THE THIRD DIALOG STATE TRACKING CHALLENGE

Matthew Henderson[1]*, Blaise Thomson*[2] *and Jason D. Williams*[3]

[1]Department of Engineering, University of Cambridge, UK
[2]VocalIQ Ltd., Cambridge, UK
[3]Microsoft Research, Redmond, WA, USA

mh521@eng.cam.ac.uk blaise@vocaliq.com jason.williams@microsoft.com

ABSTRACT

In spoken dialog systems, dialog state tracking refers to the task of correctly inferring the user's goal at a given turn, given all of the dialog history up to that turn. This task is challenging because of speech recognition and language understanding errors, yet good dialog state tracking is crucial to the performance of spoken dialog systems. This paper presents results from the third Dialog State Tracking Challenge, a research community challenge task based on a corpus of annotated logs of human-computer dialogs, with a blind test set evaluation. The main new feature of this challenge is that it studied the ability of trackers to generalize to new entities – i.e. new slots and values not present in the training data. This challenge received 28 entries from 7 research teams. About half the teams substantially exceeded the performance of a competitive rule-based baseline, illustrating not only the merits of statistical methods for dialog state tracking but also the difficulty of the problem.

Index Terms— Dialog state tracking, spoken dialog systems, spoken language understanding.

1. INTRODUCTION

Task-oriented spoken dialog systems interact with users using natural language to help them achieve a goal. As the interaction progresses, the dialog manager maintains a representation of the state of the dialog in a process called *dialog state tracking* (DST). For example, in a tourist information system, the dialog state might indicate the type of business the user is searching for (pub, restaurant, coffee shop), and further constraints such as their desired price range and type of food served. Dialog state tracking is difficult because automatic speech recognition (ASR) and spoken language understanding (SLU) errors are common, and can cause the system to misunderstand the user. At the same time, state tracking is crucial because the system relies on the estimated dialog state to choose actions – for example, which restaurants to suggest.

The dialog state tracking challenge is a series of community challenge tasks that enables studying the state tracking problem by using common corpora of human-computer dialogs and evaluation methods. The first dialog state tracking challenge (DSTC1) used data from a bus timetable domain [1]. The second DSTC (DSTC2) used restaurant information dialogs, and added emphasis on handling user goal changes [2]. Entries to these challenges broke new ground in dialog state tracking, including the use of conditional random fields [3, 4, 5], sophisticated and robust hand-crafted rules [6], neural networks and recurrent neural networks [7, 8], multi-domain learning [9], and web-style ranking [10].

This paper presents results from the third dialog state tracking challenge (DSTC3). Compared to previous DSTCs, the main feature of this challenge is to study the problem of handling of new entity (slot) types and values. For example, the training data for DSTC3 covered only restaurants, but the test data also included pubs and coffee shops. In addition, the test data included slots not in the train data, such as whether a coffee shop had internet, or whether a pub had a TV. This type of generalization is crucial for deploying real-world dialog systems, and has not been studied in a controlled fashion before. Seven teams participated, submitting a total of 28 dialog state trackers. The fully labelled dialog data, tracker output, evaluation scripts and baseline trackers are provided on the DSTC2/3 website[1].

This paper first describes the data and evaluation methods used in this challenge, in sections 2-3. Next, the results from the 7 teams are analyzed in section 4, with a particular emphasis on the problem of handling new slots not in the training data. Section 5 concludes.

2. CHALLENGE OVERVIEW

This challenge was very similar in design to the second dialog state tracking challenge [2]. This section gives a summary of the design, with particular emphasis on the new aspects. Full details are given in [11].

