Multi-Task Learning for Domain-General Spoken Disfluency Detection in Dialogue Systems

Igor Shalyminov, Arash Eshghi, and Oliver Lemon
The Interaction Lab, Department of Computer Science
Heriot-Watt University, Edinburgh, EH14 4AS, UK
{is33, a.eshghi, o.lemon}@hw.ac.uk

Abstract

Spontaneous spoken dialogue is often disfluent, containing pauses, hesitations, self-corrections and false starts. Processing such phenomena is essential in understanding a speaker's intended meaning and controlling the flow of the conversation. Furthermore, this processing needs to be *word-by-word incremental* to allow further downstream processing to begin as early as possible in order to handle real spontaneous human conversational behaviour. In addition, from a developer's point of view, it is highly desirable to be able to develop systems which can be trained from 'clean' examples while also able to generalise to the very diverse disfluent variations on the same data – thereby enhancing both data-efficiency and robustness. In this paper, we present a multi-task LSTM-based model for incremental detection of disfluency structure[1], which can be hooked up to any component for incremental interpretation (e.g. an incremental semantic parser), or else simply used to 'clean up' the current utterance as it is being produced. We train the system on the Switchboard Dialogue Acts (SWDA) corpus and present its accuracy on this dataset. Our model outperforms prior neural network-based incremental approaches by about 10 percentage points on SWDA while employing a simpler architecture. To test the model's generalisation potential, we evaluate the same model on the bAbI+ dataset, without any additional training. bAbI+ is a dataset of synthesised goal-oriented dialogues where we control the distribution of disfluencies and their types. This shows that our approach has good generalisation potential, and sheds more light on which types of disfluency might be amenable to domain-general processing.

1 Introduction

It is uncontested that humans process (parse and generate) language, *incrementally*, word by word, rather than turn by turn, or sentence by sentence (Howes et al., 2010; Crocker et al., 2000; Ferreira et al., 2004). This leads to many characteristic phenomena in spontaneous dialogue that are difficult to capture in traditional linguistic approaches and are still largely ignored by dialogue system developers. These include various kinds of context-dependent fragment (Fernández and Ginzburg, 2002; Fernández, 2006; Kempson et al., 2017), false starts, suggested add-ons, barge-ins and disfluencies.

In this paper, we focus on disfluencies: pauses, hesitations, false starts and self-corrections that are common in natural spoken dialogue. These proceed according to a well-established general structure with three phases (Shriberg, 1994):

(1) with $\underbrace{\text{[Italian}}_{reparandum} + \underbrace{\{uh\}}_{interregnum} \underbrace{\text{Spanish]}}_{repair}$ cuisine

Specific disfluency structures have been shown to serve different purposes for both the speaker & the hearer (see e.g Brennan and Schober (2001)), for example, a filled pause such as 'uhm' can elicit a completion from the interlocutor, but also serve as a turn-holding device; mid-sentence self-corrections are utilised to deal with the speaker's own error as early as possible, thus minimising effort.

In dialogue systems, the detection, processing & integration of disfluency structure is thus crucial to understanding the interlocutor's intended meaning (i.e. robust Natural Language Understanding), but

[1] Code and trained models available at https://bit.ly/multitask_disfluency

also for coordinating the flow of the interaction. Like dialogue processing in general, the detection & integration of disfluencies needs to be *strongly incremental*: it needs to proceed word by word, enabling downstream processing to begin as early as possible, leading to more efficient and more naturally interactive dialogue systems (Skantze and Hjalmarsson, 2010; Schlangen and Skantze, 2009).

Furthermore, incremental disfluency detection needs to proceed with minimal latency & commit to hypotheses as early as possible in order to avoid 'jittering' in the output and having to undo the downstream processes started based on erroneous hypotheses (Schlangen and Skantze, 2009; Hough and Purver, 2014; Hough and Schlangen, 2015) .

