SEMDIAL 2017 (SaarDial) Workshop on the Semantics and Pragmatics of Dialogue
15-17 August 2017, Saarbrücken, Germany

Challenging Neural Dialogue Models with Natural Data: Memory Networks Fail on Incremental Phenomena

Igor Shalyminov
Interaction Lab
Heriot-Watt University
is33@hw.ac.uk

Arash Eshghi
Interaction Lab
Heriot-Watt University
a.eshghi@hw.ac.uk

Oliver Lemon
Interaction Lab
Heriot-Watt University
o.lemon@hw.ac.uk

Abstract

Natural, spontaneous dialogue proceeds incrementally on a word-by-word basis; and it contains many sorts of disfluency such as mid-utterance/sentence hesitations, interruptions, and self-corrections. But training data for machine learning approaches to dialogue processing is often either cleaned-up or wholly synthetic in order to avoid such phenomena. The question then arises of how well systems trained on such clean data generalise to real spontaneous dialogue, or indeed whether they are trainable at all on naturally occurring dialogue data. To answer this question, we created a new corpus called bAbI+[1] by systematically adding natural spontaneous incremental dialogue phenomena such as *restart*s and *self-correction*s to the Facebook AI Research's bAbI dialogues dataset. We then explore the performance of a state-of-the-art retrieval model, MemN2N (Bordes et al., 2017; Sukhbaatar et al., 2015), on this more natural dataset. Results show that the semantic accuracy of the MemN2N model drops drastically; and that although it is in principle able to learn to process the constructions in bAbI+, it needs an impractical amount of training data to do so. Finally, we go on to show that an incremental, semantic parser – DyLan – shows 100% semantic accuracy on both bAbI and bAbI+, highlighting the generalisation properties of linguistically informed dialogue models.

1 Introduction

A key problem for the practical data-driven (rather than hand-crafted) development of task-oriented dialogue systems is that they are generally turn-based, and so do not support natural, everyday *incremental* (i.e. word-by-word) dialogue processing. This means that they often cannot process naturally occurring incremental dialogue phenomena such as mid-sentence restarts and self-corrections (Hough, 2015; Howes et al., 2009). Dialogue systems will not be able to make sense of the everyday language produced by users which is replete with pauses, interruptions, self-corrections and other inherently incremental dialogue phenomena, until they incorporate one or another form of incremental language processing. Indeed incremental dialogue systems (i.e. processing word-by-word instead of at utterance/turn boundaries) have previously been empirically shown to be beneficial and more natural for users (Aist et al., 2007; Skantze and Hjalmarsson, 2010).

In this paper, we explore the performance of the state-of-the-art neural retrieval model of Bordes et al. (2017) on dialogues containing some prototypical incremental dialogue structures. Bordes et al. (2017) initially presented the bAbI dialog tasks dataset aimed at learning goal-oriented dialogue systems in an end-to-end fashion: there are no annotations in the data whatsoever, and the model learns all components of a dialogue system. On this dataset, they report that End-to-End Memory Networks (henceforth MemN2Ns) achieve an impressive 100% performance on a test set of 1000 dialogues, after being trained on 1000 similar dialogues.

However, the bAbI dataset is both synthetic and clean: it contains none of the more interesting naturally occurring, disfluent phenomena identified above. To assess the effectiveness of the

[1] this dataset is freely available at https://bit.ly/babi_plus

10.21437/SemDial.2017-11

MemN2N model on more natural dialogue data, we introduce an extended, incremental version of the bAbI dataset – dubbed bAbI+ (see section 2.2) – which we created by systematically adding self-corrections, hesitations, and restarts to the original bAbI dataset.

We go on to explore the performance of MemN2N on this new dataset. The results of our experiments show that the semantic accuracy of MemN2N , measured in terms of how well the model predicts API calls (a non-linguistic action – in this case querying a data-base with the user's requirements) at the end of a dialogue segment, drops very significantly (by about 50%) even when trained on the full bAbI+ dataset.

Finally, we compare these results to the methodologically distinct, linguistically informed model of (Eshghi et al., 2017b; Kalatzis et al., 2016), who employ an incremental dialogue parser, `DyLan` (Eshghi, 2015; Eshghi et al., 2011; Purver et al., 2011); based around the Dynamic Syntax grammar framework (Kempson et al., 2001; Cann et al., 2005)). We show here that there is no drop in performance in the same semantic accuracy metric from bAbI to bAbI+ with both at 100% due to the rich, theoretically-grounded knowledge incorporated within the model.

