2018-11-08

# A Multi-Task Approach to Incremental Dialogue State Tracking

Anh Duong Trinh
*Technological University Dublin*, anhduong.trinh@tudublin.ie

Robert J. Ross
*Technological University Dublin*, robert.ross@tudublin.ie

John D. Kelleher
*Technological University Dublin*, john.d.kelleher@tudublin.ie



## Recommended Citation

Trinh, A., Ross, R. & Kelleher, J. (2018). A multi-task approach to incremental dialogue state tracking. *SEMDIAL 2018 (AixDial): the 22nd workshop on the Semantics and Pragmatics of Dialogue*, Aix-en-Provence, France, 8-10 November, 2018. doi:10.21427/cvkg-0p89

# A Multi-Task Approach to Incremental Dialogue State Tracking

**Anh Duong Trinh***, **Robert J. Ross***, **John D. Kelleher****
* School of Computing
** Information, Communications & Entertainment Institute
Dublin Institute of Technology, Ireland
anhduong.trinh@mydit.ie, {robert.ross, john.d.kelleher}@dit.ie

## Abstract

Incrementality is a fundamental feature of language in real world use. To this point, however, the vast majority of work in automated dialogue processing has focused on language as turn based. In this paper we explore the challenge of incremental dialogue state tracking through the development and analysis of a multi-task approach to incremental dialogue state tracking. We present the design of our incremental dialogue state tracker in detail and provide evaluation against the well known Dialogue State Tracking Challenge 2 (DSTC2) dataset. In addition to a standard evaluation of the tracker, we also provide an analysis of the Incrementality phenomenon in our model's performance by analyzing how early our models can produce correct predictions and how stable those predictions are. We find that the Multi-Task Learning-based model achieves state-of-the-art results for incremental processing.

## 1 Introduction

In recent years significant progress has been made in Dialogue State Tracking. Early work on rule-based updates to dialogue state has now widely been replaced with variants on data driven systems. While probabilistic systems dominated the early work in this area, error-based learning systems such as those based on Deep Neural Network architectures are now common place. More formally we can think of Dialogue Tracking Components as being split between Rule Based, Generative and Discriminative methods. Discriminative models based on Partially Observable Markov Decision Process (POMDP) are found to yield very high results. Currently, many architectures yield state-of-the-art type performance including Structure Discriminative Modelling (Lee, 2013), web-style ranking (Williams, 2014), Recurrent Neural Networks (RNN) (Henderson et al., 2014b; Mrksic et al., 2015), Convolutional Neural Networks (CNN) (Shi et al., 2016), attention mechanism (Hori et al., 2016) and hybrid modelling (Dernoncourt et al., 2016; Vodolan et al., 2017).

While recent progress in Dialogue State Tracking (DST) is considerable, the vast amount of work to date treats DST, like dialogue management in general, as a turn-based phenomenon. In other words, systems wait for a user to pass the turn back to the system before attempts are made to update the dialogue state. Such an approach ignores the fact that a turn can have multiple functional contributions (Levinson, 1983; Bunt, 2011), and that in fluid natural interactions an interlocutor will often provide within-turn feedback to their dialogue partner (Schlangen and Skantze, 2009; Hough et al., 2015). Given the importance of incremental updates and feedback, in our work we are focused on the longer term problem of incremental (i.e. word by word) dialogue management and dialogue tracking in particular.

In recent years the community has begun to address the problem of incremental dialogue modeling with the proposal of a number of DST models that include incremental encoders (Jagfeld and Vu, 2017; Platek et al., 2016; Zilka and Jurcicek, 2015). However, within this subfield of DST research significant challenges remain to be overcome. Of these challenges we believe the most significant is a common presumption of independence between target labels for dialogue state. An example of this independence can be seen in (Zilka and Jurcicek, 2015) where the authors developed a separate model for each DST subtask. While such an assumption is useful in simplifying the underlying model, it does not correspond to the reality of modeling user intents where elements of user intent are often inter-related (Williams et al., 2016; Oraby et al., 2017).

To consider the challenge of non-independence of goals, it is useful to view DST as a machine learning task. For our current purposes we can interpret Dialogue States as combinations of slot-value pairs that in turn can be considered instances of a multi-label classification task. For example, in the flight booking domain the system always requires certain slots such as *departure*, *destination* and *date* to be filled before offering suitable options. These pieces of information are often given in just one utterance in various forms. We see the motivation of a Multi-Task Learning (MTL) (Caruana, 1997) approach in investigating task relatedness and variable correlation, and also boosting the performance on several related tasks. The system benefits a lot from tracking multiple dialogue states rather than single dialogue state.

In our work presented in this paper we explore a Multi-Task Model as a novel approach to solving the dialogue state tracking problem for incremental analysis. We present our model design including details on input representations in section 3, before detailing our experiments and validation results in section 4. In section 5 we provide an evaluation in terms of common metrics as well as an incremental performance evaluation to help address our main questions around the incrementality phenomenon; specifically, how early can our model predict correct the useful dialogue state? and what is the quality of those predictions in Dialogue State Tracking? Following this, in section 6 we discuss several similar approaches to the DST tasks. Finally, in section 7 we conclude and outline future work. We begin with a brief detailing of approaches to Multi-Task Learning in the context of dialogue state modeling.

## 2 Multi-Task Learning

Within the Machine Learning discipline, Multi-Task Learning (MTL) (Caruana, 1997) is a modelling approach where we use shared useful information between related tasks in order to achieve better performance across these tasks. This is in contrast to the traditional multi-label approach to classification where we train multiple models for multiple tasks and do not explicitly incorporate useful feedback across tasks. In MTL, shared parameters and representations allow the model to look at the training process of all tasks at the same time and consider the useful signals in order to boost the end performance. In other words, an MTL approach aims to optimize more than one metric at the same time.