2.1. Challenge design

The data used in the challenge is taken from human-computer dialogs in which people are searching for information about restaurants, pubs, and coffee shops in Cambridge, UK. As in DSTC2, the callers are paid crowd-sourced users with a given

[1]http://camdial.org/~mh521/dstc/

task. Users may specify constraints (such as price range), and may query for information such as a business's address. Constraints and queries are drawn from a common, provided ontology of slots and slot values – see table 1. Thus, in this challenge, the dialog state includes (1) the **goal constraints**, which is the set of constraints desired by the user specified as slot-value pairs, such as `type=pub, pricerange=cheap`; (2) the requested slots, which is a set of zero or more slots the user wishes to hear, such as `address` and `phone`; and (3) the **search method** employed by the user, which is one of – *byconstraints* when the user is searching by constraining slot/value pairs of interest, *byalternatives* as in "What else do you have like that?", *byname* as in "Tell me about Brasserie Gerard", or *finished* if the user is done as in "Thanks, bye." Each turn of each dialog is labelled with these three dialog state components, and the goal of dialog state tracking is to predict the components at each turn, given the ASR, SLU, and system output prior to that turn. This is a challenging problem because the ASR and SLU often contain errors and conflicting information on their N-best lists.

Like the previous challenges, DSTC3 studies the problem of dialog state tracking as a corpus-based task. The challenge task is to re-run dialog state tracking over a test corpus of dialogs. A corpus-based challenge means all trackers are evaluated on the same dialogs, allowing direct comparison between trackers. There is also no need for teams to expend time and money in building an end-to-end system and getting users, meaning a low barrier to entry.

Handling new unseen slots and values is a crucial step toward enabling dialog state tracking to adapt to new domains. To study this – and unlike past DSTCs – the test data includes slots and slot values which are not present in the training data. In particular, whereas the training data included dialogs about only restaurants, the test data included coffee shops and pubs – two new values for the `type` slot. The sets of possible values for slots present in the training set changed in the test set, and several new slots were also introduced: `near` which indicates nearby landmarks such as `Queens College`, and three binary slots: `childrenallowed`, `hastv`, `hasinternet`. Table 1 gives full details.

When a tracker is deployed, it will inevitably alter the performance of the dialog system it is part of, relative to any previously collected dialogs. The inclusion of new slots in the test data ensures that simply fitting the distribution in the train set will not result in good performance.

2.2. Data

A corpus of 2,275 dialogs was collected using paid crowd-sourced workers, as part of a study into the Natural Actor and Belief Critic algorithms for parameter and policy learning in POMDP dialog systems [12]. A set of 11 labelled dialogs were published in advance for debugging, the rest comprising a large test set used for evaluation. The training set consisted of a large quantity of data from DSTC2 in a smaller domain (see table 1).

	Size		
Slot	**Train**	**Test**	**Informable**
type	1*	3	yes
area	5	15	yes
food	91	28	yes
name	113	163	yes
pricerange	3	4	yes
addr	—	—	no
phone	—	—	no
postcode	—	—	no
near	—	52	yes
hastv	—	2	yes
hasinternet	—	2	yes
childrenallowed	—	2	yes

Table 1. Ontology used in DSTC3 for tourist information. Counts do not include the special *Dontcare* value. All slots are requestable, and all slots are present in the test set. (*) For the type slot, 1 value was present at training time (*restaurant*), and 3 values were present at test time (*restaurant, pub, coffee shop*).

Table 2 gives details of the train and test sets, including the Word Error Rate of the top hypothesis from the Automatic Speech Recognition (ASR), and the *F-score* of the top Spoken Language Understanding (SLU) hypothesis, which is calculated as in [13]. One key mis-match is the frequency of goal changes in the data, it being much more common in the training data for the user to change their mind for their constraint on a slot (most often when the system informs their existing constraint cannot be satisfied.)

	# Dialogs	**Goal Changes**	**WER**	**F-score**
Train	3,235	41.1%	28.1%	74.3%
Test	2,275	16.5%	31.5%	78.1%

Table 2. Statistics for the Train and Test sets. *Goal Changes* is the percentage of dialogs in which the user changed their mind for at least one slot. Word Error Rate and F-score are on the top ASR and SLU hypotheses respectively.

3. EVALUATION

A tracker is asked to output a distribution over the three dialog state components – goal constraints, requested slots, and search method – as described in section 2.1. To allow evaluation of the tracker output, the single correct dialog state at each turn is labelled.

Labelling of the dialog state is facilitated by first labelling each user utterance with its semantic representation, in the dialog act format described in [11]. The semantic labelling was achieved by first crowd-sourcing the transcription of the audio to text. Next a semantic decoder was run over the transcriptions, and the authors corrected the decoder's results by hand. Given the sequence of machine actions and user ac-

tions, both represented semantically, the true dialog state is computed deterministically using a simple set of rules.