2 Exploring the performance of MemN2Ns

Our focus in this paper is to explore the approach of Bordes et al. (2017), and its performance on spontaneous dialogue data.

2.1 The Dialog bAbI tasks dataset

We use Facebook AI Research's Dialogue bAbI tasks dataset (Bordes et al., 2017). These are goal-oriented dialogues in the domain of restaurant search. In the dataset, there are 6 tasks of increasing complexity ranging from only collecting the user's preferences on restaurant and up to conducting full dialogues with changes in the user's goal and providing extra information upon request. The first 5 tasks are 'clean' dialogues composed synthetically and they thus lack the features of natural everyday conversations. Task 6, in turn, is based on real dialogues collected for the Dialog State Tracking Challenge 2.

Recent studies have shown different ways in which MemN2Ns are outperformed: Eric and Manning (2017) introduced the Copy-Augmented Sequence-to-Sequence model that outperforms MemN2N on Task 6; Williams et al. (2017) presented a hybrid RNN + rule-based model trainable in a 2-stage supervised + reinforcement learning setup, outperforming MemN2N on Tasks 5 and 6.

However, none of these studies control for *the type of complexity* that might result in worse performance, and thus do not shed any light on why a particular architecture such as MemN2N might be at a disadvantage. While Task 5 dialogues have the full task complexity, conducting full dialogues with an unfixed user goal and additional information requests, they are still composed programmatically and contain minimal surface variation. The Task 6 dialogues on the other hand are complex both in terms of the surface variation and the task itself.

Here, in order to study the specific effects of incremental variations in dialogue such as conversational disfluencies, we focus on Task 1, where in each dialogue the system asks the user about their preferences for the properties of a restaurant, and each dialogue results in an *API call* containing values of each slot obtained (e.g. `food-type=french`) – the ability of predicting the API calls correctly thus provides a direct measure of how a well a particular model can interpret the dialogues.

Using the MemN2N model, the approach of Bordes et al. (2017) achieves 100% performance – measured as per-utterance accuracy including the final API call – after training on 1000 dialogues.

2.2 The bAbI+ dataset

While containing sufficient lexical variation, the original bAbI Task 1 dialogues significantly lack incremental and interactional variations vital for natural real-life dialogues. In order to obtain such variation while keeping the controllable environment close to the laboratory conditions that bAbI offers, we created the bAbI+ dataset by systematically transforming the original dataset's dialogues.

bAbI+ is an extension of the bAbI Task 1 dialogues with everyday incremental dialogue phenomena (hesitations, restarts, and corrections – see below). This extension can be seen as orthogonal to the increasing task complexity which Tasks 2–6 offer: we instead increase the complexity of surface forms of dialogue utterances, while keeping every other aspect of the task fixed.