While many current data-driven dialogue systems tend to be trained end-to-end on natural data, they don't normally take the existence of disfluencies into account. Recent experiments have shown that end-to-end dialogue models such as Memory Networks (MemN2N) (Bordes et al., 2017) need impractically large amounts of training data containing disfluencies and with sufficient variation in order to obtain reasonable performance (Eshghi et al., 2017; Shalyminov et al., 2017). The problem is that, taken together with the particular syntactic and semantic contexts in which they occur, disfluencies are very sparsely distributed, which leads to a large mismatch between the training data and actual real-world spontaneous user input to a deployed system. This suggests a more modular, pipelined approach, where disfluencies are detected and processed by a separate, domain-general module, and only then any resulting representations are passed on for downstream processing. The upshot of such a modular approach would be a major advantage in generality, robustness, and data-efficiency.

In this paper, we build on the state-of-the-art neural models of Hough and Schlangen (2015) and Schlangen and Hough (2017). Our contributions are that: (1) we produce a new, multi-task LSTM-based model with a simpler architecture for incremental disfluency detection, with significantly improved performance on the SWDA, a disfluency-tagged corpus of open-domain conversations; and (2) we perform a generalisation experiment measuring how well the models perform on unseen data using the controlled environment of bAbI+ (Eshghi et al., 2017), a synthetic dataset of goal-oriented dialogues in a restaurant search domain augmented with spoken disfluencies.

2 Related work

Work on disfluency detection has a long history, going back to Charniak and Johnson (2001) who set the challenge. One of the important dividing lines through this work is the *incrementality* aspect, i.e. whether disfluency structure is predicted word by word.

In the non-incremental setting, as the problem is essentially sequence tagging, neural models have been widely used. As such, there are approaches using an encoder-decoder model (seq2seq) with attention (Wang et al., 2016) and a Stack-LSTM model working as a buffer of a transition-based parser (Wang et al., 2016; Wang et al., 2017), the latter being state-of-the-art for the non-incremental setting.

Incremental, online processing of disfluencies is a more challenging task, if only because there is much less information available for tagging, viz. only the context on the left. In a practical system, it also involves extra constraints and evaluation criteria such as minimal latency and revisions to past hypotheses which lead to 'jittering' in the output with all the dependent downstream processes having to be undone, thus impeding efficiency (see the illuminating discussions in Hough and Purver (2014) and Purver et al. (2018)).

Incremental disfluency detection models include Hough and Purver (2014) who approach the problem information-theoretically, using local surprisal/entropy measures and a pipeline of classifiers for recognition of the various components of disfluency structure. While the model is very effective, it leaves one desiring a simpler alternative. This was made possible after the overall success of RNN-based models, which Hough and Schlangen (2015) exploit. We build on this model here, as well as evaluate it further (see below). On the other hand, Schlangen and Hough (2017) tackle the task of joint disfluency prediction and utterance segmentation, and demonstrate that the two tasks interact and thus are better approached jointly.

Language models have been extensively used for improving neural models' performance. For example, Peters et al. (2018) showed that a pre-trained language model improves RNN-based models'