The components of the dialog state (goal constraint for each slot, the requested slots, and the search method) are each evaluated separately by comparing the tracker output to the correct label. The joint over the goal constraints is evaluated in the same way, where the tracker may either explicitly enumerate and score its joint hypotheses, or let the joint be computed as the product of the distributions over the slots.

A bank of metrics are calculated in the evaluation. The full set of metrics is described in [2], including *Mean reciprocal rank*, *Average probability*, *Log probability* and *Update accuracy*. This section defines the *Accuracy*, *L2* and *ROC V2 CA 05* metrics, which are the *featured metrics* of the evaluation. These metrics were chosen to be featured in DSTC2 and DSTC3 as they each represent one of three groups of mostly uncorrelated metrics as found in DSTC1 [1].

Accuracy is a measure of 1-best quality, and is the fraction of turns where the top hypothesis is correct. **L2** gives a measure of the quality of the tracker scores as probability distributions, and is the square of the l^2 norm between the distribution and the correct label (a delta distribution). The **ROC V2** metrics look at the receiver operating characteristic (ROC) curves, and measure the discrimination in the tracker's output. Correct accepts (CA), false accepts (FA) and false rejects (FR) are calculated as fractions of correctly classified utterances, meaning the values always reach 100% regardless of the accuracy. These metrics measure discrimination independently of the accuracy, and are therefore only comparable between trackers with similar accuracies. Multiple metrics are derived from the ROC statistics, including **ROC V2 CA05**, the correct acceptance rate at a false-acceptance rate 0.05.

Two *schedules* are used to decide which turns to include when computing each metric. **Schedule 1** includes every turn. **Schedule 2** only includes a turn if any SLU hypothesis up to and including the turn contains some information about the component of the dialog state in question, or if the correct label is not *None*. E.g. for a goal constraint, this is whether the slot has appeared with a value in any SLU hypothesis, an affirm/negate act has appeared after a system confirmation of the slot, or the user has in fact informed the slot regardless of the SLU.

The data is labelled using two schemes. The first, **scheme A**, is considered the standard labelling of the dialog state. Under this scheme, each component of the state is defined as the most recently asserted value given by the user. The *None* value is used to indicate that a value is yet to be given.

A second labelling scheme, **scheme B**, is included in the evaluation, where labels are propagated backwards through the dialog. This labelling scheme is designed to assess whether a tracker is able to predict a user's intention before it has been stated. Under scheme B, the label at a current turn for a particular component of the dialog state is considered to be the next value which the user settles on, and is reset in the case of goal constraints if the slot value pair is given in a *canthelp* act by the system (i.e. the system has informed that this constraint is not satisfiable).

The featured metrics (Accuracy, L2 and ROC V2 CA05) are calculated using schedule 2 and labelling scheme A for the joint goal constraints, the search method and the requested slots. This gives 9 numbers altogether. Note that all combinations of schedules, labelling schemes, metrics and state components give a total of 1,265 metrics reported per tracker in the full results, available online.

3.1. Baseline trackers

Four baseline trackers are included in the results, under the ID 'team0'. Source code for all the baseline systems is available on the DSTC website. The first (team0, entry0) follows simple rules commonly used in spoken dialog systems. It gives a single hypothesis for each slot, whose value is the top scoring suggestion so far in the dialog. Note that this tracker does not account well for goal constraint changes; the hypothesised value for a slot will only change if a new value occurs with a higher confidence.

The *focus* baseline (team0, entry1) includes a simple model of changing goal constraints. Beliefs are updated for the goal constraint $s = v$, at turn t, $P\left(s = v\right)$, using the rule:

$$P\left(s = v\right)_t = q_t P\left(s = v\right)_{t-1} + SLU\left(s = v\right)_t$$

where $0 \leq SLU(s = v)_t \leq 1$ is the evidence for $s = v$ given by the SLU in turn t, and $q_t = \sum_{v'} SLU(s = v')_t \leq 1$.

Two further baseline trackers (team0, entry2 and entry3) are included. These are based on the tracker presented in [14], and use a selection of domain independent rules to update the beliefs, similar to the focus baseline.

