Disfluency Detection for Spoken Learner English

Y. Lu, M.J.F. Gales, K.M. Knill, P. Manakul, Y. Wang

ALTA Institute / Engineering Department
Cambridge University, UK
{ytl28,mjfg,kate.knill,pm574,yw396}@eng.cam.ac.uk

Abstract

One of the challenges for computer aided language learning (CALL) is providing high quality feedback to learners. An obstacle to improving feedback is the lack of labelled training data for tasks such as spoken "grammatical" error detection and correction. One approach to addressing this lack of data is to convert the output of an automatic speech recognition (ASR) system into a form that is closer to text data, for which there is significantly more labelled data available. Disfluency detection, locating regions of the speech where for example false starts and repetitions occur, and subsequent removal of the associated words, helps to make speech transcriptions more text-like. Additionally, ASR systems do not usually generate sentence-like units, the output is simply a sequence of words associated with the particular speech segmentation used for coding. This motivates the needs for automated systems for sentence segmentation. By combining these approaches, advanced text processing techniques should perform significantly better on the output from spoken language processing systems. Unfortunately labelled data for these tasks is not available for learners of English. In this work disfluency detection and "sentence" segmentation systems trained on data from native speakers are applied to spoken grammatical error detection and correction tasks for learners of English. Performance gains using these approaches are shown on a free speaking test.

Index Terms: speech recognition, grammatical error detection, computer-assisted language learning (CALL)

1. Introduction

Disfluencies often present in spontaneous speech. A standard disfluency structure [1] comprises reparandum, interregnum and repair. For example:

$$\text{I want a train } [\underbrace{\textit{to Oxford}}_{\text{reparandum}} \underbrace{\textit{uh I mean}}_{\text{interregnum}} + \underbrace{\text{to Cambridge}}_{\text{repair}}]$$

Removing the reparandum and interregnum parts recovers the underlying fluent sentence.

Since they are usually trained on clean and fluent corpora, the presence of disfluencies in speech degrades downstream natural language processing tasks [2]. Disfluency detection (DD), followed by the removal of detected reparandum and interregnum components, is therefore an interesting option to make speech transcriptions more text-like and consequently help improve downstream systems. For this study, we are interested in applying DD to non-native spoken English for computer aided language learning (CALL), to provide feedback on a learner's spoken language. In particular on the downstream tasks of grammatical error detection (GED) on free speaking and conversational tests.

Interregnum regions often consist of fixed phrases of filled pauses and discourse markers ('um','you know' etc), so are easy to detect using rule-based methods [3]. Automatic DD therefore focuses on reparandum detection. Parsing-based methods [4, 5, 6] identify disfluency structures by learning the "syntactic structure" of spoken sentences. They are efficient, jointly performing parsing and DD but training requires a large amount of annotated tree-banks. These are not generally available, particularly for non-native speech which has a different disfluency structure [7]. Alternatively, sequence labelling methods assign fluent/disfluent tags to each word [8, 9]. This was the approach adopted as it can be easily generalised across domains.

Since the text of free and conversational speech is unknown at test time, automatic speech recognition (ASR) must be used to transcribe the speech. Compared to native speakers, non-native learners generate more disfluencies, which co-occur with grammatical errors, and pronunciation errors making it hard to recognise the speech accurately. In addition, ASR transcriptions are not typically segmented into "sentence-like" units so need to be automatically segmented prior to DD. These aspects further complicate the spoken language processing so providing reliable feedback is challenging. Only labelled evaluation data is currently available for GED on non-native speech. Therefore, in this work the automatic segmentation and DD is trained on native speech data, for which training data exists. The use of ASR transcriptions is compared to the manual transcriptions used in [9, 6]. For the feedback tasks, the speech segmenter and DD are applied to the output of an ASR system trained on non-native learner English and fed into non-native spoken GED system.