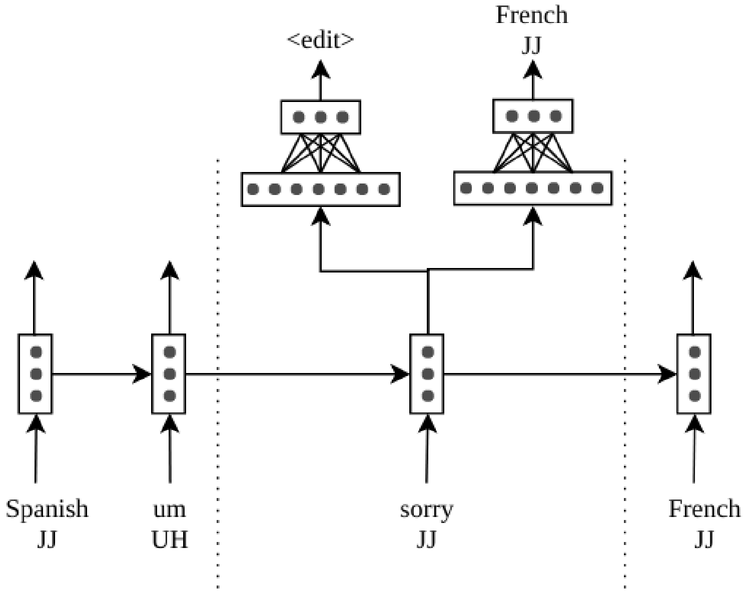

Figure 1: Multi-task LSTM model architecture

performance in a number of NLP tasks — either as the main feature representation for the downstream model, or as additional information in the form of a latent vector in the intermediate layers of complex models. The latter way was also employed by Peters et al. (2017) in the task of sequence labeling.

Finally, a multitask setup with language modelling as the second objective – the closest to our approach – was used by Rei (2017) to improve the performance of RNN-based Name Entity Recognition.

We note that there is no previous approach to multitask disfluency detection using a secondary task as general and versatile as language modelling. Furthermore, none of the works mentioned study how well their models *generalise* across datasets, nor do they shed much light on what kinds of disfluency structure are harder to detect, and why, as we try to do below.

3 Disfluency detection model

Our approach to disfluency detection is a sequence tagging model which makes single-word predictions given context words $w_{t-n+1}, ..., w_t$ of a maximum length n. We train it to perform two tasks jointly (c.f. Hough and Schlangen (2015)): (1) predicting the disfluency tag of the current word, $P(y_t|w_{t-n+1}, ..., w_t)$; and (2) predicting the next word in the sequence in a language model way, $P(w_{t+1}|w_{t-n+1}, ..., w_t)$.

At training time, we optimise the two tasks jointly, but at test time we only look at the resulting tags and ignore the LM predictions.

Our model uses a shared LSTM encoder (Hochreiter and Schmidhuber, 1997) with combined `word/POS-tag` tokens which provides context embedding for two independent multilayer perceptrons (MLPs) making the predictions for the two tasks. The combined token vocabulary (word+POS) size for the SWDA dataset is approximately 30% larger than the original word-only version — given this, concatenation is the simplest and most efficient way to pass part-of-speech information into the model.

The intuition behind adding an additional task to optimise for is that it *serves as a natural regulariser*: given an imbalanced label distribution (see Section 4 for the dataset description), only learning disfluency labels may lead to a higher degree of overfitting, and introducing an additional task with more uniformly distributed labels can help the model generalise better.

Other potential benefits of having the model work as an LM is the possibility of unsupervised model improvements, e.g. pre-training of the model's LM part from larger text corpora or 1-shot fine-tuning to new datasets with different word sequence patterns.

In order to address the problem of significantly imbalanced training data (the majority of the words in the corpus are fluent), we use a weighted cross-entropy loss in which the weight of a data point is inversely proportional to its label's frequency in the training set. Our overall loss function is of the form:

$$L = WL_{main} + \alpha L_{lm} + \frac{\lambda}{2}\sum_i w_i^2$$

– where WL_{main} and L_{lm} are respective losses for the disfluency tagging (class-weighted) and language modeling tasks (LM loss coefficient α is tuned empirically). The last term is L2 regularisation which we apply to the model's weight parameters w_i (those of word embeddings, LSTM gates, and MLPs) leaving all the biases intact. L2 coefficient λ is also tuned empirically (see Appendix A for the values of the constants).

The model is implemented in Tensorflow (Abadi et al., 2015) and is openly available.

