Disfluency Detection using a Noisy Channel Model and a Deep Neural Language Model

Paria Jamshid Lou
Department of Computing
Macquarie University
Sydney, Australia
paria.jamshid-lou@hdr.mq.edu.au

Mark Johnson
Department of Computing
Macquarie University
Sydney, Australia
mark.johnson@mq.edu.au

Abstract

This paper presents a model for disfluency detection in spontaneous speech transcripts called *LSTM Noisy Channel Model*. The model uses a Noisy Channel Model (NCM) to generate *n*-best candidate disfluency analyses and a Long Short-Term Memory (LSTM) language model to score the underlying fluent sentences of each analysis. The LSTM language model scores, along with other features, are used in a MaxEnt reranker to identify the most plausible analysis. We show that using an LSTM language model in the reranking process of noisy channel disfluency model improves the state-of-the-art in disfluency detection.

1 Introduction

Disfluency is a characteristic of spontaneous speech which is not present in written text. Disfluencies are informally defined as interruptions in the normal flow of speech that occur in different forms, including false starts, corrections, repetitions and filled pauses. According to Shriberg's (1994) definition, the basic pattern of speech disfluencies contains three parts: *reparandum*[1], *interregnum* and *repair*. Example 1 illustrates a disfluent structure, where the reparandum *to Boston* is the part of the utterance that is replaced, the interregnum *uh, I mean* is an optional part of a disfluent structure that consists of a filled pause *uh* and a discourse marker *I mean* and the repair *to Denver* replaces the reparandum. The fluent version of Example 1 is obtained by deleting reparandum and interregnum words.

$$\underbrace{\textit{I want a flight } \overbrace{\textit{to Boston,}}^{\text{reparandum}}}_{\text{interregnum}} \underbrace{\textit{uh, I mean}}_{\text{interregnum}} \underbrace{\textit{to Denver on Friday}}_{\text{repair}} \quad (1)$$

While disfluency rate varies with the context, age and gender of speaker, Bortfeld et al. (2001) reported disfluencies once in every 17 words. Such frequency is high enough to reduce the readability of speech transcripts. Moreover, disfluencies pose a major challenge to natural language processing tasks, such as dialogue systems, that rely on speech transcripts (Ostendorf et al., 2008). Since such systems are usually trained on fluent, clean corpora, it is important to apply a speech disfluency detection system as a pre-processor to find and remove disfluencies from input data. By disfluency detection, we usually mean identifying and deleting reparandum words. Filled pauses and discourse markers belong to a closed set of words, so they are trivial to detect (Johnson and Charniak, 2004).

In this paper, we introduce a new model for detecting restart and repair disfluencies in spontaneous speech transcripts called *LSTM Noisy Channel Model (LSTM-NCM)*. The model uses a Noisy Channel Model (NCM) to generate *n*-best candidate disfluency analyses, and a Long Short-Term Memory (LSTM) language model to rescore the NCM analyses. The language model scores are used as features in a MaxEnt reranker to select the most plausible analysis. We show that this novel approach improves the current state-of-the-art.

2 Related Work

Approaches to disfluency detection task fall into three main categories: sequence tagging, parsing-based and noisy channel model. The sequence

[1] Reparandum is sometimes called *edit*.

tagging models label words as fluent or disfluent using different techniques, including conditional random fields (Ostendorf and Hahn, 2013; Zayats et al., 2014; Ferguson et al., 2015), hidden Markov models (Liu et al., 2006; Schuler et al., 2010) or recurrent neural networks (Hough and Schlangen, 2015; Zayats et al., 2016). Although sequence tagging models can be easily generalized to a wide range of domains, they require a specific state space for disfluency detection, such as begin-inside-outside (BIO) style states that label words as being inside or outside of a reparandum word sequence. The parsing-based approaches refer to parsers that detect disfluencies, as well as identifying the syntactic structure of the sentence (Rasooli and Tetreault, 2013; Honnibal and Johnson, 2014; Yoshikawa et al., 2016). Training a parsing-based model requires large annotated tree-banks that contain both disfluencies and syntactic structures. Noisy channel models (NCMs) use the similarity between reparandum and repair as an indicator of disfluency. However, applying an effective language model (LM) inside an NCM is computationally complex. To alleviate this problem, some researchers use more effective LMs to rescore the NCM disfluency analyses. Johnson and Charniak (2004) applied a syntactic parsing-based LM trained on the fluent version of the Switchboard corpus to rescore the disfluency analyses. Zwarts and Johnson (2011) trained external n-gram LMs on a variety of large speech and non-speech corpora to rank the analyses. Using the external LM probabilities as features to the reranker improved the baseline NCM (Johnson and Charniak, 2004). The idea of applying external language models in the reranking process of the NCM motivates our model in this work.