4. RESULTS

In total, 7 research teams participated, submitting a total of 28 trackers. Appendix A gives the featured metrics for all submitted trackers, and also indicates whether each tracker used the SLU and/or ASR as input. Tracker output and full evaluation reports (as well as scripts to recreate the results) are available on the DSTC website.

The baseline trackers proved strong competition, with only around half of entries beating the top baseline (team0, entry 2) in terms of joint goal accuracy. Figure 1 shows the fraction of turns where each tracker was better or worse than this baseline for joint goal accuracy. Results are shown for the best-performing entry for each team. "Better" means the tracker output the correct user goal where the baseline was incorrect; "worse" means the tracker output the incorrect user goal where the baseline was correct. This shows that even high-performing trackers such as teams 3 and 4 – which *in total* make fewer errors than the baseline – still make some errors that the baselines do not.

Figure 2 shows the same analysis for an "SLU-based oracle tracker", again for the best-performing entry for each team. This tracker considers the items on the SLU N-best list

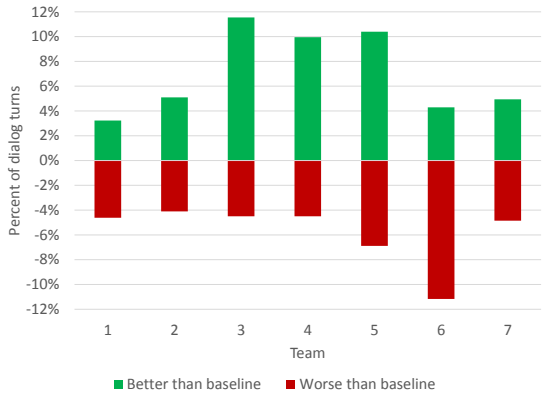


Fig. 1. Fraction of 17,667 dialog turns where the best tracker entry from each team was better or worse than the **best baseline** (team0, entry 2) for schedule 2a joint goal accuracy on the test set.

– it is an "oracle" in the sense that, if a slot/value pair appears that corresponds to the user's goal, it is added to the state with confidence 1.0. In other words, when the user's goal appears somewhere in the SLU N-best list, the oracle always achieves perfect accuracy. The only errors made by the oracle are omissions of slot/value pairs which have not appeared on any SLU N-best list. Figure 2 shows that – for teams 2, 3, 4 and 5 – 3-7% of tracker turns outperformed the oracle. These teams also used ASR features, which suggests they were successfully using ASR results to infer new slot/value pairs. Unsurprisingly, despite these gains no team was able to achieve a net performance gain over the oracle.

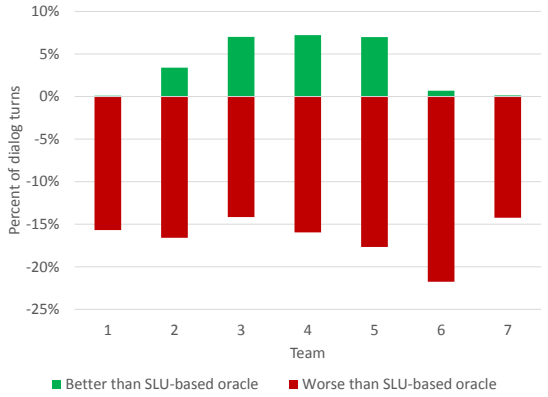


Fig. 2. Fraction of 17,667 dialog turns where the best tracker entry from each team was better or worse than the **oracle tracker** for schedule 2a joint goal accuracy on the test set.

4.1. Tracking unseen slots

Figure 3 shows the performance of each team on tracking slots for which training data was given, and slots unseen in training. Some teams performed worse on the slots for which examples existed (such as teams 1, 2 and 7). This may be evidence of over-tuning in training, if systems attempted to tune to the seen slots, but defaulted to general models for the unseen slots. Generalization to new conditions was found to be a key limitation of some approaches in DSTC1 and DSTC2, where for example trackers often over-estimated their performance relative to a baseline on development sets.