The natural motivation of a Multi-Task Learning approach comes from mimicking human behaviours as they are always combinations of single actions. On the other hand, from Machine Learning aspects MTL can be viewed as a form of inductive transfer. It is also related to other areas in ML such as transfer learning. The significant difference between Transfer Learning and MTL is however that Transfer Learning aims to use knowledge of related tasks to improve the target task while MTL uses multiple tasks to help each other. Multi-Task Learning has been applied successfully to many fields of Machine Learning including Natural Language Processing (NLP) for sequential data (Cheng et al., 2015; Rei, 2017) and Speech Recognition (Deng et al., 2013).

In the context of slot-filling Dialogue systems, dialogue states are presented as joint sets of slot-value pairs across domains, or in the case of probabilistic systems, these are probability distributions over slots. In our current work we make use of the Dialogue State Tracking Challenge 2 (DSTC2) dataset. Within this a dialogue state is a combination of probability distributions over multiple slots such as *food* and *price range*, and logistic regression over requested slots such as *address* and *phone number*. Therefore, DST tasks can be classified as multi-label learning. This is the case of Multi-Task Supervised Learning when different tasks share the same training data.

In general the MTL approach enhances the correlation of variables through the shared training signals. In the DSTC2 restaurant information domain, it is the correlation between the slots and the tasks that we wish to take advantage of. For example users are more likely to provide the type of food with preferred price range and area, or tell the system the restaurant's name before asking for address or phone number. Keeping this in mind we have a strong motivation to apply an MTL approach to solving incremental DST.

## 3 Dialogue State Tracking Model

### 3.1 Dataset

In order to explore the particular difficulties of incremental dialogue state processing, we make use of the second Dialogue State Tracking Challenge (DSTC2) dataset (Henderson et al., 2014a). DSTC2 provides a common testbed for explicit research on Dialogue State Tracking tasks. The dataset is split into 3 sets of dialogues: 1612 dialogues in a training dataset, 506 in a development (validation) set and 1117 in a test set. A dialogue in DSTC2 contains up to 30 turns consisting of 2 parts: a machine dialogue act in a semantic representation format, and user input in ASR utterance and preprocessed SLU (Spoken Language Understanding) format. The DSTC2 required trackers to produce dialogue states for each turn.

Dialogue States of each turn in DSTC2 contain three components, each of which can be thought of as a grouping of target variables. *Joint Goals*: the goal constraint captures what users want, such as type of food and preferred price range. *Search Method:* captures the manner in which users interact with the system, e.g. users can issue clear constraints such as `'korean food'` or request alternative options. Finally, *Requested Slots* capture any user request for information from the system.

### 3.2 Model Architecture

Our underlying approach is based on a Recurrent Neural Network (RNN) architecture. Given our focus on incremental analysis, we process dialogue content in a word-by-word manner where a set of classifiers predict class labels after each word in the utterance. Moreover, we evaluate two MTL-based Deep RNN architectures for the task. Each architecture, visualized in Fig. 1, has 4 layers including an input, an output and two hidden RNN layers . The model presented in this paper is a significant improvement of our early work (Trinh et al., 2017).

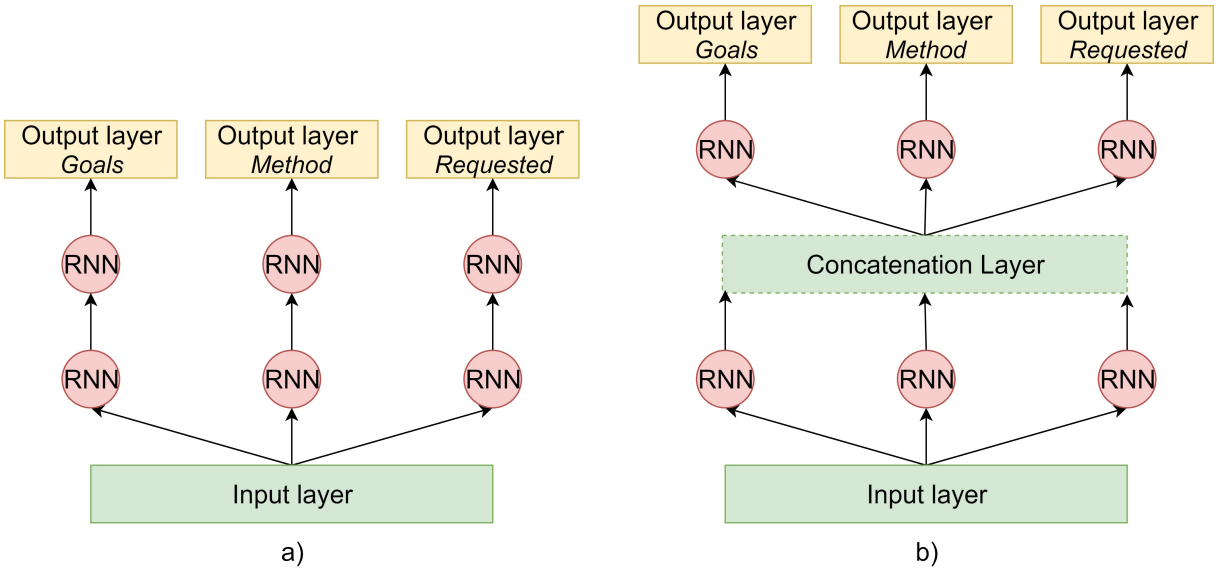


Figure 1: Multi-Task Learning Deep RNN-based Dialogue State Tracking Models. *Goals* denotes the Joint Goals task including 4 informable slot subtasks, *Method* denotes the Search Method task, and *Requested* stands for the Requested Slots task.

At each time step, we preprocess dialogue input into a vector representation (see 3.3 for detail) and feed this vector into the networks. At this point there are two alternatives to how our MTL-based models predict the output. One scenario is that model *a* uses task-specific RNNs and classifiers to predict the output of the tasks. Another more complex scenario is based on the model *b* processing mechanism. At the 1st layer, all RNN cells process the input vector and produce multiple hidden states. Then these hidden states are concatenated into a joint vector representation, that we hypothesize is the representation of the whole dialogue until the current time. In this approach model *b* then uses task-specific RNNs and

classifiers to produce predictions based on this universal dialogue representation. In practice model *b* is a true MTL approach in that individual task learning can influence learning for the related tasks through the shared layers. Model *a* while being a multi-task architecture in the broad sense by combining the training process does not share weights across tasks, and is thus unable to leverage shared modeling at any layer except initial input encoding layers.