Our contributions in this paper are: 1) we propose an advanced sequence tagging-based disfluency detection method with improved robustness, achieved by combining part-of-speech based language models and pattern match features; 2) we evaluate the disfluency model against ASR, as well as manual, transcriptions; 3) we present initial investigations into applying disfluency detection to non-native English learner speech. Performance gains are shown to be achieved in downstream grammatical error detection (GED) task by performing automatic disfluency detection and speech segmentation.

2. Feedback Framework

Figure 1: *Feedback pipeline*

This paper reports on research supported by Cambridge Assessment, University of Cambridge. Thanks to Cambridge English Language Assessment for support and access to the BULATS data.

A modular feedback framework is shown in Figure 1. It is composed of an automatic speech recognition (ASR) module, an automatic speech segmentation (SEG) module, a disfluency detection (DD) module as well as a downstream task-specific CALL module. In practice, the SEG and DD modules, once trained, can be universally applied regardless of the corpora or the downstream task; the ASR module is often trained on corpora from the target usage domain; and the downstream task-specific system needs to be trained separately and can be easily adapted to work with the pipeline. This section mainly discusses the SEG and DD modules; their performance is discussed in Section 3, and their impact on non-native CALL tasks presented in Section 4.

2.1. Automatic speech segmentation

It is often found that performance of natural language tasks degrades with sentence length. By default, an ASR system produces transcriptions separated by conversational turns or maximum segment length. Each transcription might cover multiple sentences. To prevent the DD performance from degrading in the case of long runs of sentences in ASR transcriptions, automatic segmentation in the form of a sentence boundary detector is first applied. The segmenter aims to predict sentence-like "speech" units (SUs) [10] within each conversational turn or response. Here SU detection is modeled as a sequence labeling task. Words immediately before a speech unit boundary (including end-of-turn words) are labeled as SU; others are labeled as non-SU. A bi-directional LSTM is adopted for this task.

Prosodic and lexical-based features were used to detect SUs. Pause duration was calculated using time stamps from force aligned ASR transcriptions. Prosodic information was extracted from audio files: for each word, f_0 trajectories were stylized by median-filtering followed by piecewise linearisation [11]. The complete set of features are listed in Table 1. All numerical features were converted into categorical features by running K-means clustering followed by nearest neighbour grouping. For each feature, an additional null class is added to handle start-of-turn, end-of-turn and unvoiced region.

1. Word
2. Pause duration before / after word
3. f_0 slope mean/min/max normalized over pitch range
4. log differences (division / subtraction) of f_0 properties (mean / min / max / start / end) between neighbouring words
5. f_0 slope difference between neighbouring words

Table 1: *Speech unit detection features.*

2.2. Disfluency detection

Here disfluency detection (DD) is modeled as a sequence labelling task i.e. each word token has to be labelled with a disfluent/fluent tag. Begin-inside-outside (BIO) labels [8] are used: BE (begin edit), IE (in edit), EE (end of edit), SE (single word edit) or O (other). Complex disfluencies sometimes occur in succession or contain nested structures [2]. Flattening the nested structure of a repetition region helps to improve DD [4]. We further flatten the entire disfluency region by keeping only the top-level reparandum, interregnum and repair e.g.:

Nested: *[a req- + [[a + a] + a] requirement]*

Flattened: *[a req- a a + a requirement]*
Labels: BE IE IE EE O O

Following [9], a bi-directional LSTM is used as the classifier in this work. Features consist of word tokens, part of speech (PoS) tags as well as N-gram based patterns. To improve cross-domain robustness, we use PoS-based language models. Character-level embedding has been shown to achieve gains in sequence labeling tasks on text [12]. We extend its application to speech transcriptions by concatenating character and word level embeddings to provide a richer representation for each word token.

DD is a binary task. All words labelled with *E (*E denotes BE, IE, EE, SE) are considered as disfluent. The F_1 score [8] of detected disfluent words is used to measure performance.

3. Native Speaker Results

Following previous work [9, 6], the Switchboard Corpus [13] is used for both automatic segmentation (SEG) and disfluency detection (DD) training. Switchboard consists of telephone conversations of native English speakers, annotated with sentence boundaries, part of speech (PoS) tags and disfluency structures. For SEG and DD the corpus is divided into the standard DD train/dev/test sets [3]. For ASR training, the Switchboard ASR training set is used excluding the DD dev and test sets. We removed punctuation and capitalisation from the transcriptions prior to input to SEG as they are not generally generated by ASR systems.