Performance on the individual slots is detailed in appendix A. No tracker was able to beat the top baseline accuracy on the `childrenallowed` slot, however this may be influenced by a small error in labelling found by the authors which affected 14 turns (out of 17,677 total).

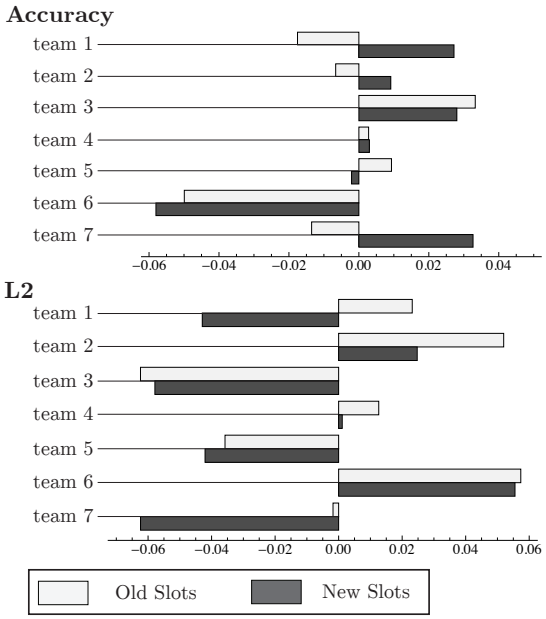


Fig. 3. Performance of each team's best entry under schedule 2a relative to the best baseline (team0, entry2) on *Old slots* and *New slots*, i.e. slots found and not found in the training data respectively. Recall a lower L2 score is better.

4.2. Types of errors

Following [15], for tracking the user's goal three types of slot-level errors can be distinguished:

- *Wrong*: when the user's goal contains a value for a slot, and the tracker outputs an incorrect value for that slot
- *Extra*: when the user's goal does not contain a value for a slot, and the tracker outputs a value for that slot
- *Missing*: when the user's goal contains a value for a slot, and the tracker does not output a value for that slot

Note that a single turn may have multiple slot-level errors. Figure 4 shows the average number of slot-level errors per turn for the best entry from each team, including the best baseline. This figure also shows the average number of correct slots per turn. *Missing* slot errors account for most of the variation in performance, whereas *wrong* slot errors were rather consistent across teams.

5. CONCLUSIONS

The third Dialog State Tracking Challenge built on the tradition of the first two DSTCs in providing an evaluation of the

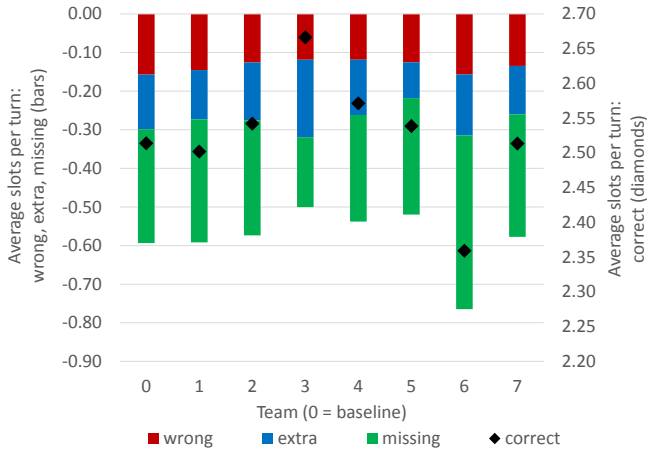


Fig. 4. Average number of slots in error per turn (bar chart, left axis), and average number of correct slots per turn (black diamonds, right axis) for the best tracker from each team, for schedule 2a joint goal accuracy on the test set. See text for explanation of error types.

state of the art in state tracking, with a particular focus on the ability of trackers to generalize to an extended domain.

Results of the blind evaluation show that around half the teams were able to beat the competitive rule-based baseline in terms of joint goal accuracy. Several teams were found to perform better on new parts of the dialog state than they did on parts for which training examples existed. This may be an example of failing to generalize slot-specific models in new conditions, which was an issue found in the first two challenges.

Studying dialog state tracking as an offline corpus task has advantages, and has lead to notable advances in the field, but it is clear that more work should be done to verify improving in these metrics translates to higher quality end-to-end dialog systems.

