the hypothesis accuracy, i.e.

[1] The worst case happens when (s_1, v_1) and (s_2, v_2) are stated for the first time in the dialogue and cannot merge with any other marginal hypotheses in the current turn, as their marginal beliefs will remain p without being either propagated by the belief update rules, or increased by the merging procedure.

[2] Detailed descriptions of these metrics can be found in the DSTC handbook at `http://research.microsoft.com/en-us/events/dstc/`

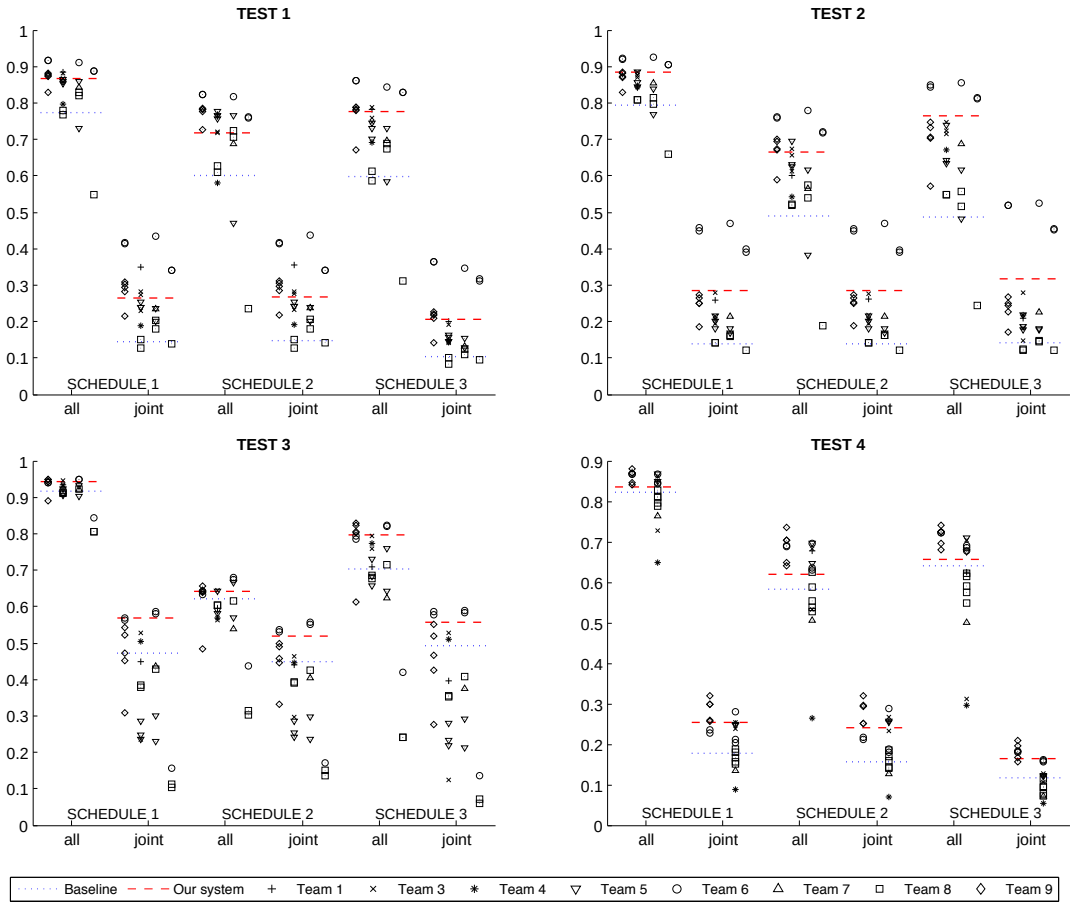

Figure 1: Hypothesis accuracy on the four test sets: the columns in each `schedule`, from left to right, stand for the *ensemble*, *mixed-domain*, *in-domain* and *out-of-domain* system groups, except for `test4` where the last three groups are merged into the right-hand side column.

percentage of turns in which the tracker's 1-best hypothesis is correct, but with the receiver operating characteristic (ROC) performance briefly discussed as well. In addition, there are 3 'schedules' for determining which turns to include when measuring a metric: `schedule 1` – including all turns, `schedule 2` – including a turn for a given concept only if that concept either appears on the SLU n-best list in that turn, or if the system action references that concept in that turn, and `schedule 3` – including only the turn before the `restart` system action (if there is one), and the last turn of the dialogue.