Our incremental modifications model the disflu-

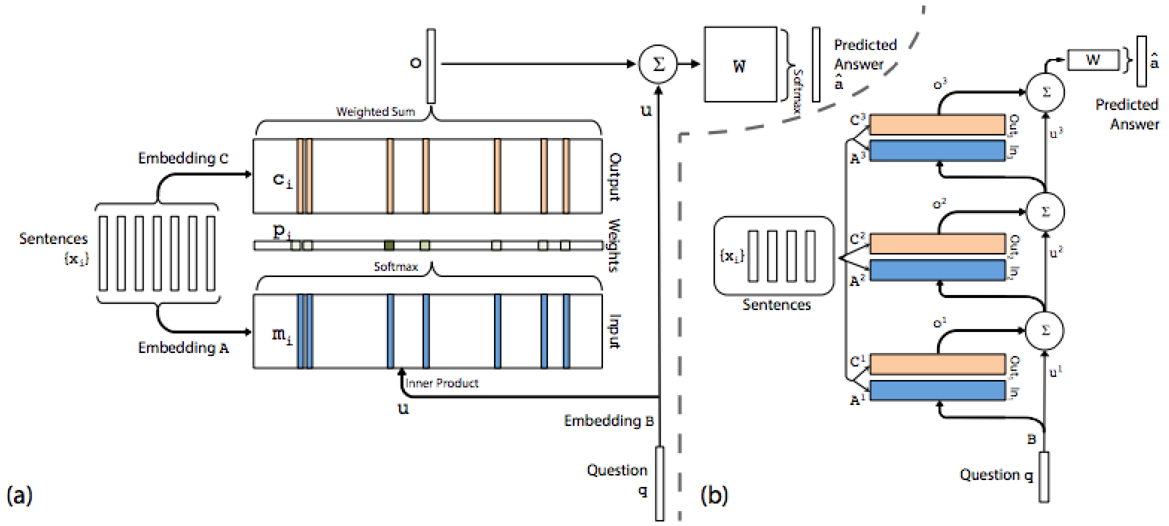

Figure 1: MemN2N architecture – (a) single memory hop and (b) multiple memory hops, from (Sukhbaatar et al., 2015)

encies and communication problems in everyday spoken interaction in real-world environments. These variations are:

- **1. Hesitations**, e.g. as in "we will be uhm eight";
- **2. Restarts**, e.g. "can you make a restaurant uhm yeah can you make a restaurant reservation for four people with french cuisine in a moderate price range";
- **3. Corrections** affecting task-specific information – both short-distance ones correcting one token, e.g. "with french no spanish food", and long-distance NP/PP-level corrections, e.g. "with french food uhm sorry with spanish food", all within a single user utterance, rather than across multiple turns.

The phenomena above are mixed in probabilistically from the fixed sets of templates to the original data[2]. The modifications affect a total of **11336** utterances in the **3998** dialogues. Around **21%** of user's turns contain corrections, **40%** hesitations, and **5%** restarts (they are not mutually exclusive, so that an utterance can contain up to 3 modifications). Our modifications, with respect to corrections in particular, are more conservative than those observed in real-world data: (Hough, 2015) reports that self-corrections appear in **20%** of all turns natural conversations from British National Corpus, and in **40%** of turns in the Map Task, a corpus of human-human goal-oriented dialogues.

Here's part of an example dialogue in the bAbI+ corpus, showing corrections in all user turns:

sys: hello what can I help you with today?
usr: I'd like to book a uhm yeah I'd like to book a table in a expensive price range no sorry in a cheap price range
sys: I'm on it. Any preference on a type of cuisine?
usr: with indian food no sorry with spanish food
sys: where should it be?
usr: in tokyo
sys: how many people would be in your party?
usr: for uhm yeah for eight people please
sys: ok let me look into some options for you

2.3 Memory Networks

We follow Bordes et al.'s setup by using a MemN2N (we took an open source Tensorflow implementation for bAbI QA tasks and modified it[3] according to their setup – see details below).

The architecture of a MemN2N is shown in Figure 1 (Sukhbaatar et al., 2015).

One of the model's key features is what can be thought of as an "explicit memory" component: before the inference process, all the input sentences are written into the memory from which they are then read during the inference. The internal memory representation is as follows: an utterance x_i is first vectorized as a fixed-sized array of 1-hot vectors (zero padding is used), then

[2] See https://github.com/ishalyminov/babi_tools

[3] See https://github.com/ishalyminov/memn2n

each vector is embedded via the matrix A, and finally these embeddings are encoded into a single memory vector m_i using temporal encoding (it preserves the information of word order in a sentence – for the details, please refer to (Sukhbaatar et al., 2015)). The same procedure is applied to the user's input using another embedding matrix B.

Another important feature in the MemN2N architecture is reading from memory with attention. With the input sentences and the utterance encoded, the match between each of the memory vectors m_i and the utterance u is calculated:

$$p_i = Softmax(u^T m_i)$$

This is used as the attention vector over the encoded memories further in the inference process.

Next, for the final answer prediction, both attention-weighted memories and user's utterance are passed through the final weight matrix W:

$$\hat{a} = Softmax(W(o + u))$$

where $o = \sum_i p_i c_i$ is weighted memories.

For the QA tasks, the answer $\hat{a}$ is just an index of a word from the vocabulary. In dialogue tasks, however, answers are the entire utterances, either system utterances (e.g. *"how many people would be in your party?"*) or API calls ("`api_call french london four expensive`"). They are still predicted as indices from the answer candidates list, but given that there is e.g. absolutely no overlap in exact api call examples between train and test sets, an internal representation of each candidate answer is added to the architecture (Bordes et al., 2017). Thus, the final step now looks as follows:

$$\hat{a} = Softmax((o + u)^T \cdot W(y))$$

where y is a vector of answer candidates processed just as described above for the input sentences, with W as the embedding matrix.