At the output layer all predicted outcomes are combined to form dialogue states. A *Joint Loss Function* of all the subtasks is calculated and used to backpropagate through the whole networks. In these models the network parameters are updated according to the task to which they contribute.

The processing mechanism summary of our trackers is presented in Table 1.

| | Model *a* | Model *b* |
|---|---|---|
| Output layer | $y^t_{food} = P_{food}(h^t_{2,food})$ | $y^t_{food} = P_{food}(h^t_{2,food})$ |
| Hidden layer 2 | $h^t_{2,food} = RNN_{food}(h^t_{1,food}, s^{t-1}_{2,food})$ | $h^t_{2,food} = RNN_{food}(h^t_1, s^{t-1}_{2,food})$ |
| Hidden layer 1 | $h^t_{1,food} = RNN_{food}(x^t, s^{t-1}_{1,food})$ | $h^t_1 = \sum^{\oplus}_k h^t_{1,k} = \sum^{\oplus}_k RNN_k(x^t, s^{t-1}_{1,k})$ |
| Input layer | $x^t$ | $x^t$ |

Table 1: Processing mechanism of the trackers for slot *food* at the time step $t$. $x^t$ and $y^t_{food}$ denote the input and output of slot *food* at time step $t$. $h^t_i$ and $s^t_i$ are the hidden state and RNN inner memory of layer $i$ at time $t$. $\sum^{\oplus}$ denotes concatenation operation on multiple vector representations.

### 3.3 Input Representations

In our representation approach, the dialogue input of a turn consists of two parts: the machine dialogue act and the user utterance. In order to process dialogue data incrementally we treat the whole dialogue as a sequence of words or tokens. Each turn in the dialogue is presented in a sequence starting with token *<mact>*, which stands for machine dialogue act, following by the utterance embedded into vectors by Word2Vec, and ending with token *<eos>*.

The machine dialogue act is given in the format $act(slot = value)$. We use a similar technique to Henderson et al. (2014b) to extract features to capture the local semantics of these acts. The result of this is a machine dialogue act with about 2000 dimensions. We the apply auto-encoder style training to develop a distributed representation of machine dialogue acts across 300 dimensions. This encoded vector is concatenated with word embedding vectors at the beginning of each turn. For the rest of the utterance we use a zero vector in place of the dialogue act vector.

In order to improve the performance of our MTL-based trackers we also investigate a number of techniques to improve the quality of the word embeddings. The three variants considered here are described below. It should be noted that in each case we assumed a dimensionality of 300 for each word embedding type.

- **Online-trained Word Embeddings** We train word embeddings along with the training process of the whole networks. The motivation for this word embedding approach is a hypothesis that it is useful for the network to learn all words in the context of dialogues and dialogue states.

- **Pre-trained Word Embeddings** Due to the nature of the dataset, the vocabulary size is relatively small. We hypothesize that the pre-trained word embedding from a large corpus such as Wikipedia or Twitter might give better representations and reduce the training time of the model. We choose Word2Vec developed by Mikolov et al. (2013) for this purpose.

- **Combined Word Embeddings** We also investigated the option of combining pre-trained word embeddings and our model trained word embeddings to give the model the benefit of information from the dialogue domain as well as general context.

## 4 Experiments

In this section we provide the details of our experiment methodology.

### 4.1 Experiment Setup

In the proposed models we configured all RNNs cells with Long Short-Term Memory units (Hochreiter and Schmidhuber, 1997) of hidden size 128 and drop out rate for training 0.2. The standard deviation was set to 0.05 for the truncated normal initializer, and the initial value was set to 0.001 for the constant initializer. We trained the models with mini-batches of 10. We implement our MTL-based models in TensorFlow platform[1] (Abadi et al., 2015) and trained using the Adam Optimizer (Kingma and Ba, 2015) to minimize a Joint Loss Function. We use the cross-entropy loss function for each individual subtask.

For development we train our models on the training dataset and used the development dataset to evaluate and consider the best training parameters for the DSTC2 tasks. To prevent overfitting we used a number of techniques: drop out training rate, early stopping, and averaging Neural Networks weights between the multiple tasks. To be noted, our MTL models have shared layers, that have parameters trained according to all tasks. We validated our model every 100 training steps. Furthermore, we trained each model 10 times with different initializations and ensembled the output. We subsequently applied the best training parameters based on ensembled validation results to test set for this paper's result and discussion.

Model performance is evaluated using two common feature metrics that are taken as standard for work on the DSTC2 dataset: **Accuracy** measures how often a tracker predicts true dialogue states in the form of the top hypothesis; and **L2 norm** measures the squared norm $l^2$ between the correct label and predicted distribution (Henderson et al., 2014a). The better tracker must have higher accuracy and lower L2 norm in evaluation.

### 4.2 Embeddings Selection

During the development phase we evaluated a number of options to increase the performance from the raw test data. This included the evaluation of a number of embedding options (outlined above), and testing the inclusion of manual transcriptions data alongside ASR results. The result on development dataset (Table 2) is reported in a grid table of both model architectures with all Word2Vec and Input options. We also included the best baseline result provided by DSTC2 organizers (Henderson et al., 2014a) on the development dataset below for reference.

| DST Model | Input Options | Word Embeddings | | |
|---|---|---|---|---|
| | | Online-trained | Pre-trained | Combined |
| Model *a* | ASR | 0.687 | 0.687 | 0.688 |
| | ASR + Label | **0.694** | 0.682 | 0.691 |
| Model *b* | ASR | 0.683 | 0.687 | 0.675 |
| | ASR + Label | **0.697** | 0.688 | 0.684 |
| Baseline | ASR | 0.623 | | |

Table 2: Performance of our proposed models and the best baseline system on DSTC2 development dataset during the experiment phase. The performance is reported in Accuracy value for the Joint Goals task. The DSTC2 baseline system is non-incremental and rule-based.