5 Evaluation and Analysis

The method proposed in this paper corresponds to `Team 2`, `Entry 1` in the DSTC submissions. In the following analysis, we will compare it with the 26 machine learning models submitted by the other 8 anonymised participant teams plus a baseline system (`Team 0`, `Entry 1`) that only considers the top SLU result.

Each team can submit up to 5 systems, whilst the systems from a same team may differ from each other in either the statistical model or the training data selection (or both of them). There is a brief description of each system available after the challenge. For the convenience of analysis and illustration, on each test set we categorise these systems into the following groups: *in-domain* – systems trained only using the data sets which are similar (including the 'to-some-extent-similar' ones) to the particular test set, *out-of-domain* – systems trained on the data sets which are totally different from the particular test set, *mixed-domain* – systems trained on a mixture of the *in-domain* and *out-of-domain* data, and *ensemble* – systems combining multiple models to generate their final output. (The *ensemble* systems here are all trained on the *mixed-domain* data.) Note that,

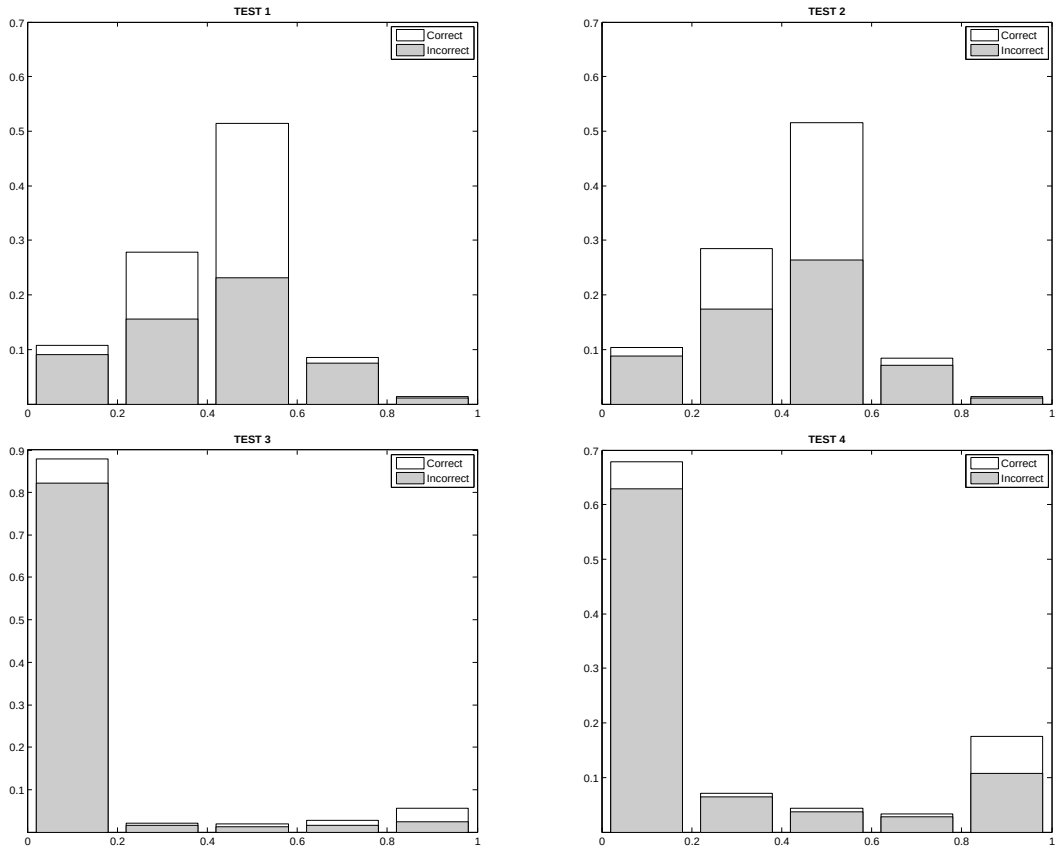

Figure 2: Distributions of SLU confidence scores on the four test sets: The x-axis stands for the confidence score interval, and the y-axis stands for the occurrence rate.

for `test4` there are no *in-domain* data available, so all those *non-ensemble* systems are merged into one group. Detailed system categorisation on each test set can be found in Appendix A.