4 Disfluency datasets and tags

4.1 The Switchboard dataset

For training our model, we use the Switchboard Dialog Acts dataset (SWDA) with manually annotated disfluency tags (Meteer et al., 1995). We use a pre-processed version of the dataset by Hough and Schlangen (2015) containing 90,497 utterances with transformed tagging: following their convention, there are 27 tags in total consisting of: `<f/>` tag for fluent tokens; `<e/>` for edit tokens; `<rm-{n}/>` tags for repair tokens that determine the start of the reparandum to be n tokens/words back; and `<rpSub>` & `<rpDel>` tags which mark the end of the `repair` and classify whether the repair is a *substitution* or *deletion* repair. The latter tokens can be combined with `<rm-{n}>` tokens, which explains the total of 27 tags - see (2) for an example where the `repair` word, 'Spanish', is tagged as `<rm-4><rpSub>` meaning this is a substitution repair that retraces 4 tokens back from the current token.

(2) $$\underset{\langle f/\rangle}{\text{with}}\quad [\underbrace{\underset{\langle f/\rangle}{\text{Italian}}}_{reparandum}\ + \{\underbrace{\underset{\langle e/\rangle}{\text{uh}}\ \underset{\langle e/\rangle}{\text{no}}\ \underset{\langle e/\rangle}{\text{uh}}}_{interregnum}\}\ \underbrace{\underset{\substack{\langle rm-4\rangle \\ \langle rpSub\rangle}}{\text{Spanish}}}_{repair}]\quad \underset{\langle f/\rangle}{\text{cuisine}}$$

The distribution of different types of tokens is highly imbalanced: only about 4% of all tokens are involved in disfluency structures (the detailed statistics are shown in the Appendix A). See above, Section 3 for how our model deals with this.

4.2 The bAbI+ dataset

To evaluate the cross data-set generalisation properties of our model and that of Hough and Schlangen (2015), we employ an additional dataset – bAbI+ introduced by Shalyminov et al. (2017). bAbI+ is an extension of the original bAbI Task 1 dialogues (Bordes et al., 2017) where different disfluency structures – such as hesitations, restarts, and corrections – can be mixed in probabilistically. Crucially these can be mixed in with complete control over the syntactic and semantic contexts in which the phenomena appear, and therefore the bAbI+ environment allows controlled, focused experimentation of the effect of different phenomena and their distributions on the performance of different models. Here, we use bAbI+ tools[2] to generate new data for the controlled generalisation experiment[3] of what kinds of disfluency phenomena are captured better by each model.

We focus here on the following disfluency patterns:

- **Hesitations**, e.g. as in "we will be *uhm* eight" (mixed in are single edit tokens);
- **Prepositional Phrase restarts (PP-restart)**, e.g. "in a *in a um in a* moderate price range" (repair of a PP at its beginning with or without an interregnum);

[2] See `https://bit.ly/babi_tools`

[3] Data is available at `http://bit.ly/babi_plus_disfluencies_study`

Model	F_e	F_{rm}	F_{rps}
(Hough and Schlangen, 2015)	0.902	0.711	0.689
(Schlangen and Hough, 2017)	0.918	—	0.719
LSTM	0.915	0.693	0.775
Multi-task LSTM	**0.919**	**0.753**	**0.816**

Table 1: Evaluation of the disfluency tagging models

Model	hesitations (F_e)	PP restarts			CL-restarts		
		F_e	F_{rm}	F_{rps}	F_e	F_{rm}	F_{rps}
(Hough and Schlangen, 2015)	0.917	0.774	0.875	0.877	0.938	0.471	0.630
LSTM	**0.956**	**1.0**	0.982	0.993	0.948	0.36	0.495
Multi-task LSTM	0.910	**1.0**	**0.993**	**0.997**	**0.991**	**0.484**	**0.659**

Table 2: Controlled generalisation evaluation

- **Clausal restarts (CL-restart)**, 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" (repair of the utterance from the beginning starting at arbitrary positions);

- **Corrections (NP and PP)**, e.g. "with Italian *sorry Spanish* cuisine", as was initially discussed in Section 1.