The comparison of different word embeddings shows that the systems can learn similarly in different word vector spaces. However, using pre-trained Word2Vec reduces the number of parameters to learn in the training process, therefore the training time is reduced. On the other hand, both models perform better when we improve the data quality by including manual transcriptions into the training data. The best results on the development dataset were achieved by the systems trained on the expanded dataset with their own custom trained word embeddings. For test evaluation we selected these options and deployed for testset evaluation.

[1] Version 1.5, retrieved from `https://www.tensorflow.org/`

## 5 Results and Discussions

We demonstrate the performance of our models against DSTC2 test dataset and compare them with the state-of-the-art incremental systems that we know of (Table 3). The results are reported on the Joint Goals, Requested Slots, and Search Method tasks with two evaluation metrics Accuracy and L2. The reported results are sorted in the order of descending Joint Goals Accuracy. In the bottom of the table we include the performance of the best turn-based and the best baseline systems to provide a comparison of Incremental and non-Incremental approaches.

| DST Model | Joint Goals | | Requested Slots | | Search Method | |
|---|---|---|---|---|---|---|
| | Acc. | L2 | Acc. | L2 | Acc. | L2 |
| EncDec Framework (Platek et al., 2016) | **0.730** | – | – | – | – | – |
| MTL Model *b* (this work) | 0.728 | 0.458 | **0.980** | 0.035 | **0.946** | 0.093 |
| MTL Model *a* (this work) | 0.720 | 0.498 | 0.978 | 0.037 | 0.944 | 0.096 |
| LecTrack (Zilka and Jurcicek, 2015) | 0.72 | 0.64 | 0.97 | 0.06 | 0.93 | 0.14 |
| CNET Tracker (Jagfeld and Vu, 2017) | 0.714 | – | 0.972 | – | – | – |
| IJM Tracker (Trinh et al., 2017) | 0.707 | 0.545 | 0.975 | 0.047 | 0.940 | 0.114 |
| Best turn-based system | 0.796 | 0.338 | – | – | – | – |
| Best baseline system | 0.719 | 0.464 | 0.879 | 0.206 | 0.867 | 0.210 |

Table 3: Performance of our proposed models and state-of-the-art incremental systems on DSTC2 test dataset. The evaluation metrics are Accuracy (Acc.) and L2 norm. The best turn-based system is Hybrid Tracker (Vodolan et al., 2017). The best baseline system is Focus baseline (Henderson et al., 2014a).

The result on the DSTC2 testset shows that our MTL-based models achieve state-of-the-art level results among Incremental Dialogue State Trackers. Our trackers are capable of predicting full dialogue states including all informable slots, requested slots and search methods. To the best of our knowledge the EncDec Framework (Platek et al., 2016) is the best Incremental Tracker on the DSTC2 dataset. However, this tracker was implemented to track informable slots only, meaning the full dialogue states are not reported. Looking at the difference of Joint Goals result between EncDec Framework and our MTL-based model, the margin is very small, while our model is capable of producing full dialogue states with state-of-the-art results in Requested Slots and Search Method tasks.

Comparing the two MTL-based models of this work, we see that model *b* generally performs better than model *a* in all tasks. We would argue that the reason of this result lies in the shared hidden RNN layer of model *b*. We use multiple RNNs to extract information from dialogue input by multiple channels, that are separate from each other, and concatenate their output to form a dialogue joint representation. These RNNs are updated based on backpropagation of the whole Neural Networks according to the errors. We believe that this particular architecture ensures the control over correlation between slots in the domain, while still keeping the independence of prediction by using task-specific RNN layer and classifiers. In our attention, the number of parameters of model *a* is much bigger than model *b*, therefore the training time is also longer.

While neither of our models improve on the EncDec framework, it is notable that the performance improvement that we observe in Model b over Model a would suggest that the performance of EncDec may be improved if a Multi-Task approach leveraging Requested Slots and Search Method is taken.

### 5.1 Incremental Processing Analysis

Incremental dialogue processing requires accuracy as early as possible during an interaction. Given this we provide an analysis of accuracy over time rather than waiting for the well-defined end of a turn. Given the Joint Goals task is the most crucial and challenging task in DSTC2, we provide an analysis specifically for that task. Table 4 provides the results for this analysis where Model a and Model b are considered. Unfortunately it is not possible to repeat this analysis for the EncDec framework and similar works since no previous study on these frameworks has considered incremental accuracy. Performance

is measured in Accuracy of Joint Goals along the length of utterances. As the utterance length varies from 1 to 24 words, we chose to scale between 0-100% of utterance length.

| Length | 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Model *a* | 0.468 | 0.468 | 0.480 | 0.494 | 0.501 | 0.513 | 0.522 | 0.546 | 0.580 | 0.623 | 0.720 |
| Model *b* | 0.471 | 0.471 | 0.482 | 0.496 | 0.505 | 0.523 | 0.536 | 0.557 | 0.591 | 0.634 | 0.728 |

Table 4: Incremental performance of MTL-based models. Performance is measured in Accuracy.

These results show that the trackers have the ability to predict the dialogue state at a reasonable rate long before the utterances is complete. Even with less than 50% of the utterance considered, accuracy levels are over 50%. Correct dialogue states, even at very early points in the utterance, can be produced. Empirically we believe this is due to state carried over from previous turns - note that our modeling approach, like similar works, does not reset at a turn boundary. It is also noteworthy that there is a considerable jump in accuracy between 90% and 100% of the utterance being consumed.

It is also notable from the results that MTL model *b* consistently outperfored MTL model *a* at every time step. While the performance improvement was slight, we believe this supports the assertion that a true multi-task learning approach where information is shared at multiple points in the network can improve overall goal performance.