The architecture described above may be stacked into several layers called hops (Figure 1 (b)) – refer for details to (Sukhbaatar et al., 2015); here we're initially interested in the single hop configuration (see the next section), for which (Bordes et al., 2017) report their results.

2.4 Data preprocessing and the MemN2N setup

In order to adapt the data for the MemN2N, we transform the dialogues into <*story, question, answer*> triplets. The number of triplets for a single dialogue is equal to the number of the system's turns, and in each triplet, the `answer` is the current system's turn, the `question` is the user's turn preceding it, and the `story` is a list of all the previous turns from both sides. Other than that, each sentence in the `story` gets 2 additional tokens: the number of the turn, and the ID of the speaker (Bordes et al., 2017).

We also use the single embedding matrix A for both input memories and the user's question; the same matrix is used for the output memories representation – in that we follow (Bordes et al., 2017), and it corresponds to the "Adjacent" weight tying model in (Sukhbaatar et al., 2015).

In our setup, there are no out-of-vocabulary words for the model during both training and testing, and for both bAbI and bAbI+ with the maximum sentence length taking account of the increase due to the transformations in bAbI+.

We train our MemN2Ns with a Stochastic Gradient Descent optimizer for **100** epochs with a learning rate of **0.01** and a batch size of **8** – in this we again follow the configuration reported by (Bordes et al., 2017) to be the best for bAbI Task 1.

2.5 Experiments

We are here interested in: (1) how robust MemN2Ns are to the surface transformations in bAbI+ when trained on bAbI; (2) can MemN2Ns learn to interpret bAbI+ when they are in fact trained on similar data that actually contain the bAbI+ structures – i.e. when trained on bAbI+; and (3) if so, how much bAbI+ data is needed for this. While (1) is a question about generalisation properties of a model, (2) & (3) are about potential in principle and/or practical limitations of MemN2Ns to learn to interpret dialogues containing, e.g. self-corrections where utterances contain both the correct, and an incorrect (and subsequently repaired) slot value (e.g. "for four sorry five people"). To answer (1) we therefore train the model on the bAbI dataset and test on bAbI+; and to answer (2) & (3) we train the model on the bAbI+ train set and test it on the bAbI+ test set. Furthermore, in order to explore the impact of the amount of training data on the model's performance, we perform the latter experiment with varying train set size, as well as varying the hyperparameters, embedding size & number of hops.

train / test set configuration	train accuracy	test accuracy
bAbI / bAbI	100	100
bAbI / bAbI+	100	28
bAbI+ / bAbI	67	99
bAbI+ / bAbI+	72	53

Table 1: MemN2N API call accuracy (%)

training bAbI+ dialogues	memory hops	embedding size	train accuracy	test accuracy
2000	2	128	72.5	57.5
5000	2	128	72.7	60.7
10000	2	128	72.8	65.8
50000	1	128	82.6	78.2
100000	1	64	83.3	80.5

Table 2: MemN2N API call accuracy (%) with extended training data

The extended training data is obtained in the same way as the initial bAbI+ dataset: we go over the same original bAbI dialogues and keep randomly mixing in the incremental modifications.

Performance Measure: Semantic Accuracy Self-corrections and restarts are especially problematic because processing them is potentially a non-monotonic operation involving deletion and replacement in the resulting semantic representations. To measure the model's effectiveness in processing such structures we therefore consider the *semantic accuracy* of the model defined as how accurately it predicts the final API calls – recall that the API calls contain all the values of the slots corresponding to the user's request expressed in the preceding dialogue.

Hypotheses We predicted that (i) given the positional encoding of memory vectors in the MemN2N model and the attendant attention mechanisms, it would be able to learn to process bAbI+ dialogues given that it was trained on similar data, resulting in an insignificant drop in performance from bAbI to bAbI+ data; (ii) a lot more data would be needed to learn to process the bAbI+ structures; and (iii) if trained on bAbI data, there would be a very significant drop in performance on bAbI+ with incorrect API calls predicted as a result of incorrect weightings and total lack of opportunity to learn the meaning of words such as "no" or "sorry" which trigger the self-corrections and restarts.

Finally, we also perform training on bAbI+ and testing on bAbI to see if the model is able to generalise from more complex back to the simpler data.