A separate PoS tagger was trained on Switchboard. This tagger adopts a bidirectional LSTM framework[1]. Rule-based predictions were made for filler marks. Other DD features are automatically generated given word tokens and PoS tags. Word embeddings were initialised using Google's 300 dimensional word2vec embeddings [14]. Other input mappings were randomly initialised. All embeddings were updated through network back propagation. The LSTM hidden layer size was set to be 50 and a standard softmax layer was used to make predictions.

3.1. Evaluation on manual transcription

Following previous work [9], the standard Switchboard test set was used and partial words were included for evaluation. Our model achieved an F_1 score of 87.6 on manual transcriptions. This is a 1.7 absolute gain compared to 85.9 obtained in [9]. Preliminary experiments showed this was due to the PoS-based language model features.

Ensemble methods are often used to obtain better predictive performance [15]. We generated an ensemble by creating models for each fold in a 10-fold cross validation. Performance was evaluated against the held out test set, by averaging the predicted probabilities over the 10 models. Ensemble learning boosted the performance to 88.6 on the manual transcriptions. The gains, however, when using ASR transcripts were reduced to only 0.4 so for simplicity ensembles weren't used for the experiments below.

3.2. Evaluation on ASR transcription

In practice, manual transcriptions are not always available and DD must be run on transcriptions produced by an ASR system. Thus, in this work, the model is also evaluated against ASR

[1] https://github.com/marekrei/sequence-labeler

Transcription	Segment	F_1
REF	utt	87.6
	turn	82.0
ASR	utt	75.9
	auto	72.1
	turn	70.8

Table 2: *Disfluency detection performance on Switchboard test set with manual (REF) and ASR transcription using a single sequence labeling model on utterance, turn-level and automatic segmented data.*

transcriptions of the Switchboard test set. The ASR system used has a factorized time-delay neural network (TDNN-F) acoustic model [16], with a 4-gram language model trained on the Fisher Corpus [17] and the Switchboard Corpus. The word error rate (WER) on the DD test set is 15.6%.

To obtain reference disfluency labels for the ASR output, the manual and ASR transcriptions were aligned. Alignment was based on the Damerau-Levenshtein algorithm[2]. We modified the alignment as follows: token transportation was disabled; token deletion cost = 3, insertion cost = 3 and substitution cost = 4+Δ. $\Delta \in [0, 1]$ is calculated using character-level Levenshtein distance to improve token matching. After alignment, disfluency labels were mapped to the ASR transcriptions along with sentence breaks. ASR insertions were excluded when evaluating F_1 score.

By default, ASR transcriptions are separated by conversational turns. Table 2 shows that the F_1 score is reduced from 87.6 to 70.8 on the manual utterance-level and ASR turn-level transcriptions, respectively. When evaluated against ideal utterance-level ASR transcriptions, F_1 improved to 75.9. Effective sentence-like segmentation is therefore needed to improve performance on ASR transcriptions.

By combining the SEG module with an ASR system, it is possible to run DD in a fully automated manner: an ASR system produces transcriptions from audio files; transcriptions are automatically segmented using both prosodic and lexical information; disfluency regions can then be predicted on utterance-level ASR transcriptions. Our system yields a final F_1 score of 72.1 on auto-segmented ASR transcriptions of the Switchboard test set, better than 70.8 F_1 achieved on turn-based segmentation (Table 2).

4. Spoken Grammatical Error Systems

The previous sections have described the development of sentence-like segmentation and disfluency detection for native speakers of English. However, the main aim of this work to examine how these approaches can be used to improve the performance of feedback systems for learners of English. One form of useful feedback to learners is when they are making grammatical errors. In this section, grammatical error detection (GED) is described, and the form of the system used to address this task presented.