In Appendix B we present the Incremental performance of our trackers on the dialogues in testset. We select the dialogues randomly for some specific scenarios where our models perform both well and badly.

### 5.2 Error Analysis

In this subsection we provide a more detailed error analysis on the incremental result. As we know that user utterances give different information at different time. According to the Henderson et al. (2014a) user intents of slot *food* change most frequently, up to 40.9% dialogues in the testset, and it is the most difficult to track. Henderson et al's analysis was carried on the dialogue level; however, we expect that user intent can also change on turn and word level.

To quantify this hypothesis, we carried out a small analysis on DSTC2 testset regarding the informable slots to monitor in detail the performance of our trackers (see Table 5). The analysis shows that in the DSTC2 testset the total number of turns is 9890, in which there are 1596 (16.14%) turns where users change the food, 932 (9.42%) turns where the price range value is changed, 1046 (10.58%) turns with the change in area, and only 9 (0.09%) with regard of slot name.

| Informable Slot | Food | Price | Area | Name |
|---|---|---|---|---|
| Turns | 9890 | | | |
| Model *a* | 0.847 | 0.881 | 0.919 | 0.995 |
| Model *b* | 0.848 | 0.893 | 0.920 | 0.995 |
| Turns with change | 1596 | 932 | 1046 | 9 |
| Model *a* | 0.780 | 0.767 | 0.856 | 0.000 |
| Model *b* | 0.786 | 0.804 | 0.870 | 0.000 |

Table 5: Detailed performance evaluation of our proposed models on the informable slots. The results are reported in Accuracy.

We observe that our MTL trackers perform well in tracking three out of four informable slots, that are *food*, *price range* and *area*, both in general and when the user intentions change. On the other hand, the trackers overfit in tracking slot *name*, that can be explained by the lack of training data as we mentioned above that less than 10% of total turns that users mention the name of restaurants. That being said, our trained models always assign the value *'none'* for slot *name*. We also see that our model *b* outperforms model *a* marginally in detecting the goal change per slot.

We present our analysis on the Incrementality performance regarding the slot *food*, the most difficult slot to track, in the format of graphs in Fig. 2. The graph on the left shows the first moment our trackers predict correct *food* value to answer the question "How early can our models pick up the right food value?". The one on the right shows the stability of *food* predictions, that the earliest moment of correct prediction that can be kept until the end of turns. All the results are reported by counting the number of turns.

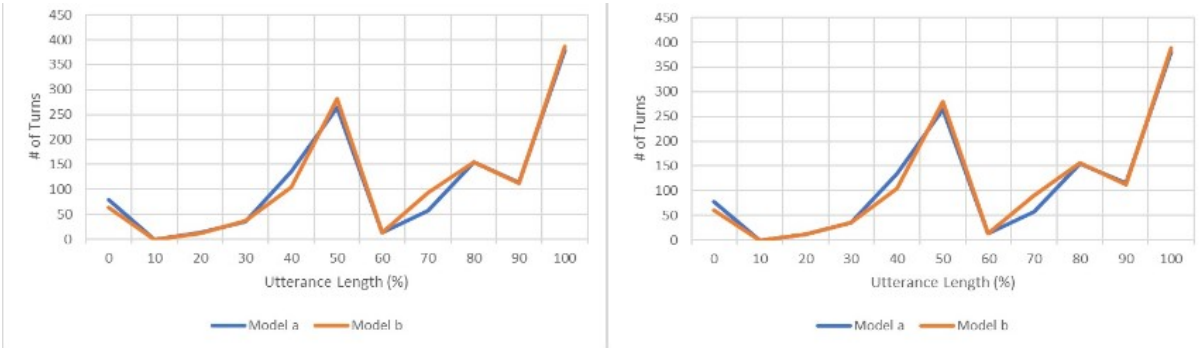


Figure 2: Error Analysis of slot *food* predictions according to number of turns with correct prediction.

The nearly identical patterns in graphs show that our trackers are capable of tracking *food* value as early as reaching the middle of utterance. These predictions are of good quality as they are correct and kept until the end of utterance to produce the end-of-turn dialogue states.

In detail, the analysis shows that our models are capable of picking up key words in utterances to predict particular values. This word-based mechanism is similar to the idea to extract ASR features proposed by Henderson et al. (2014b) . We also realize that user intent in DSTC2 dataset changes on the turn level, but not on the word level. For example, in one turn the user would say "I'm looking for Chinese food" rather than "I'm looking for Chinese food, no, wait, Italian food".

On a related note, the analysis shows the peak of prediction for the *food* slot at 50%. Looking at the data, this peak can be explained by the patterns of user utterance. The system's question for *food* slot is set up to "What food do you want?". Naturally, the user would respond "Italian food", meaning the value is predicted exactly in the middle of utterance.

There exist many factors that influence the trackers' prediction ability such as ASR and SLU errors (see Appendix B), and many types of errors that the trackers produce. For detailed comparative error analysis of DSTC2 models, read (Smith, 2014).

## 6 Related Work

To date, the state-of-the-art results in DST are achieved by non-incremental models (Henderson et al., 2014b; Vodolan et al., 2017). Both of these models use RNNs to process dialogues on the turn-based level. The work published by Henderson et al. (2014b) is notable for the novelty and high performance. Its technique of extracting word features of ASR input has shown the advantages against other feature extraction techniques. This technique is also adopted in the Hybrid tracker by Vodolan et al. (2017). While the Word-based tracker using only RNNs by Henderson et al. could achieve the highest performance accuracy at the time, the Hybrid tracker by Vodolan et al. used RNNs and a set of hand-crafted rules to improve the results. These approaches' results are not yet overcome by Incremental models. We adopt the feature extraction technique into our model to encode the machine dialogue acts.