2.6 Results and Discussion

2.6.1 The original setup

Table 1 shows how the MemN2N model performs in different conditions. For this, we used identical hyperparameter settings to those of Bordes et al. (2017): **1** hop, **128** embedding size, **100** epochs, learning rate of **0.01**, and batch size of **8**. The train and test sets each contain 1000 dialogues, i.e. the entire corpus.

First note that the first row shows identical results to those of Bordes et al. (2017): training on bAbI and testing on the bAbI test set results in 100% accuracy in API call prediction. It is therefore highly unlikely that the rest of the results reported here are due to implementational differences between our setup and that of Bordes et al. (2017).

As we had predicted, the model performs very badly when trained on bAbI and tested on bAbI+ showing very poor robustness to the variations we had introduced, and indicating significant overfitting to the original data.

When the model is trained on bAbI+ data, its performance on the bAbI+ API calls nearly doubles, showing that the model can potentially learn to process the bAbI+ test set given enough data – see below. Nevertheless, it remains very low at about 53% making any system created in this fashion unusable in the face of spontaneous dialogue data. We also note that the accuracy on the train set itself is now lower. This suggests that bAbI+

is a dataset significantly harder to learn (or overfit to), and given the extreme homogeneity of the original bAbI train and test sets, overfitting might be one reason for the model's outstanding results. However, training on bAbI+ and testing on bAbI shows that our assumption about the model's ability to generalize to more simple data appears to be correct.

2.6.2 How much data is enough data?

Table 2 shows how MemN2N performs on the same initial, fixed bAbI+ test set, when trained on progressively more data and up to 100000 bAbI+ dialogues. As MemN2N 's performance on bigger data highly depends on the model's hyperparameters, in this experiment we perform a grid search over the number of memory hops (1, 2, 3), and the embeddings dimensionality (32, 64, 128) for each train set size independently – everything else is fixed as in the previous experiment. The table only shows the best performing hyperparameter configuration for each of the train set sizes.

The results confirm hypothesis (ii) above, i.e. that MemN2Ns are in principle able to learn to process the incremental dialogue phenomena in bAbI+ but that they require tens of thousands of training instances for this: even with 100000 dialogues, the semantic accuracy on the original test set stands at 80.5%.

These experiments shed significant light on the currently ambiguous robustness results reported in the dialogue systems literature today. Specifically, they show that, from the point of view of dialogue system developers in the real world, learning to process natural spontaneous dialogue using MemN2N*s only* in an end-to-end fashion may not be practical: in bAbI+, the disfluent incremental phenomena were mixed in at will, thus affording access to arbitrarily large training sets; furthermore, the test set was synthetically constructed to follow the same pattern as in the train set; whereas real, natural, spontaneous dialogue data is not only very expensive to collect, but is bound to be more complex, with the closeness between train & test data very difficult to control.

A potential solution to this 'small data' problem is the use of computational dialogue models (such as e.g. (Ginzburg, 2012; Larsson, 2002; Poesio and Rieser, 2010; Eshghi et al., 2015)) with studied empirical foundation as a form of bias or prior in subsequent learning, thus exploiting the linguistic knowledge inherent in such models. Even if they are not used directly, they can be used to inform the architecture of particular machine learning methods, especially deep learning architectures and techniques, with a view to more modularity in such architectures, with general language processing modules that are transferable from one domain to another, much like a NL grammar.

3 Testing an incremental, semantic grammar on bAbI & bAbI+

In this section, we first quickly introduce an incremental, semantic parser for dialogue processing – DyLan (Eshghi et al., 2011; Eshghi, 2015; Purver et al., 2011) – based around the Dynamic Syntax and Type Theory with Records framework (Kempson et al., 2001; Cann et al., 2005; Eshghi et al., 2012; Cooper, 2005; Cooper, 2012), which has been used recently in combination with Reinforcement Learning for automatically inducing fully incremental dialogue systems from small amounts of raw, unannotated dialogue data (Eshghi and Lemon, 2014; Kalatzis et al., 2016), showing remarkable generalisation properties (Eshghi et al., 2017b; Eshghi et al., 2017a). We then go on to perform the same experiments on semantic accuracy as we did above with MemN2Ns using this linguistically informed model instead.