5.1 Hypothesis Accuracy

We plot the hypothesis accuracy of our method (red dashed line) on the 4 test sets in comparison with the baseline system (blue dotted line) and other systems in Figure 1, where different markers are used to identify the systems from different teams. Here we use the overall accuracy of the marginal hypotheses (`all`) and the accuracy of the joint hypotheses (`joint`) to sketch the general performance of the systems, without looking into the result for each individual slot.

It can be seen that the proposed method produces more accurate marginal and joint hypotheses than the baseline on all the test sets and in all the `schedules`. Moreover, generally speaking, further improvement can be achieved by properly designed machine learning techniques. For example, some systems from `Team 6`, especially their *in-domain* and *ensemble* ones, almost consistently outperform our approach (as well as most of the models from the other teams) in all the above tasks. In addition, the following detailed trends can be found.

Firstly, and surprisingly, our method tends to be more competitive when measured using `schedule 1` and `schedule 3` than using `schedule 2`. As `schedule 2` is supposed to measure system performance on the concepts that are in focus, and to prevent a belief tracker receiving credit for new guesses about those concepts not in focus, the results disagree with our original expectation of the proposed method. A possible explanation here is that some machine learning models tend to give a better belief estimation when a concept is in focus, however their correct top hypotheses might more easily be replaced by other incorrect ones when the focus on the concepts in those correct hypotheses are lost (possibly due to improperly assigned correlations among the concepts). In this sense, our method is more robust, as the beliefs will not change if their corresponding concepts are not in focus.