The number of Incremental DST models to our knowledge is currently limited to LecTrack (Zilka and Jurcicek, 2015), EncDec Framework (Platek et al., 2016), and CNET tracker (Jagfeld and Vu, 2017). Among these trackers, LecTrack and EndDec Framework process dialogues on the word-based level, that can be compared directly to our work. There are several differences between our MTL models and Lectrack and EncDec Framework. First of all, we handle machine dialogue acts or response differently. In our MTL models, we encode these acts into only one token and engage them when it is the machine

turn. While the other two models straighten them into sequences of words to make the dialogues continuous word sequence. Secondly, we apply different mechanism to predict the dialogue states. Zilka and Jurcicek (2015) developed multiple single models to predict outcomes of each slot, then combine the predictions into dialogue states. Platek et al. (2016) developed an Encoder-Decoder language model to predict the slot value in a particular order. Their model is limited to predicting the joint goal state for the three informable slots only and does not include other two subtasks. Different from these models, our MTL model is capable of predicting all slots simultaneously.

Currently, we can handle only the best ASR hypothesis in the data, while the prediction might possibly be improved by processing multiple ASR hypotheses. Jagfeld and Vu (2017) have been able to improve this limitation by integrating a confusion network into dialogue state tracking. However, confusion networks generate more errors in ASR of DSTC2 than the live recognizers, therefore they reduce the accuracy of the outcome. Even though their approach is the only one of its kind, the result is not yet state-of-the-art.

Apart from approaches mentioned above, there are numerous models introduced to solve DST problems. Many of those models are also RNN-based with different architectures and techniques (Jang et al., 2016; Hori et al., 2016; Yoshino et al., 2016). However their results are reported against other tasks, that we cannot compare to our model directly. On the other hand, there are also various approaches proposed for DSTC2 tasks that are not based on RNN but achieve notable results (Williams, 2014; Sun et al., 2014; Kadlec et al., 2014; Yu et al., 2015; Fix and Frezza-Buet, 2015; Lee and Stent, 2016; Mrksic et al., 2017).

## 7 Conclusion

This paper presents Incremental approaches to Dialogue State Tracking using Multi-Task Learning techniques. To our knowledge our work is the only one applying MTL-based models in DST tasks. The results suggest that our models achieve state-of-the-art results among incremental trackers. To address the importance of Incremental phoenomenon in dialogue processing, we also report a detailed error analysis as the measure of quality on the incremental DST phenomenon. Furthermore, our MTL-based trackers show that the correlations between in-domain slots in dialogues processing are essential and should be learned in dialogue.

Our models work well on the Incrementality phenomenon. First, our work predicts the correct values by recognising key words at the early point of the time sequence (see Appendix B). Second, our predictions are stable through out the dialogues. However, there is still room to improve our work that we would like to apply our approach to more complex dialogue data, where user intention is dynamic within utterances.

To date, the Incrementality of all incremental models is limited to turn-based analysis due to the limit in dataset. There is no dataset yet for evaluating incremental dialogue state trackers. Therefore to continue investigating the incremental mechanism for dialogue state tracking, we are considering reannotating the DSTC2 data into word-level annotated data. In the future we also plan to put more effort in investigating useful incremental Natural Language features for dialogue modelling.

## Acknowledgements

This research was supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

## References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals,

Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.

Harry Bunt. 2011. Multifunctionality in dialogue. *Computer Speech and Language*, 25(2):222–245.

Rich Caruana. 1997. Multi-task Learning. *Machine Learning*, 28:41–75.

Hao Cheng, Hao Fang, and Mari Ostendorf. 2015. Open-Domain Name Error Detection using a Multi-Task RNN. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 737–746.

Li Deng, Geoffrey Hinton, and Brian Kingsbury. 2013. New types of deep neural network learning for speech recognition and related applications: an overview. In *Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8599–8603.

Franck Dernoncourt, Ji Young Lee, Trung H. Bui, and Hung H. Bui. 2016. Robust Dialog State Tracking for Large Ontologies. In *Proceedings of the International Workshop on Spoken Dialogue Systems, IWSDS 2016*.

Jeremy Fix and Herve Frezza-Buet. 2015. YARBUS: Yet Another Rule Based belief Update System. Technical report, CentraleSupelec.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014a. The Second Dialog State Tracking Challenge. In *Proceedings of the SIGDIAL 2014 Conference*, pages 263–272.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014b. Word-Based Dialog State Tracking with Recurrent Neural Networks. In *Proceedings of the SIGDIAL 2014 Conference*, pages 292–299.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Takaaki Hori, Hai Wang, Chiori Hori, Shinji Watanabe, Bret Harsham, Jonathan Le Roux, John R. Hershey, Yusuke Koji, Yi Jing, Zhaocheng Zhu, and Takeyuki Aikawa. 2016. Dialog State Tracking With Attention-Based Sequence-To-Sequence Learning. In *Proceedings of 2016 IEEE Workshop on Spoken Language Technology*, pages 552–558.

Julian Hough, Casey Kennington, David Schlangen, and Jonathan Ginzburg. 2015. Incremental Semantics for Dialogue Processing : Requirements , and a Comparison of Two Approaches. In *Proceedings of the 11th International conference on Computational Semantics*, pages 206–216.

Glorianna Jagfeld and Ngoc Thang Vu. 2017. Encoding Word Confusion Networks with Recurrent Neural Networks for Dialog State Tracking. In *Proceedings of the 1st Workshop on Speech-Centric Natural Language Processing*, pages 10–17.

Youngsoo Jang, Jiyeon Ham, Byung-Jun Lee, Youngjae Chang, and Kee-eung Kim. 2016. Neural Dialog State Tracker for Large Ontologies by Attention Mechanism. In *Proceedings of 2016 IEEE Workshop on Spoken Language Technology*, pages 531–537.

Rudolf Kadlec, Miroslav Vodolan, Jindrich Libovicky, Jan Macek, and Jan Kleindienst. 2014. Knowledge-based Dialog State Tracking. In *Proceedings of 2014 IEEE Workshop on Spoken Language Technology*, pages 348–353.