Acknowledgements

The authors would like to thank the DSTC advisory committee and those on the DST mailing list for their invaluable contributions. The authors also thank Zhuoran Wang for providing a baseline tracker. Finally thanks to SIGdial for their endorsement, SLT for providing a special session, and the participants for their hard work in creating high quality submissions.

6. REFERENCES

[1] Jason D Williams, Antoine Raux, Deepak Ramachadran, and Alan Black, "The Dialog State Tracking Challenge," in *Proceedings of SIGDIAL*, August 2013.

[2] Matthew Henderson, Blaise Thomson, and Jason D Williams, "The Second Dialog State Tracking Challenge," in *Proceedings of SIGDIAL*, 2014.

[3] Sungjin Lee and Maxine Eskenazi, "Recipe For Building Robust Spoken Dialog State Trackers: Dialog State Tracking Challenge System Description," in *Proceedings of SIGDIAL*, 2013.

[4] Sungjin Lee, "Structured Discriminative Model For Dialog State Tracking," in *Proceedings of SIGDIAL*, 2013.

[5] Hang Ren, Weiqun Xu, Yan Zhang, and Yonghong Yan, "Dialog State Tracking using Conditional Random Fields," in *Proceedings of SIGDIAL*, 2013.

[6] Zhuoran Wang and Oliver Lemon, "A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information," in *Proceedings of SIGDIAL*, 2013.

[7] Matthew Henderson, Blaise Thomson, and Steve Young, "Deep Neural Network Approach for the Dialog State Tracking Challenge," in *Proceedings of SIGDIAL*, 2013.

[8] Matthew Henderson, Blaise Thomson, and Steve Young, "Word-Based Dialog State Tracking with Recurrent Neural Networks," in *Proceedings of SIGDIAL*, 2014.

[9] Jason D Williams, "Multi-domain learning and generalization in dialog state tracking," in *Proceedings of SIGDIAL*, 2013.

[10] Jason D Williams, "Web-style ranking and SLU combination for dialog state tracking," in *Proceedings of SIGDIAL*, 2014.

[11] Matthew Henderson, Blaise Thomson, and Jason Williams, "Dialog State Tracking Challenge 2 & 3 Handbook," camdial.org/~mh521/dstc/, 2013.

[12] Filip Jurccek, Blaise Thomson, and Steve Young, "Natural actor and belief critic: Reinforcement algorithm for learning parameters of dialogue systems modelled as POMDPs.," *TSLP*, vol. 7, 2011.

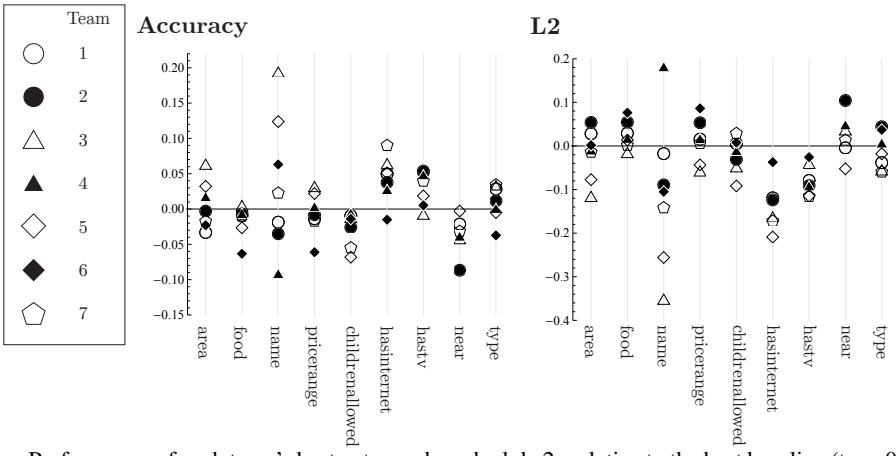
[13] Matthew Henderson, Milica Gašić, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young, "Discriminative Spoken Language Understanding Using Word Confusion Networks," in *Spoken Language Technology Workshop, 2012. IEEE*, 2012.

[14] Zhuoran Wang and Oliver Lemon, "A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information," in *Proceedings of SIGDIAL*, 2013.