3.1 DyLan[4]: parser for Dynamic Syntax

DyLan (Eshghi et al., 2011; Eshghi, 2015) is the parser/implementation for Dynamic Syntax (DS), an action-based, word-by-word incremental, semantic grammar formalism (Kempson et al., 2001; Cann et al., 2005), especially suited to the highly fragmentary and contextual nature of dialogue. In DS, words are conditional actions – semantic updates; and dialogue is modelled as the interactive and incremental construction of contextual and semantic representations (Eshghi et al., 2015) – see Fig. 2 which shows how semantic representations are constructed incrementally as Record Types of Type Theory with Records (TTR) (Cooper, 2005; Cooper, 2012). The contextual representations afforded by DS are of the fine-grained semantic content that is jointly negotiated/agreed upon by the interlocutors, as a result of processing questions and answers, clarification interaction, acceptances, self-/other-corrections, restarts, and other characteristic incremental phenomena in dialogue – see Fig. 3 for a sketch of how self-

[4] DyLan is derived from "Dynamics of Language"

$$\begin{bmatrix} event & : e_s \\ p1_{=today(event)} & : t \end{bmatrix} \mapsto \begin{bmatrix} event_{=arrive} & : e_s \\ p1_{=today(event)} & : t \\ p2_{=pres(event)} & : t \\ x_{=robin} & : e \\ p3_{=subj(event,x)} & : t \end{bmatrix} \mapsto \begin{bmatrix} event_{=arrive} & : e_s \\ p1_{=today(event)} & : t \\ p2_{=pres(event)} & : t \\ x_{=robin} & : e \\ p3_{=subj(event,x)} & : t \\ x1 & : e \\ p3_{=from(event,x1)} & : t \end{bmatrix} \mapsto \begin{bmatrix} event_{=arrive} & : e_s \\ p1_{=today(event)} & : t \\ p2_{=pres(event)} & : t \\ x_{=robin} & : e \\ p_{=subj(event,x)} & : t \\ x1_{=Sweden} & : e \\ p3_{=from(event,x1)} & : t \end{bmatrix}$$

"A: Today" ↦ *"..Robin arrives"* ↦ *"B: from?"* ↦ *"A: Sweden"*

Figure 2: Incremental parsing with DyLan

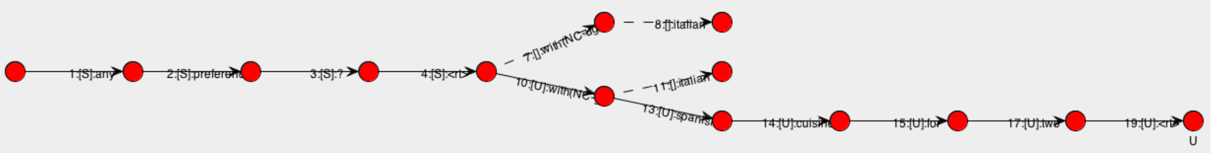

Figure 3: Processing self-corrections & restarts with DyLan: "A: any preference? B: with italian yeah sorry with spanish cuisine"

corrections and restarts are processed via a backtrack and search mechanism over the parse search graph. The nodes in this graph are (partial) semantic trees, and the edges correspond to words uttered by particular speakers. Context of a partial tree in DS is the path back to root on this parse search graph (see Hough (2015; Hough and Purver (2014; Eshghi et al. (2015) for details of the model). The upshot of this is that using DS, one can not only track the semantic content of some current turn as it is being constructed (parsed or generated) word-by-word, but also the context of the conversation as whole, with the latter also encoding the grounded/agreed content of the conversation (see Eshghi et al. (2015); Purver et al. (2010) for details). Crucially for (Eshghi et al., 2017b)'s model, the inherent incrementality of DS together with the word-level, as well as cross-turn, parsing constraints it provides, enables the word-by-word exploration of the space of grammatical dialogues, thus lending itself very well to Reinforcement Learning (Kalatzis et al., 2016; Eshghi et al., 2017a).

3.2 Parsing bAbI and bAbI+ dialogues with DS

The Dynamic Syntax (DS) grammar is learnable from data (Eshghi et al., 2013a; Eshghi et al., 2013b). But since the lexicon was induced from a corpus of child-directed utterances in this prior work, there were some constructions as well as individual words that it did not include[5]. One of the authors therefore extended this induced grammar manually to cover the bAbI dataset, which, despite not being very diverse, contains a wide range of complex grammatical constructions, such as long sequences of prepositional phrases, adjuncts, short answers to yes/no and wh-questions, appositions of NPs, causative verbs etc – and all of this within and across dialogue turns/speakers.