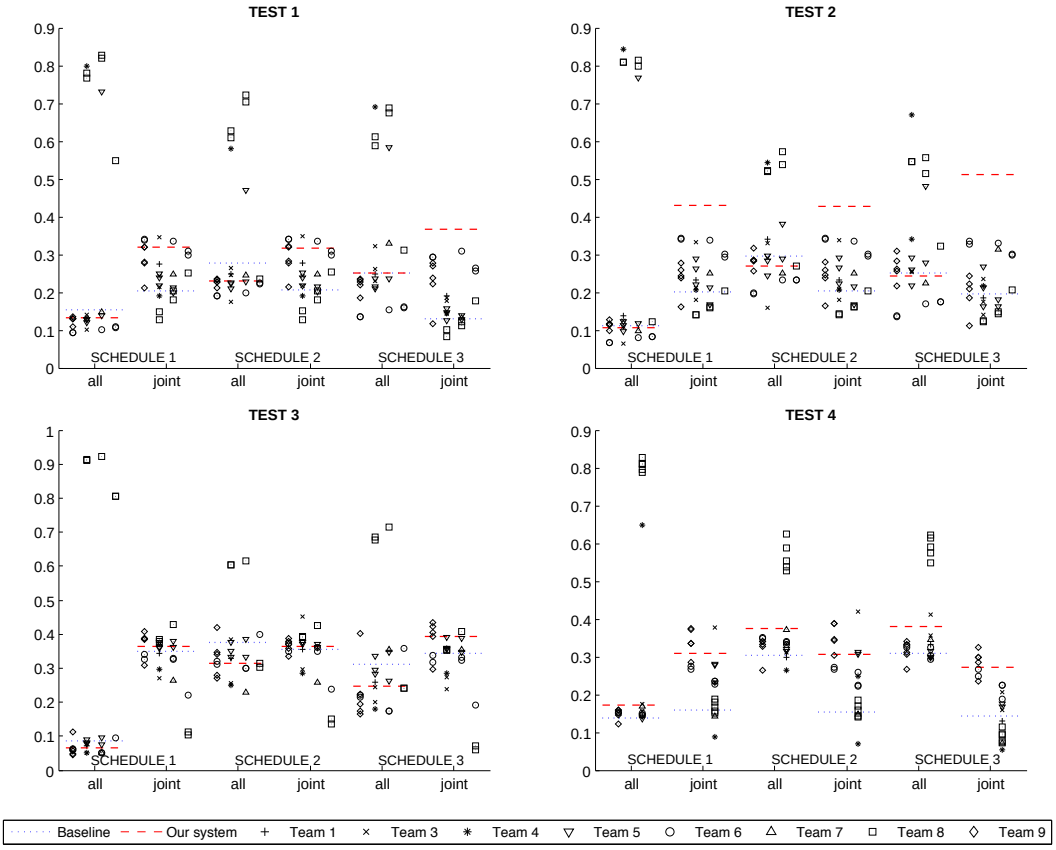

Figure 3: ROC equal error rate on the four test sets: The columns in each `schedule`, from left to right, stand for the *ensemble*, *mixed-domain*, *in-domain* and *out-of-domain* system groups, except for `test4` where the last three groups are merged into the right-hand side column.

Secondly, the proposed method had been supposed to be more preferable when there are no (or not sufficient amount of) *in-domain* training data available for those statistical methods. Initial evidence to support this point of view can be observed from the results on `test1`, `test2` and `test3`. More concretely, when the test data distribution becomes less identical to the training data distribution on `test2`, out system outperforms most of the other systems except those from `Team 6` (and a few others in the `schedule 2/all` task only), compared to its middle-level performance on `test1`. Similarly, on `test3` when the amount of available *in-domain* training data is small, our approach gives more accurate beliefs than most of the others with only a few exceptions in each scenario, even if extra *out-of-domain* data are used to enlarge the training set for many systems. However, the results on `test4` entirely contradicts the previous trend, where a significant number of machine learning techniques perform better than our domain-independent rules without using any *in-domain* training data at all. We analyse such results in detail as follows.

To explain the unexpected outcome on `test4`, our first concern is the influence of Rule 4, which is relatively 'stronger' and more artificial than the other rules. Hence, for the four test sets, we compute the percentage of dialogues where a `impl-conf` system action occurs. The statistics show that the occurrence rates of the implicit confirmation system actions in `test1...4` are 0.01, 0, 0.94 and 0.67, respectively. This means that the two very extreme cases happen in `test3` and `test2` (the situation in `test1` is very similar to `test2`), and the result for `test4` is roughly right in the middle of them, which suggests that Rule 4 will not be the main factor to affect our performance on `test4`. Therefore, we further look into the distributions of the SLU confidence scores across these different test sets. A normalised histogram of the confidence scores for correct and incorrect SLU hypotheses observed in each test set is plotted in Figure 2. Here we only consider

the SLU hypotheses that will actually contribute during our belief tracking processes, i.e. only the `inform`, `deny`, `affirm` and `negate` user dialogue acts. It can be found that the dialogue system used to collect the data in `test4` tends to produce significantly more 'very confident' SLU hypotheses (those with confidence scores greater than 0.8) than the dialogue systems used for collecting the other test sets, where, however, a considerable proportion of its highly confident hypotheses are incorrect. In such a case, our system would be less capable in revising those incorrect hypotheses with high confidence scores than many machine learning techniques, since it to a greater extent relies on the confidence scores to update the beliefs. This finding indicates that statistical approaches will be helpful when observed information is less reliable.

5.2 Discussions on the ROC Performance

Besides the hypothesis accuracy, another important issue will be the ability of the beliefs to discriminate between correct and incorrect hypotheses. Williams (2012) suggests that a metric to measure such performance of a system is the ROC curve. Note that, in the DSTC task, most of the systems from the other teams are based on discriminative models (except two systems, a simple generative model from `Team 3` and a deep neural network method from `Team 1`), which are optimised specifically for discrimination. Unsurprisingly, our approach becomes much less competitive when evaluated based on the ROC curve metrics, as illustrated in Figure 3 using the ROC equal error rate (EER) for the `all` and `joint` scenarios. (ERR stands for the intersection of the ROC curve with the diagonal, i.e. where the false accept rate equals the false reject rate. The smaller the ERR value, the better a system's performance is.) However, our argument on this point is that since an optimised POMDP policy is not a linear classifier but has a manifold decision surface (Cassandra, 1998), the ROC curves may not be able to accurately reflect the influence of beliefs on a system's decision quality, for which further investigations will be needed in our future work.