[15] Ronnie Smith, "Comparative Error Analysis of Dialog State Tracking," in *Proceedings of SIGDIAL*, 2014.

Appendix A: Featured results of evaluation

		Tracker Inputs		Joint Goal Constraints			Search Method			Requested Slots		
Team	**Entry**	**SLU**	**ASR**	**Acc.**	**L2**	**ROC**	**Acc.**	**L2**	**ROC**	**Acc.**	**L2**	**ROC**
0	0	✓		0.555	0.860	0.000	0.922	0.154	0.000	0.778	0.393	0.000
	1	✓		0.556	0.750	0.000	0.908	0.134	0.000	0.761	0.435	0.000
(*baselines*)	2	✓		0.575	0.744	0.000	0.966	0.067	0.000	0.698	0.562	0.000
	3	✓		0.567	0.691	0.000	0.967	0.062	0.000	0.767	0.417	0.000
1	0	✓		0.561	0.761	0.000	0.962	0.077	0.000	0.778	0.393	0.000
	1	✓		0.561	0.761	0.000	0.962	0.077	0.000	0.778	0.393	0.000
	2	✓		0.559	0.736	0.000	0.963	0.097	0.000	0.774	0.401	0.000
	3	✓		0.561	0.733	0.000	0.963	0.097	0.000	0.774	0.401	0.000
2	0	✓	✓	0.585	0.697	0.000	0.965	0.114	0.171	0.929	0.121	0.061
	1	✓	✓	0.529	0.741	0.000	0.924	0.123	0.279	0.931	0.122	0.106
	2	✓	✓	0.555	0.677	0.158	0.950	0.088	0.247	0.938	0.105	0.062
	3	✓	✓	0.582	0.639	0.148	**0.970**	0.065	0.141	0.938	0.138	0.369
	4	✓	✓	0.574	0.650	0.152	0.966	0.073	0.162	0.939	0.111	0.387
3	0	✓	✓	**0.646**	0.538	0.169	0.966	**0.061**	0.434	0.943	0.091	0.441
	1	✓	✓	0.645	**0.534**	0.172	0.966	**0.061**	0.434	0.943	0.091	0.441
	2		✓	0.616	0.565	0.179	0.966	0.061	0.400	0.939	0.100	0.309
	3		✓	0.615	0.564	0.190	0.966	0.061	0.400	0.939	0.100	0.309
4	0	✓	✓	0.630	0.627	0.072	0.853	0.272	0.255	0.923	0.136	0.355
	1	✓	✓	0.630	0.627	0.072	0.853	0.272	0.255	0.923	0.136	0.355
	2	✓	✓	0.630	0.627	0.072	0.853	0.272	0.255	0.923	0.136	0.355
	3	✓	✓	0.630	0.627	0.072	0.853	0.272	0.255	0.923	0.136	0.355
5	0		✓	0.610	0.556	0.258	0.968	0.091	0.258	0.945	**0.090**	0.471
	1		✓	0.587	0.634	0.181	0.958	0.068	0.443	0.944	0.096	0.000
	2		✓	0.588	0.639	0.160	0.961	0.063	0.272	**0.949**	0.090	0.000
6	0	✓		0.507	0.736	0.110	0.927	0.120	0.198	0.908	0.157	0.192
	1	✓		0.507	0.739	0.109	0.927	0.122	0.220	0.909	0.156	0.190
	2	✓		0.503	0.743	0.111	0.927	0.120	0.198	0.908	0.157	0.192
	3	✓		0.503	0.746	0.110	0.927	0.122	0.218	0.909	0.156	0.190
7	0	✓		0.572	0.677	0.080	0.956	0.113	0.048	0.933	0.104	0.463
	1	✓		0.576	0.652	0.055	0.957	0.116	0.154	0.938	0.101	0.448
	2	✓		0.575	0.667	0.063	0.934	0.130	0.183	0.929	0.107	0.470
	3	✓		0.570	0.658	0.084	0.957	0.116	0.154	0.938	0.101	0.448



Performance of each team's best entry under schedule 2a relative to the best baseline (team0, entry2) for the goal constraint on every slot. Recall a lower L2 score is better.