Using DyLan we parsed all dialogues in the bAbI train and test sets, as well as on the bAbI+ corpus word-by-word, including both user and system utterances, in context. The grammar parses 100% of the dialogues, i.e. it does not fail on any word in any of the dialogues.

3.3 Semantic Accuracy of DyLan

Merely parsing all dialogues in the bAbI and bAbI+ datasets doesn't mean that the semantic representations compiled for the dialogues were in fact correct. To measure the semantic accuracy of the parser, we used, as before, the API call annotations at the end of bAbI and bAbI+ task 1 dialogues. This was done programmatically by checking that the correct slot values – those in the API call annotations – were in fact present in the semantic representations produced by the parser for each dialogue (see Fig. 2 for example semantic representations). We further checked that there is no other incorrect slot value present in these representations.

The results showed that the parser has 100% se-

[5] in the near future we will use the learning method in Eshghi et al. (2013a) to induce DS grammars from larger semantic corpora such as the Groningen Meaning Bank, leading to much more wide-coverage lexicons than the present one

mantic accuracy on both bAbI and bAbI+. This result is not surprising, given that Dynamic Syntax is a general model of incremental language processing, including phenomena such as self-corrections & restarts (see (Hough, 2015) for details of the model)[6]. It is worth noting that even though new lexical entries would have to be added for each new dataset/domain, given the parts-of-speech of the words in any given dataset, this can mostly be done automatically.

Moreover, this result further reinforces the point made by Eshghi et al. (2017a) about the generalisation power of the Dynamic Syntax grammar: the grammar automatically generalises to a combinatorially large number of dialogue variations with various phenomena such as self-corrections, hesitations, restarts, clarification interaction, continuations, question-answer pairs etc. without having actually observed these in any of the seed/training dialogues.

4 Conclusion and ongoing work

Our main advance is in exploring incremental processing for wider coverage of more natural everyday dialogue (e.g. containing self-corrections).

Our experiments show that a state-of-the-art model for end-to-end goal-oriented dialogue, MemN2N , lacks the ability to generalise to such phenomena, and performs poorly when confronted with natural spontaneous dialogue data. Our experiments further show that although this particular model is in principle able to learn to process incremental dialogue phenomena, it requires an impractically large amount of data to do so. The results in this paper therefore shed significant light on the currently ambiguous robustness results reported for end-to-end systems.

We also assessed the performance of the DyLan dialogue parser on bAbI and bAbI+ which showed 100% parsing and semantic accuracy, highlighting the generalisation power of models that are linguistically informed, and theoretically grounded as compared with pure machine learning methods that aim to learn to process dialogue bottom up from textual data alone, without any linguistic bias. These issues are explored further in (Eshghi et al., 2017a).

[6] A helpful reviewer points out that the DyLan setup is a carefully tuned rule-based system, thus rendering these results trivial. But we note that the results here are not due to ad-hoc constructions of rules/lexicons, but due to the generality of the grammar model, and its attendant incremental, left-to-right properties. For example, the ability to process self-corrections, restarts, etc. "comes for free", without the need to add or posit new machinery

References

Gregory Aist, James Allen, Ellen Campana, Carlos Gomez Gallo, Scott Stoness, Mary Swift, and Michael K Tanenhaus. 2007. Incremental dialogue system faster than and preferred to its nonincremental counterpart. In *Annual Conference of the Congitive Science Society*.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. *ICLR*.

Ronnie Cann, Ruth Kempson, and Lutz Marten. 2005. *The Dynamics of Language*. Elsevier, Oxford.

Robin Cooper. 2005. Records and record types in semantic theory. *Journal of Logic and Computation*, 15(2):99–112.

Robin Cooper. 2012. Type theory and semantics in flux. In Ruth Kempson, Nicholas Asher, and Tim Fernando, editors, *Handbook of the Philosophy of Science*, volume 14: Philosophy of Linguistics, pages 271–323. North Holland.

Mihail Eric and Christopher D. Manning. 2017. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. *CoRR*, abs/1701.04024.

Arash Eshghi and Oliver Lemon. 2014. How domain-general can we be? Learning incremental dialogue systems without dialogue acts. In *Proceedings of Semdial 2014 (DialWatt)*.