Grammatical error detection (GED) systems aim to label each word as either grammatically correct or incorrect e.g.

the cat **seated** on **mat**
c c i c i

The GED system [18] used for this work is a strong deep learning-based bidirectional LSTM framework trained on the written Cambridge Learner Corpus (CLC) [19]. GED uses a sequence labelling model [20]. For an input word sequence, a reference label (1:incorrect; 0:correct) is given to each word. The probability distribution over the two labels is the target to be predicted. The training objective function is the log-likelihood of the predicted label summing over all sentences and all words in each sentence.

5. Non-Native Speaker Results

5.1. Corpora

Two learner corpora were used to evaluate the impact of the post-processing of both manual transcribed learner speech, and the output from an ASR system, on GED. The first corpus is publicly available the NICT Japanese Learner English (NICT-JLE), but does not have any audio available. The second corpus, BULATS, is made available by Cambridge Assessment English and enables the evaluation of the complete pipeline including ASR.

The NICT-JLE Corpus [21] is a set of interview tests conducted with 167 Japanese learners of English, from grades A1-B2 on the CEFR scale [22]. The corpus is annotated with manual transcriptions and associated meta-data including grammatical errors. Tokens marked with a repetition (R) or self-correct (SC) meta-data tag are mapped to the BIO disfluency tags. NICT-JLE does not provide audio data, thus analyses are restricted to manual transcriptions.

BULATS [23] is a free speaking business English test consisting of prompted responses of up to 1 minute. It consists of 225 English learners from 6 L1s and equally distributed across all CEFR grades. The data is carefully transcribed and annotated with grammatical errors and meta-data [24], from which disfluency regions can be derived. The reparandum parts of repetition (RE) and false start (FS) tagged tokens are annotated as disfluencies.

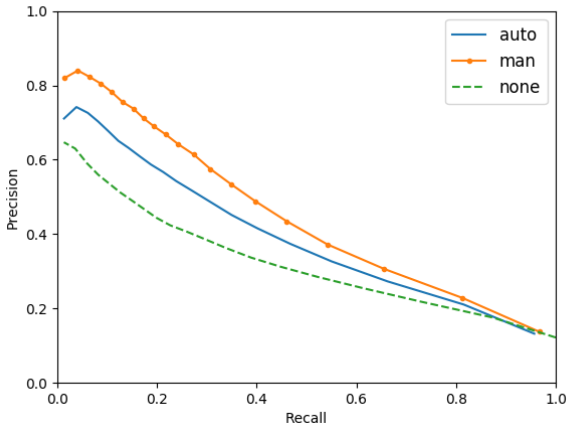

Figure 2: *Precision-recall curves of GED on NICT-JLE manual transcriptions.*

[2] https://github.com/chrisjbryant

Corpus	Trans.	SEG	DD (F_1)	GED ($F_{0.5}$)
NICT-JLE	REF	—	none	36.5
		—	auto (79.8)	**43.7**
		—	man	49.2
BULATS	REF	—	none	38.3
		—	auto (64.0)	**41.4**
		—	man	42.0
	ASR	none	none	23.7
		auto	auto (42.6)	**24.5**
		man	auto (44.6)	24.1
		man	man	24.4

Table 3: *Disfluency detection (DD) & Grammatical error detection (GED) performance on non-native data.*

5.2. Results

GED was performed with three different pre-processing approaches: original transcriptions (none); transcriptions with automatic disfluency detection and removal (auto); transcriptions with manual disfluency removal (man). To offset the impact of false positives in automatic DD, the denominator for the recall rates was adjusted to be the total number of grammatical errors before disfluency removal. Automatic segmentation is also applied to ASR transcriptions.

Table 3 shows that DD on NICT-JLE manual transcriptions scored at 79.8. This is a drop compared to the native speaker Switchboard F_1 score (87.6). This is expected as non-native disfluency patterns vary from native speech [7]. The NICT-JLE corpora has a grammatical error rate (GER) of 12.2%. Table 3 shows that GED $F_{0.5}$ gained 12.7 by manually removing disfluencies, and running automatic disfluency removal achieved a 7.2 absolute gain. The precision-recall curve in Figure 2 shows a clear performance improvement introduced by both auto and manual disfluency removal.