3 LSTM Noisy Channel Model

We follow Johnson and Charniak (2004) in using an NCM to find the n-best candidate disfluency analyses for each sentence. The NCM, however, lacks an effective language model to capture more complicated language structures. To overcome this problem, our idea is to use different LSTM language models to score the underlying fluent sentences of the analyses proposed by the NCM and use the language model scores as features to a MaxEnt reranker to select the best analysis. In the following, we describe our model and its components in details.

In the NCM of speech disfluency, we assume that there is a well-formed source utterance X to which some noise is added and generates a disfluent utterance Y as follows.

X = a flight to Denver
Y = a flight *to Boston uh I mean* to Denver

Given Y, the goal of the NCM is to find the most likely source sentence $\hat{X}$ such that:

$$\hat{X} = \arg\max_{X} P(Y|X)P(X) \tag{2}$$

As shown in Equation 2, the NCM contains two components: the channel model $P(Y|X)$ and the language model $P(X)$. Calculating the channel model and language model probabilities, the NCM generates 25-best candidate disfluency analyses as follows.

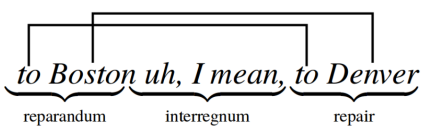
1. ~~a flight~~ to Boston uh I mean to Denver
2. a flight ~~to Boston~~ uh I mean to Denver
3. a flight to Boston ~~uh I mean~~ to Denver

... (3)

Example 3 shows sample outputs of the NCM, where potential reparandum words are specified with strikethrough text. The MaxEnt reranker is applied on the candidate analyses of the NCM to select the most plausible one.

3.1 Channel Model

We assume that X is a substring of Y, so the source sentence X is obtained by deleting words from Y. For each sentence Y, there are only a finite number of potential source sentences. However, with the increase in the length of Y, the number of possible source sentences X grows exponentially, so it is not feasible to do exhaustive search. Moreover, since disfluent utterances may contain an unbounded number of crossed dependencies, a context-free grammar is not suitable for finding the alignments. The crossed dependencies refer to the relation between repair and reparandum words which are usually the same or very similar words in roughly the same order as in Example 4.



(4)

We apply a Tree Adjoining Grammar (TAG) based transducer (Johnson and Charniak, 2004)

which is a more expressive formalism and provides a systematic way of formalising the channel model. The TAG channel model encodes the crossed dependencies of speech disfluency, rather than reflecting the syntactic structure of the sentence. The TAG transducer is effectively a simple first-order Markov model which generates each word in the reparandum conditioned on the preceding word in the reparandum and the corresponding word in the repair. More details about the TAG channel model can be found in (Johnson and Charniak, 2004).

3.2 Language Model

The language model of the NCM evaluates the fluency of the sentence with disfluency removed. The language model is expected to assign a very high probability to a fluent sentence X (e.g. *a flight to Denver*) and a lower probability to a sentence Y which still contains disfluency (e.g. *a flight to Boston uh I mean to Denver*). However, it is computationally complex to use an effective language model within the NCM. The reason is the polynomial-time dynamic programming parsing algorithms of TAG can be used to search for likely repairs if they are used with simple language models such as a bigram LM (Johnson and Charniak, 2004). The bigram LM within the NCM is too simple to capture more complicated language structure. In order to alleviate this problem, we follow Zwarts and Johnson (2011) by training LMs on different corpora, but we apply state-of-the-art recurrent neural network (RNN) language models.

LSTM

We use a long short-term memory (LSTM) neural network for training language models. LSTM is a particular type of recurrent neural networks which has achieved state-of-the-art performance in many tasks including language modelling (Mikolov et al., 2010; Jozefowicz et al., 2016). LSTM is able to learn long dependencies between words, which can be highly beneficial for the speech disfluency detection task. Moreover, it allows for adopting a distributed representation of words by constructing word embedding (Mikolov et al., 2013).