We generated independent bAbI+ datasets with each disfluency type. The disfluency phenomena above were chosen to resemble disfluency patterns in the original SWDA corpus (see Tables 3, 4, and 5 for examples), as well as intuitive considerations for the phenomena relevant for goal-oriented dialogue (namely, corrections).

The intuition for a generalisation experiment with data like this is as follows: while having similar disfluency patterns, our bAbI+ utterances differ from SWDA in the vocabulary and the word sequences themselves as they are in the domain of goal-oriented human-computer dialogue — this property makes it possible to evaluate the generalisation capabilities of a model outside its training domain.

5 Evaluation and experimental setup

We employ exactly the same evaluation criteria as Hough and Schlangen (2015): micro-averaged F1-scores for edit (F_e) and `<rm-{n}/>` tokens (F_{rm}) as well as for whole repair structures (F_{rps}). We compare our Multi-task LSTM model to its single-task version (disfluency tag predictions only) as well as to the system of Hough and Schlangen (2015) and the joint disfluency tagging/utterance segmentation model of Schlangen and Hough (2017) (all of the applicable word-level metrics on dialogue transcripts). These use a hand-crafted Markov Model for post-processing, whereas our model learns in an end-to-end fashion.

We train our model using the SGD optimiser and monitor the F_{rm} on the dev set as a stopping criterion. The model's hyperparameters are tuned heuristically, the final values are listed in the Appendix A. We use class weights in the main task's loss to deal with the highly imbalanced data, so that the weight of the k^{th} class is calculated as $W_k = 1/(C_k)^\gamma$, where C_k is the number of k^{th} class instances in the training set, and γ is a smoothing constant set empirically.

5.1 Results

The results are shown in Table 1. Both single- and multi-task LSTM are able to outperform the Hough and Schlangen (2015) model on edit tokens and repair structures, but the multi-task one performs significantly better on `<rm-{n}/>` tags and surpasses both previous models. The reason F_{rps} is higher than F_{rm} in general is that due to the tag conversion, fluent tokens inside reparandums and repairs are treated as part of repair, and they contribute to the global positive and negative counters used in the micro-averaged F1.

Repair length	Repair text	Frequency
1	i i *i*	139
	the the *the*	33
	and and *and*	31
	it it *it*	29
	its its *its*	26
2	it was *it was*	67
	i dont *i dont*	57
	i think *i think*	44
	in the *in the*	39
	do you *do you*	23
3	a lot of *a lot of*	7
	that was *uh that was*	5
	it was *uh it was*	5
	what do you *what do you*	4
	i i dont *i dont*	4

Table 3: Most common repairs in SWDA

POS pattern	Examples	repairs %
DT NN DT NN	this woman this socialite a can a garage the school that school	0.1
JJ NN JJ NN	high school high school good comedy good humor israeli situation palestinian situation	0.03
DT UH DT NN	that uh that punishment the uh the cauliflower that uh that adjustment	0.02
DT NN UH DD NN	a friend uh a friend a lot uh a lot a lot um a lot	0.01
NN PRP VBP NN NN	ribbon you know hair ribbon thing you know motion detector	0.01

Table 4: SWDA repairs by POS-tag pattern

Keyword pattern	Examples	repairs %
sorry<e/> *	or *im sorry* no *um im sorry* what thank you *im sorry* i just got home from work	0.02
sorry<e/> *<rm-*/>	and he told us theres two sixteen bit slots and two eight bit *sorry two four sixteen bit slots and two eight bit* slots available for the user	0.009
i<e/> mean<e/> *	i mean i mean yeah i mean uh i mean i	4
i<e/> mean<e/> *<rm-*/>	i mean i i but i mean whats whats happened here is is is i mean you youve	0.5

Table 5: SWDA repairs by interregnum

Controlled generalisation experiment results are shown in Table 2 — note that we could only run the model of Hough and Schlangen (2015) on bAbI+ data because that of Schlangen and Hough (2017) works in a setup different from ours. It can be seen that the LSTM tagger is somewhat overfitted to edit tokens on SWDA. This is the reason it outperforms the Multi-task LSTM on the hesitations dataset and has a tied 1.0 on edit tokens on PP restarts dataset. In all other cases, Multi-task LSTM demonstrates superior generalisation.