6 Further Discussions

In this paper, we made the rules for our belief tracker as generic as possible, in order to ensure the generality of the proposed mechanism. However, in practice, it is extendable by using more detailed rules to address additional phenomena if those phenomena are deterministically identifiable in a particular system. For example, when the system confirms a joint hypothesis $(s_1 = v_1, s_2 = v_2)$ and the user negates it and only re-informs one of the two slot-values (e.g. $\texttt{inform}(s_1 = v_1')$), one may consider that it is more reasonable to only degrade the belief on $s_1 = v_1$ instead of reducing the beliefs on both $s_1 = v_1$ and $s_2 = v_2$ synchronously as we currently do in Rule 3.2. However, the applicability of this strategy will depend on whether it is possible to effectively determine such a compact user intention from an observed SLU n-best list without ambiguities.

7 Conclusions

This paper introduces a simple rule-based belief tracker for dialogue systems, which can maintain beliefs over both marginal and joint representations of user goals using only the information observed within the dialogue itself (i.e. without needing training data). Based on its performance in the DSTC task, potential advantages and disadvantages of machine learning techniques are analysed. The analysis here is more focused on general performance of those statistical approaches, where our concerns include the similarity of distributions between the training and test data, the adequacy of available training corpus, as well as the SLU confidence score distributions. Model-specific features for different machine learning systems are not addressed at this stage. Considering its competitiveness and simplicity of implementation, we suggest that the proposed method can serve either as a reasonable baseline for future research on dialogue state tracking problems, or a module in an initial system installation to collect training data for those machine learning techniques.

Acknowledgments

The research leading to these results was supported by the EC FP7 projects JAMES (ref. 270435) and Spacebook (ref. 270019). We thank Jason D. Williams for fruitful comments on an earlier version of this paper. We also acknowledge helpful discussions with Simon Keizer and Heriberto Cuayáhuitl.

References

Dan Bohus and Alexander I. Rudnicky. 2005. Constructing accurate beliefs in spoken dialog systems. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 272–277.

Anthony R. Cassandra. 1998. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. Ph.D. thesis, Brown University, Providence, RI, USA.

James Henderson and Oliver Lemon. 2008. Mixture model POMDPs for efficient handling of uncertainty in dialogue management. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 73–76.

Neville Mehta, Rakesh Gupta, Antoine Raux, Deepak Ramachandran, and Stefan Krawczyk. 2010. Probabilistic ontology trees for belief tracking in dialog systems. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 37–46.

Nicholas Roy, Joelle Pineau, and Sebastian Thrun. 2000. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 93–100.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech and Language*, 24(4):562–588.

Blaise Thomson, Filip Jurčiček, Milica Gašić, Simon Keizer, Francois Mairesse, Kai Yu, and Steve Young. 2010. Parameter learning for POMDP spoken dialogue models. In *Proceedings of IEEE Workshop on Spoken Language Technology*.

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

Jason D. Williams and Steve Young. 2007b. Scaling POMDPs for spoken dialog management. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2116–2129.

Jason D. Williams, Antoine Raux, Deepak Ramachandran, and Alan W. Black. 2013. The Dialog State Tracking Challenge. In *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Jason D. Williams. 2010. Incremental partition recombination for efficient tracking of multiple dialog states. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 5382–5385.

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

Steve Young, Milica Gašić, Simon Keizer, Francois Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The Hidden Information State model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

A System Categorisation

Table 1 shows detailed categorisation of the systems submitted to DSTC, where `TiEj` stands for `Team i, Entry j`.

<table>
<tr><td colspan="4">ensemble</td><td rowspan="12">non-ensemble for <code>test4</code></td></tr>
<tr><td colspan="4"><code>T6E3, T6E4, T9E1, T9E2, T9E3
T9E4, T9E5</code></td></tr>
<tr><td colspan="4">mixed-domain</td></tr>
<tr><td colspan="4"><code>T1E1, T3E1, T3E2, T3E3, T4E1
T5E2, T5E4, T5E5, T8E4, T8E5</code></td></tr>
<tr><td colspan="4">in-domain</td></tr>
<tr><td><code>test1
test2</code></td><td colspan="2"><code>T6E1, T8E1, T8E2</code></td><td><code>T5E1
T5E3</code></td></tr>
<tr><td><code>test3</code></td><td colspan="2"><code>T6E2, T6E5, T8E3</code></td><td><code>T7E1</code></td></tr>
<tr><td colspan="4">out-of-domain</td></tr>
<tr><td><code>test1
test2</code></td><td colspan="3"><code>T6E2, T6E5, T8E3</code></td></tr>
<tr><td><code>test3</code></td><td colspan="3"><code>T6E1, T8E1, T8E2</code></td></tr>
</table>

Table 1: Categorisation of the systems submitted to DSTC.