We train forward and backward (i.e. input sentences are given in reverse order) LSTM language models using truncated backpropagation through time algorithm (Rumelhart et al., 1986) with mini-batch size 20 and total number of epochs 13. The LSTM model has two layers and 200 hidden units. The initial learning rate for stochastic gradient optimizer is chosen to 1 which is decayed by 0.5 for each epoch after maximum epoch 4. We limit the maximum sentence length for training our model due to the high computational complexity of longer histories in the LSTM. In our experiments, considering maximum 50 words for each sentence leads to good results. The size of word embedding is 200 and it is randomly initialized for all LSTM LMs[2].

Using each forward and backward LSTM language model, we assign a probability to the underlying fluent parts of each candidate analysis.

3.3 Reranker

In order to rank the the 25-best candidate disfluency analyses of the NCM and select the most suitable one, we apply the MaxEnt reranker proposed by Johnson et al. (2004). We use the feature set introduced by Zwarts and Johnson (2011), but instead of n-gram scores, we apply the LSTM language model probabilities. The features are so good that the reranker without any external language model is already a state-of-the-art system, providing a very strong baseline for our work. The reranker uses both model-based scores (including NCM scores and LM probabilities) and surface pattern features (which are boolean indicators) as described in Table 1. Our reranker optimizes the expected f-score approximation described in Zwarts and Johnson (2011) with L2 regularisation.

4 Corpora for Language Modelling

In this work, we train forward and backward LSTM language models on Switchboard (Godfrey and Holliman, 1993) and Fisher (Cieri et al., 2004) corpora. Fisher consists of 2.2×10^7 tokens of transcribed text, but disfluencies are not annotated in it. Switchboard is a widely available corpus (1.2×10^6 tokens) where disfluencies are annotated according to Shriberg's (1994) scheme. Since the bigram language model of the NCM is trained on this corpus, we cannot directly use Switchboard to build LSTM LMs. The reason is that if the training data of Switchboard is used both for predicting language fluency and optimizing the loss function, the reranker will overestimate the

[2]All code is written in TensorFlow (Abadi et al., 2015)

model-based features
1-2. forward & backward LSTM LM scores 3-7. log probability of the entire NCM 8. sum of the log LM probability & the log channel model probability plus number of edits in the sentence 9. channel model probability
surface pattern features
10. CopyFlags_X_Y: if there is an exact copy in the input text of length X ($1 \leq X \leq 3$) and the gap between the copies is Y ($0 \leq Y \leq 3$) 11. WordsFlags_L_n_R: number of flags to the left (L) and to the right (R) of a 3-gram area ($0 \leq L, R \leq 1$) 12. SentenceEdgeFlags_B_L: it captures the location and length of disfluency. The Boolean B sentence initial or sentence final disfluency, L ($1 \leq L \leq 3$) records the length of the flags.

Table 1: The features used in the reranker. They, except for the first and second one, were applied by Zwarts and Johnson (2011).

weight related to the LM features extracted from Switchboard. This is because the fluent sentence itself is part of the language model (Zwarts and Johnson, 2011). As a solution, we apply a k-fold cross-validation ($k = 20$) to train the LSTM language models when using Switchboard corpus.

We follow Charniak and Johnson (2001) in splitting Switchboard corpus into training, development and test set. The training data consists of all sw[23]$*$.dps files, development training consists of all sw4[5-9]$*$.dps files and test data consists of all sw4[0-1]$*$.dps files. Following Johnson and Charniak (2004), we remove all partial words and punctuation from the training data. Although partial words are very strong indicators of disfluency, standard speech recognizers never produce them in their outputs, so this makes our evaluation both harder and more realistic.

5 Results and Discussion

We assess the proposed model for disfluency detection with all MaxEnt features described in Table 1 against the baseline model. The noisy channel model with exactly the same reranker features except the LSTM LMs forms the baseline model.

To evaluate our system, we use two metrics *f-score* and *error rate*. Charniak and Johnson (2001) used the f-score of labelling reparanda or "edited" words, while Fiscus et al (2004) defined an "error rate" measure, which is the number of words falsely labelled divided by the number of reparanda words. Since only 6% of words are disfluent in Switchboard corpus, accuracy is not a good measure of system performance. F-score, on the other hand, focuses more on detecting "edited" words, so it is a decent metric for highly skewed data.