As for NP/PP self-corrections which are not present in Table 2: none of the systems tested were able to handle these. Evaluation on the this dataset revealed 0.0 accuracy with all systems. We discuss these results below.

6 Discussion and future work

We have presented a multi-task LSTM-based disfluency detection model which outperforms previous neural network-based incremental models while being significantly simpler than them.

For the first time, we have demonstrated the generalisation potential of a disfluency detection model by cross-dataset evaluation. As the results show, all models achieve reasonably high generalisation level on the very local disfluency patterns such as hesitations and PP restarts. However, the accuracy drops significantly on less restricted restarts spanning arbitrary regions of utterances from the beginning. On the majority of those disfluency patterns, our model achieves a superior generalisation level.

Interestingly, none of the models were able to detect NP or PP corrections such as those often glossed in disfluency papers (e.g. “A flight to Boston uh I mean to Denver”). The most likely explanation for this could be the extreme sparsity of such disfluencies in the SWDA dataset.

We performed analysis of SWDA disfluencies in order to explore this hypothesis and examined their distribution based on length in tokens and POS-tag sequence patterns of interest. As shown in Tables 3 and 4, the vast majority of disfluencies found are just repetitions without speakers actually correcting themselves. This observation is in line with prior studies, showing that the distribution of repair types varies significantly across domains (Colman and Healey, 2011), modalities (Oviatt, 1995), and gender & age groups (Bortfeld et al., 2001) — see Purver et al. (2018) for a nice discussion.

While this is very likely the correct explanation, we cannot rule out the possibility that such self-corrections are inherently more difficult to process for particular models - that needs a separate experiment that holds frequency of particular repair structures constant in the training data.

Addressing this issue is our next step since we designed the multi-task LSTM with this in mind. As such, we will explore possibilities of knowledge transfer to new closed domains in a 1-shot setting, both with regular supervised training and unsupervised LM fine-tuning.

7 Acknowledgements

We are very grateful to Julian Hough for offerring extensive help with running their 2015 disfluency detection model (Hough and Schlangen, 2015) and their SWDA tagging conventions, as well as constructive feedback throughout.

References

[Abadi et al.2015] 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, Dandelion Mané, 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 Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

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

[Bortfeld et al.2001] Heather Bortfeld, Silvia D Leon, Jonathan E Bloom, Michael F Schober, and Susan E Brennan. 2001. Disfluency rates in conversation: Effects of age, relationship, topic, role, and gender. *Language and speech*, 44(2):123–147.

[Brennan and Schober2001] S.E. Brennan and M.F. Schober. 2001. How listeners compensate for disfluencies in spontaneous speech. *Journal of Memory and Language*, 44(2):274–296.

[Charniak and Johnson2001] E. Charniak and M. Johnson. 2001. Edit detection and parsing for transcribed speech. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–9. Association for Computational Linguistics.

[Colman and Healey2011] M. Colman and P. G. T. Healey. 2011. The distribution of repair in dialogue. In *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society*, pages 1563–1568, Boston, MA.

[Crocker et al.2000] Matthew Crocker, Martin Pickering, and Charles Clifton, editors. 2000. *Architectures and Mechanisms in Sentence Comprehension*. Cambridge University Press.

[Eshghi et al.2017] Arash Eshghi, Igor Shalyminov, and Oliver Lemon. 2017. Bootstrapping incremental dialogue systems from minimal data: the generalisation power of dialogue grammars. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2220–2230.

[Fernández and Ginzburg2002] Raquel Fernández and Jonathan Ginzburg. 2002. Non-sentential utterances: Grammar and dialogue dynamics in corpus annotation. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, pages 253–259.