A. Eshghi, M. Purver, and Julian Hough. 2011. Dylan: Parser for dynamic syntax. Technical report, Queen Mary University of London.

Arash Eshghi, Julian Hough, Matthew Purver, Ruth Kempson, and Eleni Gregoromichelaki. 2012. Conversational interactions: Capturing dialogue dynamics. In S. Larsson and L. Borin, editors, *From Quantification to Conversation: Festschrift for Robin Cooper on the occasion of his 65th birthday*, volume 19 of *Tributes*, pages 325–349. College Publications, London.

Arash Eshghi, Julian Hough, and Matthew Purver. 2013a. Incremental grammar induction from child-directed dialogue utterances. In *Proceedings of the 4th Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, pages 94–103, Sofia, Bulgaria, August. Association for Computational Linguistics.

Arash Eshghi, Matthew Purver, Julian Hough, and Yo Sato. 2013b. Probabilistic grammar induction in an incremental semantic framework. In *CSLP, Lecture Notes in Computer Science*. Springer.

A. Eshghi, C. Howes, E. Gregoromichelaki, J. Hough, and M. Purver. 2015. Feedback in conversation as incremental semantic update. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS 2015)*, London, UK. Association for Computational Linguisitics.

Arash Eshghi, Igor Shalyminov, and Oliver Lemon. 2017a. Bootstrapping incremental dialogue systems from minimal data: linguistic knowledge or machine learning? In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Arash Eshghi, Igor Shalyminov, and Oliver Lemon. 2017b. Interactional Dynamics and the Emergence of Language Games. In *Proceedings of the ESSLLI 2017 workshop on Formal approaches to the Dynamics of Linguistic Interaction*, Barcelona.

Arash Eshghi. 2015. DS-TTR: An incremental, semantic, contextual parser for dialogue. In *Proceedings of Semdial 2015 (goDial), the 19th workshop on the semantics and pragmatics of dialogue*.

Jonathan Ginzburg. 2012. *The Interactive Stance: Meaning for Conversation*. Oxford University Press.

Julian Hough and Matthew Purver. 2014. Probabilistic type theory for incremental dialogue processing. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 80–88, Gothenburg, Sweden, April. Association for Computational Linguistics.

Julian Hough. 2015. *Modelling Incremental Self-Repair Processing in Dialogue*. Ph.D. thesis, Queen Mary University of London.

C. Howes, P. G. T. Healey, and G.J. Mills. 2009. A: An experimental investigation into... B:... Split utterances. In *Proceedings of the 10th Annual SIGDIAL Meeting on Discourse and Dialogue (SIGDIAL 2009 Conference)*, pages 79–86. Association for Computational Linguistics.

Dimitrios Kalatzis, Arash Eshghi, and Oliver Lemon. 2016. Bootstrapping incremental dialogue systems: using linguistic knowledge to learn from minimal data. In *Proceedings of the NIPS 2016 workshop on Learning Methods for Dialogue*, Barcelona.

Ruth Kempson, Wilfried Meyer-Viol, and Dov Gabbay. 2001. *Dynamic Syntax: The Flow of Language Understanding*. Wiley-Blackwell.

Staffan Larsson. 2002. *Issue-based Dialogue Management*. Ph.D. thesis, Göteborg University. Also published as Gothenburg Monographs in Linguistics 21.

Massimo Poesio and Hannes Rieser. 2010. Completions, coordination, and alignment in dialogue. *Dialogue and Discourse*, 1:1–89.

Matthew Purver, Eleni Gregoromichelaki, Wilfried Meyer-Viol, and Ronnie Cann. 2010. Splitting the 'I's and crossing the 'You's: Context, speech acts and grammar. In P. Łupkowski and M. Purver, editors, *Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue*, pages 43–50, Poznań, June. Polish Society for Cognitive Science.

Matthew Purver, Arash Eshghi, and Julian Hough. 2011. Incremental semantic construction in a dialogue system. In J. Bos and S. Pulman, editors, *Proceedings of the 9th International Conference on Computational Semantics*, pages 365–369, Oxford, UK, January.

Gabriel Skantze and Anna Hjalmarsson. 2010. Towards incremental speech generation in dialogue systems. In *Proceedings of the SIGDIAL 2010 Conference*, pages 1–8, Tokyo, Japan, September. Association for Computational Linguistics.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2440–2448.

Jason Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*. Association for Computational Linguistics, July.