According to Tables 2 and 3, the LSTM noisy channel model outperforms the baseline. The experiment on Switchboard and Fisher corpora demonstrates that the LSTM LMs provide information about the global fluency of an analysis that the local features of the reranker do not capture. The LSTM language model trained on Switchboard corpus results in the greatest improvement. Switchboard is in the same domain as the test data and it is also disfluency annotated. Either or both of these might be the reason why Switchboard seems to be better in comparison with Fisher which is a larger corpus and might be expected to make a better language model. Moreover, the backward LSTMs have better performance in comparison with the forward ones. It seems when sentences are fed in reverse order, the model can more easily detect the unexpected word order associated with the reparandum to detect disfluencies. In other words, that the disfluency is observed "after" the fluent repair in a backward language model is helpful for recognizing disfluencies.

baseline	85.3		
corpus	**forward**	**backward**	**both**
Switchboard	86.1	86.6	86.8
Fisher	86.2	86.5	86.3

Table 2: F-scores on the dev set for a variety of LSTM language models.

baseline	27.0		
corpus	**forward**	**backward**	**both**
Switchboard	25.5	24.8	24.3
Fisher	25.6	25.0	25.3

Table 3: Expected error rates on the dev set for a variey of LSTM language models.

We compare the performance of Kneser-Ney

smoothed 4-gram language models with the LSTM corresponding on the reranking process of the noisy channel model. We estimate the 4-gram models and assign probabilities to the fluent parts of disflueny analyses using the SRILM toolkit (Stolcke, 2002). As Tables 4 and 5 show including scores from a conventional 4-gram language model does not improve the model's ability to find disfluencies, suggesting that the LSTM model contains all the useful information that the 4-gram model does. In order to give a more general idea on the performance of LSTM over standard LM, we evaluate our model when the language model scores are used as the only features of the reranker. The f-score for the NCM alone without applying the reranker is 78.7, while using 4-gram language model scores in the reranker increases the f-score to 81.0. Replacing the 4-gram scores with LSTM language model probabilities leads to further improvement, resulting an f-score 82.3.

baseline	85.3		
corpus	**4-gram**	**LSTM**	**both**
Switchboard	85.1	86.8	86.1
Fisher	85.6	86.3	86

Table 4: F-score for 4-gram, LSTM and combination of both language models.

baseline	27.0		
corpus	**4-gram**	**LSTM**	**both**
Switchboard	27.5	24.3	26
Fisher	26.6	25.3	26

Table 5: Expected error rates for 4-gram, LSTM and combination of both language models.

We also compare our best model on the development set to the state-of-the-art methods in the literature. As shown in Table 6, the LSTM noisy channel model outperforms the results of prior work, achieving a state-of-the-art performance of 86.8. It also has better performance in comparison with Ferguson et al. (2015) and Zayat et al.'s (2016) models, even though they use richer input that includes prosodic features or partial words.

Model	**f-score**
Yoshikawa et al. (2016)	62.5
Johnson and Charniak (2004)	79.7
Johnson et al. (2004)	81.0
Rasooli and Tetreault (2013)	81.4
Qian and Liu (2013)	82.1
Honnibal and Johnson (2014)	84.1
Ferguson et al. (2015) *	85.4
Zwarts and Johnson (2011)	85.7
Zayats et al. (2016) *	85.9
LSTM-NCM	**86.8**

Table 6: Comparison of the LSTM-NCM to state-of-the-art methods on the dev set. *Models have used richer input.

6 Conclusion and Future Work

In this paper, we present a new model for disfluency detection from spontaneous speech transcripts. It uses a long short-term memory neural network language model to rescore the candidate disfluency analyses produced by a noisy channel model. The LSTM language model scores as features in a MaxEnt reranker improves the model's ability to detect and correct restart and repair disfluencies. The model outperforms other models reported in the literature, including models that exploit richer information from the input. As future work, we apply more complex LSTM language models such as sequence-to-sequence on the reranking process of the noisy channel model. We also intend to investigate the effect of integrating LSTM language models into other kinds of disfluency detection models, such as sequence labelling and parsing-based models.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments and suggestions.

References

Martin 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 systems. Software available from tensorflow.org.

Heather Bortfeld, Silvia Leon, Jonathan Bloom, Michael Schober, and Susan Brennan. 2001. Disfluency rates in conversation: Effects of age, relationship, topic, role, and gender. *Language and Speech* 44(2):123–147.

Eugene Charniak and Mark Johnson. 2001. Edit detection and parsing for transcribed speech. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*. Stroudsburg, USA, NAACL'01, pages 118–126. http://aclweb.org/anthology/N01-1016.