Seokhwan Kim and Rafael E. Banchs. 2014. Sequential Labeling for Tracking Dynamic Dialog States. In *Proceedings of the SIGDIAL 2014 Conference*, pages 332–336.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.

Sungjin Lee and Amanda Stent. 2016. Task Lineages: Dialog State Tracking for Flexible Interaction. In *Proceedings of the SIGDIAL 2016 Conference*, pages 11–21.

Byung-Jun Lee, Woosang Lim, Daejoong Kim, and Kee-Eung Kim. 2014. Optimizing Generative Dialog State Tracker via Cascading Gradient Descent. In *Proceedings of the SIGDIAL 2014 Conference*, pages 273–281.

Sungjin Lee. 2013. Structured Discriminative Model For Dialog State Tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 442–451.

Stephen C. Levinson. 1983. *Pragmatics*. Cambridge University Press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 3111–3119.

Nikola Mrksic, Diarmuid O Seaghdha, Blaise Thomson, Milica Gasic, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2015. Multi-domain Dialog State Tracking using Recurrent Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 794–799.

Nikola Mrksic, Diarmuid O Seaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural Belief Tracker: Data-Driven Dialogue State Tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Shereen Oraby, Vrindavan Harrison, Amita Misra, Ellen Riloff, and Marilyn Walker. 2017. Are you serious?: Rhetorical Questions and Sarcasm in Social Media Dialog. In *Proceedings of the SIGDIAL 2017 Conference*, pages 310–319.

Ondrej Platek, Petr Belohlavek, Vojtech Hudecek, and Filip Jurcicek. 2016. Recurrent Neural Networks for Dialogue State Tracking. In *Proceedings of CEUR Workshop, ITAT 2016 Conference*, volume 1649, pages 63–67.

Marek Rei. 2017. Semi-supervised Multitask Learning for Sequence Labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 2121–2130.

Hang Ren, Weiqun Xu, and Yonghong Yan. 2014. Markovian Discriminative Modeling for Dialog State Tracking. In *Proceedings of the SIGDIAL 2014 Conference*, pages 327–331.

David Schlangen and Gabriel Skantze. 2009. A General, Abstract Model of Incremental Dialogue Processing. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 710–718.

Hongjie Shi, Takashi Ushio, Mitsuru Endo, Katsuyoshi Yamagami, and Noriaki Horii. 2016. A Multichannel Convolutional Neural Network For Cross-Language Dialog State Tracking. In *Proceedings of 2016 IEEE Workshop on Spoken Language Technology*, pages 559–564.

Ronnie W. Smith. 2014. Comparative Error Analysis of Dialog State Tracking. In *Proceedings of the SIGDIAL 2014 Conference*, pages 300–309.

Kai Sun, Lu Chen, Su Zhu, and Kai Yu. 2014. The SJTU System for Dialog State Tracking Challenge 2. In *Proceedings of the SIGDIAL 2014 Conference*, pages 318–326.

Anh Duong Trinh, Robert J. Ross, and John D. Kelleher. 2017. Incremental Joint Modelling for Dialogue State Tracking. In *Proceedings of the 21st Workshop on the Semantics and Pragmatics of Dialogue*, pages 176–177.

Miroslav Vodolan, Rudolf Kadlec, and Jan Kleindienst. 2017. Hybrid Dialog State Tracker with ASR Features. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 2, pages 205–210.

Jason D. Williams, Antoine Raux, and Matthew Henderson. 2016. The Dialog State Tracking Challenge Series: A Review. *Dialogue & Discourse*, 7(3):4–33.

Jason D. Williams. 2014. Web-style ranking and SLU combination for dialog state tracking. In *Proceedings of the SIGDIAL 2014 Conference*, pages 282–291.

Koichiro Yoshino, Takuya Hiraoka, Graham Neubig, and Satoshi Nakamura. 2016. Dialogue State Tracking using Long Short Term Memory Neural Networks. In *Proceedings of the International Workshop on Spoken Dialogue Systems, IWSDS 2016*.

Kai Yu, Kai Sun, Lu Chen, and Su Zhu. 2015. Constrained Markov Bayesian Polynomial for Efficient Dialogue State Tracking. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 23(12):2177–2188.

Lukas Zilka and Filip Jurcicek. 2015. Incremental LSTM-Based Dialog State Tracker. In *Proceedings of 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 757–762.

## Appendix A. State-of-the-art Dialogue State Trackers

Detailed evaluations of various approaches to DSTC2 tasks to our knowledge are reported in the table below.

| DST Model | Joint Goals | | Requested Slots | | Search Method | |
|---|---|---|---|---|---|---|
| | Acc. | L2 | Acc. | L2 | Acc. | L2 |
| Hybrid Tracker (Vodolan et al., 2017) † | **0.796** | **0.338** | – | – | – | – |
| Web-style Ranking (Williams, 2014) | 0.784 | 0.735 | 0.957 | 0.068 | **0.947** | **0.087** |
| Word-based Tracker (Henderson et al., 2014b) † | 0.768 | 0.346 | 0.978 | 0.035 | 0.940 | 0.095 |
| CMBP Tracker (Yu et al., 2015) | 0.762 | 0.436 | – | – | – | – |
| YARBUS Tracker (Fix and Frezza-Buet, 2015) | 0.759 | 0.358 | – | – | – | – |
| SJTU System (Sun et al., 2014) | 0.750 | 0.416 | 0.970 | 0.056 | 0.936 | 0.105 |
| TL-DST (Lee and Stent, 2016) | 0.747 | 0.451 | – | – | – | – |
| Knowledge-based Tracker (Kadlec et al., 2014) | 0.737 | 0.429 | – | – | – | – |
| Neural Belief Tracker (Mrksic et al., 2017) | 0.734 | – | 0.965 | – | – | – |
| EncDec Framework (Platek et al., 2016) †✓ | 0.730 | – | – | – | – | – |
| MTL Model *b* (this work) †✓ | 0.728 | 0.458 | **0.980** | **0.035** | 0.946 | 0.093 |
| Generative Model (Lee et al., 2014) | 0.726 | – | – | – | – | – |
| MTL Model *a* (this work) †✓ | 0.720 | 0.498 | 0.978 | 0.037 | 0.944 | 0.096 |
| LecTrack (Zilka and Jurcicek, 2015) †✓ | 0.72 | 0.64 | 0.97 | 0.06 | 0.93 | 0.14 |
| Markovian Model (Ren et al., 2014) | 0.718 | 0.461 | 0.951 | 0.085 | 0.871 | 0.210 |
| CNET Tracker (Jagfeld and Vu, 2017) †✓ | 0.714 | – | 0.972 | – | – | – |
| IJM Tracker (Trinh et al., 2017) †✓ | 0.707 | 0.545 | 0.975 | 0.047 | 0.940 | 0.114 |
| CRF Tracker (Kim and Banchs, 2014) | 0.601 | 0.649 | 0.960 | 0.073 | 0.904 | 0.155 |
| Best results | 0.796 | 0.338 | 0.980 | 0.035 | 0.947 | 0.087 |