B Symmetrised KL-divergence Minimisation

We prove the following proposition to support our discussions in the end of Section 3.1.

Proposition 1 *Let $p \in \mathbb{R}^N$ be an arbitrary N-dimensional non-negative vector (i.e. $p \geq 0$). Let $\bar{p} = \frac{p}{\|p\|_1}$, where $\|\cdot\|_1$ stands for the ℓ_1-norm of a vector. Then $\bar{p}$ is the solution of the optimisation problem.* $\min_{q \geq 0, \|q\|_1 = 1} D_{\mathrm{SKL}}(p\|q)$*, where $D_{\mathrm{SKL}}(p\|q)$ denotes the symmetrised KL-divergence between p and q, defined as:*

$$\begin{aligned} D_{\mathrm{SKL}}(p\|q) &= D_{\mathrm{KL}}(p\|q) + D_{\mathrm{KL}}(q\|p) \quad (2) \\ &= \sum_i p_i \log \frac{p_i}{q_i} + \sum_i q_i \log \frac{q_i}{p_i} \end{aligned}$$

and p_i and q_i denote the ith element in p and q respectively.

Proof Let $q^* = \arg\min_{q\geq 0, \|q\|_1=1} D_{\mathrm{SKL}}(p\|q)$. Firstly, using the facts that $\lim_{x\to 0} x\log\frac{x}{y} \to 0$ and $\lim_{x\to 0} y\log\frac{y}{x} \to +\infty, \forall y > 0$, one can easily prove that if $p_i = 0$ then $q_i^* = 0$, and $p_i \neq 0$ then $q_i^* \neq 0$, because otherwise the objective value of Eq. (2) will become unbounded.

Therefore, we only consider the case $p > 0$ and $q > 0$. By substituting $p_i = \bar{p}_i\|p\|_1$ into Eq. (2), we obtain:

$$\begin{aligned}
D_{\mathrm{SKL}}(p\|q) &= \|p\|_1 \sum_i \bar{p}_i \log\frac{\|p\|_1\bar{p}_i}{q_i} \\
&\quad + \sum_i q_i \log\frac{q_i}{\|p\|_1\bar{p}_i} \\
&= \|p\|_1 \left(\sum_i \bar{p}_i \log\frac{\bar{p}_i}{q_i} + \sum_i \bar{p}_i \log\|p\|_1\right) \\
&\quad + \sum_i q_i \log\frac{q_i}{\bar{p}_i} - \sum_i q_i \log\|p\|_1 \\
&= \|p\|_1 \sum_i \bar{p}_i \log\frac{\bar{p}_i}{q_i} + \sum_i q_i \log\frac{q_i}{\bar{p}_i} \\
&\quad + (\|p\|_1 - 1)\log\|p\|_1 \\
&= \|p\|_1 D_{\mathrm{KL}}(\bar{p}\|q) + D_{\mathrm{KL}}(q\|\bar{p}) \\
&\quad + (\|p\|_1 - 1)\log\|p\|_1 \\
&\geq (\|p\|_1 - 1)\log\|p\|_1
\end{aligned}$$

where we use the facts that $\sum_i \bar{p}_i = 1$, $\sum_i q_i = 1$, $D_{\mathrm{KL}}(\bar{p}\|q) \geq 0$ and $D_{\mathrm{KL}}(q\|\bar{p}) \geq 0$, since $\bar{p}$ and q are valid distributions. It can be found that the minimum $(\|p\|_1 - 1)\log\|p\|_1$ is only achievable when $D_{\mathrm{KL}}(\bar{p}\|q) = 0$ and $D_{\mathrm{KL}}(q\|\bar{p}) = 0$, i.e. $q = \bar{p}$, which proves Proposition 1. ∎