Christopher Cieri, David Miller, and Kevin Walker. 2004. Fisher English training speech part 1 transcripts LDC2004T19. Published by: Linguistic Data Consortium, Philadelphia, USA.

James Ferguson, Greg Durrett, and Dan Klein. 2015. Disfluency detection with a semi-Markov model and prosodic features. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, USA, NAACL'15, pages 257–262.

Jonathan Fiscus, John Garofolo, Audrey Le, Alvin Martin, David Pallet, Mark Przybocki, and Greg Sanders. 2004. Results of the fall 2004 STT and MDE evaluation. In *Proceedings of Rich Transcription Fall Workshop*.

John Godfrey and Edward Holliman. 1993. Switchboard-1 release 2 LDC97S62. Published by: Linguistic Data Consortium, Philadelphia, USA.

Matthew Honnibal and Mark Johnson. 2014. Joint incremental disfluency detection and dependency parsing. *Transactions of the Association for Computational Linguistics* 2(1):131–142. http://www.aclweb.org/anthology/Q14-1011.

Julian Hough and David Schlangen. 2015. Recurrent neural networks for incremental disfluency detection. In *Proceedings of the 16th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Dresden, Germany, pages 845–853.

Mark Johnson and Eugene Charniak. 2004. A TAG-based noisy channel model of speech repairs. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Barcelona, Spain, ACL'04, pages 33–39. http://aclweb.org/anthology/P04-1005.

Mark Johnson, Eugene Charniak, and Matthew Lease. 2004. An improved model for recognizing disfluencies in conversational speech. In *Proceedings of Rich Transcription Workshop*.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *CoRR* abs/1602.02410.

Yang Liu, Elizabeth Shriberg, Andreas Stolckeand, Dustin Hillard, Mari Ostendorf, and Mary Harper. 2006. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 14(5):1526–1540.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Makuhari, Japan, pages 1045–1048.

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 27th Annual Conference on Neural Information Processing Systems (NIPS)*. Curran Associates Inc., pages 3111–3119.

Mari Ostendorf, Benoit Favre, Ralph Grishman, Dilek Hakkani-Tur, Mary Harper, Dustin Hillard, Julia Hirschberg, Heng Ji, Jeremy G. Kahn, Yang Liu, Sameer Maskey, Evgeny Matusov, Hermann Ney, Andrew Rosenberg, Elizabeth Shriberg, Wen Wang, and Chuck Wooters. 2008. Speech segmentation and its impact on spoken document processing. *IEEE Signal Processing Magazine* 25(3):59–69.

Mari Ostendorf and Sangyun Hahn. 2013. A sequential repetition model for improved disfluency detection. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Lyon, France, pages 2624–2628.

Xian Qian and Yang Liu. 2013. Disfluency detection using multi-step stacked learning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, USA, NAACL'13, pages 820–825. http://aclweb.org/anthology/N13-1102.

Mohammad Sadegh Rasooli and Joel Tetreault. 2013. Joint parsing and disfluency detection in linear time. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, USA, pages 124–129. http://aclweb.org/anthology/D13-1013.

David Rumelhart, James McClelland, and PDP Research Group. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press.

William Schuler, Samir AbdelRahman, Tim Miller, and Lane Schwartz. 2010. Broad-coverage parsing using human-like memory constraints. *Computational Linguistics* 36(1):1–30.

Elizabeth Shriberg. 1994. *Preliminaries to a theory of speech disfluencies*. Ph.D. thesis, University of California, Berkeley, USA.

Andreas Stolcke. 2002. SRILM: An extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*. Association for Computational Linguistics, Denver, Colorado, USA, volume 2, pages 901–904.

Masashi Yoshikawa, Hiroyuki Shindo, and Yuji Matsumoto. 2016. Joint transition-based dependency parsing and disfluency detection for automatic speech recognition texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1036–1041. http://aclweb.org/anthology/D16-1109.

Victoria Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2014. Multi-domain disfluency and repair detection. In *Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Singapore, pages 2907–2911.

Victoria Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2016. Disfluency detection using a bidirectional LSTM. In *Proceedings of the 16th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. San Francisco, USA, pages 2523–2527.

Simon Zwarts and Mark Johnson. 2011. The impact of language models and loss functions on repair disfluency detection. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, USA, volume 1 of *HLT'11*, pages 703–711. http://aclweb.org/anthology/P11-1071.