Table 6: Performance evaluation of our proposed models and state-of-the-art incremental trackers. *Acc.* denotes Accuracy, and *L2* denotes the squared norm $l^2$. † means RNN-based Tracker, and ✓ means Incremental Tracker.

## Appendix B. Incremental DST output examples

We demonstrate Incremental Prediction examples of our model *b* on the dialogues in the testset.

In dialogue *voip-e8997b10da-20130401_151321* during turn 4 we observe the ASR error that leads to a wrong prediction output.

In dialogue *voip-a617b6827c-20130323_170453* our tracker performs well on a good ASR hypothesis.

| | | | | | | |
|---|---|---|---|---|---|---|
| Dialogue ID | *voip-e8997b10da-20130401_151321* | | | | | |
| Transcription | Turn 4 | *"okay how about indian food"* | | | | |
| | Turn 5 | *"okay how about indian food"* | | | | |
| Turn | ASR | Predicted States | | | Dialogue States | |
| | | Slot | Value | Probability | Slot | Value |
| 3 | *"<eos>"* | food | mediterranean | 0.990 | food | mediterranean |
| | | area | south | 0.993 | area | south |
| | | method | by constraints | 0.996 | method | by constraints |
| 4 | *"okay"* | food | mediterranean | 0.991 | food | indian |
| | | area | south | 0.998 | area | south |
| | | method | by constraints | 0.984 | method | by alternatives |
| | *"how"* | food | mediterranean | 0.989 | | |
| | | area | south | 0.998 | | |
| | | method | by constraints | 0.833 | | |
| | *"much"* | food | mediterranean | 0.990 | | |
| | | area | south | 0.999 | | |
| | | method | by constraints | 0.571 | | |
| | | | by alternatives | 0.414 | | |
| | *"union"* | food | mediterranean | 0.991 | | |
| | | area | south | 0.998 | | |
| | | method | by alternatives | 0.700 | | |
| | *"please"* | food | mediterranean | 0.990 | | |
| | | area | south | 0.998 | | |
| | | method | by alternatives | 0.815 | | |
| 5 | *"okay"* | food | mediterranean | 0.990 | food | indian |
| | | area | south | 0.998 | area | south |
| | | method | by alternatives | 0.606 | method | by alternatives |
| | *"how"* | food | mediterranean | 0.987 | | |
| | | area | south | 0.997 | | |
| | | method | by alternatives | 0.637 | | |
| | *"about"* | food | mediterranean | 0.975 | | |
| | | area | south | 0.994 | | |
| | | method | by alternatives | 0.614 | | |
| | *"indian"* | food | mediterranean | 0.111 | | |
| | | | indian | 0.480 | | |
| | | area | south | 0.994 | | |
| | | method | by alternatives | 0.880 | | |
| | *"food"* | food | indian | 0.977 | | |
| | | area | south | 0.995 | | |
| | | method | by alternatives | 0.961 | | |

Table 7: Incremental predictions for Dialogue *voip-e8997b10da-20130401_151321* in the testset. We use green/red colours to show right/wrong predictions of our tracker in comparison with labeled Dialogue States.

| Dialogue ID | *voip-a617b6827c-20130323_170453* | | | | | |
|---|---|---|---|---|---|---|
| Transcription | Turn 0 | *"im looking for an expensive restaurant in the south part of town"* | | | | |
| Turn | ASR | Predicted States | | | Dialogue States | |
| | | Slot | Value | Probability | Slot | Value |
| 0 | *"i'm"* | price | – | – | price | expensive |
| | | area | – | – | area | south |
| | | method | none | 0.980 | method | by constraints |
| | *"looking"* | price | – | – | | |
| | | area | – | – | | |
| | | method | none | 0.985 | | |
| | *"for"* | price | – | – | | |
| | | area | – | – | | |
| | | method | none | 0.963 | | |
| | *"an"* | price | – | – | | |
| | | area | – | – | | |
| | | method | none | 0.936 | | |
| | *"expensive"* | price | expensive | 0.856 | | |
| | | area | – | – | | |
| | | method | by constraints | 0.942 | | |
| | *"restaurant"* | price | expensive | 0.997 | | |
| | | area | – | – | | |
| | | method | by constraints | 0.998 | | |
| | *"in"* | price | expensive | 0.999 | | |
| | | area | – | – | | |
| | | method | by constraints | 0.999 | | |
| | *"the"* | price | expensive | 0.999 | | |
| | | area | – | – | | |
| | | method | by constraints | 0.999 | | |
| | *"south"* | price | expensive | 0.999 | | |
| | | area | south | 0.846 | | |
| | | method | by constraints | 0.999 | | |
| | *"part"* | price | expensive | 0.999 | | |
| | | area | south | 0.980 | | |
| | | method | by constraints | 0.999 | | |

Table 8: Incremental predictions for Dialogue *voip-a617b6827c-20130323_170453* in the testset. The ASR hypothesis misses two words *"of town"* from the user utterance.