[Fernández2006] Raquel Fernández. 2006. *Non-Sentential Utterances in Dialogue: Classification, Resolution and Use*. Ph.D. thesis, King's College London, University of London.

[Ferreira et al.2004] Fernanda Ferreira, Ellen F. Lau, and Karl G. D. Bailey. 2004. Disfluencies, language comprehension, and tree adjoining grammars. *Cognitive Science*, 28(5):721ñ749.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

[Hough and Purver2014] Julian Hough and Matthew Purver. 2014. Strongly incremental repair detection. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 78–89.

[Hough and Schlangen2015] Julian Hough and David Schlangen. 2015. Recurrent neural networks for incremental disfluency detection. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 849–853.

[Howes et al.2010] Christine Howes, Patrick G. T. Healey, and Matthew Purver. 2010. Tracking lexical and syntactic alignment in conversation. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, Portland, OR.

[Kempson et al.2017] Ruth Kempson, Eleni Gregoromichelaki, Arash Eshghi, and Julian Hough. 2017. Ellipsis in Dynamic Syntax. In *Oxford Handbook of Ellipsis*. Oxford University Press.

[Meteer et al.1995] M. Meteer, A. Taylor, R. MacIntyre, and R. Iyer. 1995. Disfluency annotation stylebook for the switchboard corpus.

[Oviatt1995] Sharon Oviatt. 1995. Predicting spoken disfluencies during human–computer interaction. *Computer Speech & Language*, 9(1):19–35.

[Peters et al.2017] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1756–1765.

[Peters et al.2018] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

[Purver et al.2018] Matthew Purver, Julian Hough, and Christine Howes. 2018. Computational models of miscommunication phenomena. In Patrick G. T. Healey, Jan de Ruiter, and Gregory J. Mills, editors, *Topics in Cognitive Science (topiCS)*, volume 10.

[Rei2017] Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 2121–2130.

[Schlangen and Hough2017] David Schlangen and Julian Hough. 2017. Joint, incremental disfluency detection and utterance segmentation from speech. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 326–336.

[Schlangen and Skantze2009] 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 ACL (EACL 2009)*, pages 710–718, Athens, Greece, March. Association for Computational Linguistics.

[Shalyminov et al.2017] Igor Shalyminov, Arash Eshghi, and Oliver Lemon. 2017. Challenging Neural Dialogue Models with Natural Data: Memory Networks Fail on Incremental Phenomena. In *Proceedings of the 21st Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2017 - SaarDial)*.

[Shriberg1994] Elizabeth Shriberg. 1994. *Preliminaries to a Theory of Speech Disfluencies*. Ph.D. thesis, University of California, Berkeley.

[Skantze and Hjalmarsson2010] 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.

[Wang et al.2016] Shaolei Wang, Wanxiang Che, and Ting Liu. 2016. A neural attention model for disfluency detection. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 278–287.

[Wang et al.2017] Shaolei Wang, Wanxiang Che, Yue Zhang, Meishan Zhang, and Ting Liu. 2017. Transition-based disfluency detection using lstms. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2785–2794.

Appendix A

Parameter	**Value**
optimiser	stochastic gradient descent
loss function	weighted cross-entropy
vocabulary size	6157
embedding size	128
MLP layer sizes	[128]
learning rate	0.01
learning rate decay	0.9
batch size	32
α	0.1
λ	0.001
γ	1.05

Table 6: Multi-task LSTM training setup

Label type	**Label**	**Frequency**
fluent token	`<f/>`	574771
edit token	`<e/>`	45729
single-token substitution	`<rm-{1-8}/><rpEndSub/>`	13003
single-token deletion	`<rm-{1-8}/><rpEndDel/>`	1011
multi-token substitution start	`<rm-{1-8}/><rpMid/>`	6976
multi-token substitution end	`<rpEndSub>`	6818

Table 7: SWDA labels