The NICT-JLE corpus does not provide audio information. To extend the investigation to transcriptions generated by an ASR system, the BULATS corpus was used. ASR transcriptions were produced using a joint stacked hybrid DNN and LSTM system [25] with an overall WER of 25.6%.

Table 3 compares DD and GED performance on manual and ASR transcriptions of the BULATS corpus. In general, performance on BULATS is worse than on NICT-JLE. This is due to more disfluent speech (long prompt-response vs short conversational turns) and greater L1 variety, which lead to a higher GER of 15.6%[3]. It can be seen that DD is significantly disrupted by ASR errors, resulting in a drop from 64.0 to 42.6 F_1 on manual and ASR transcriptions, respectively. GED performance on ASR transcriptions shows a slightly smaller, but similar, drop. Removing disfluencies helps to improve GED $F_{0.5}$ performance from the baseline of 38.3 to 41.4 on manual transcriptions; while on ASR transcriptions, automatic SEG and DD helps to improve GED $F_{0.5}$ from 23.7 to 24.5. It is worth noting that GED run on automatic SEG and DD outperformed that using manual SEG and DD, this is because automatic segmentation tends to truncate sentences into smaller segments, which yielded better GED performance.

[3]Only one BULATS annotation is available at present affecting annotation quality, compared to NICT-JLE.

6. Conclusions

Feedback is an essential part of spoken CALL systems. This paper examines one particular form of feedback, spoken grammatical errors, effectively identifying where the learner is constructing a response in a form that a native speaker would not use. Developing these systems is hindered by a lack of annotated data. This paper examines addressing this problem by converting the output from an ASR system into a form which is similar to written text. This enables the use of annotated written text data for detecting spoken errors. Disfluency detection and speech segmentation systems are described based on native speaker data. These systems are then shown to improve the performance of spoken grammatical error detection system on speech data from learners of English.

For future work, we will apply the feedback framework to other downstream tasks such as grammatical error correction; we will explore more advanced neural network models to improve cross-domain application. We also seek to improve the performance on ASR transcriptions by combining ASR confidence scores in DD training.

7. References

[1] E. E. Shriberg, "Preliminaries to a theory of speech disfluencies," Ph.D. dissertation, Uni. of California at Berkeley, 1994. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.1977&rep=rep1&type=pdf

[2] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. P. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1526–1540, 2006.

[3] M. Johnson and E. Charniak, "A TAG-based noisy-channel model of speech repairs," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004, pp. 33–39.

[4] M. Ostendorf and S. Hahn, "A sequential repetition model for improved disfluency detection," in *Proc. INTERSPEECH 2013*, 2013, pp. 2624–2628.

[5] M. Honnibal and M. Johnson, "Joint incremental disfluency detection and dependency parsing," *Transactions of the Association for Computational Linguistics*, vol. 2, no. 1, pp. 131–142, 2014.

[6] P. J. Lou and M. Johnson, "Disfluency detection using a noisy channel model and a deep neural language model," in *Proc. 55th Annual Meeting of the Association for Computational Linguistics, ACL Volume 2: Short Papers*, 2017, pp. 547–553.

[7] R. Moore, A. Caines, C. Graham, and P. Buttery, "Incremental dependency parsing and disfluency detection in spoken learner English," in *Proc. Text, Speech, and Dialogue - 18th International Conference, TSD*, 2015, pp. 470–479.

[8] X. Qian and Y. Liu, "Disfluency detection using multi-step stacked learning," in *Proc. Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2013, pp. 820–825.

[9] V. Zayats, M. Ostendorf, and H. Hajishirzi, "Disfluency detection using a bidirectional LSTM," in *Proc. INTERSPEECH 2016*, 2016, pp. 2523–2527.

[10] S. Strassel, "Simple metadata annotation specification, version 5.0," Linguistic Data Consortium, 2003. [Online]. Available: https://catalog.ldc.upenn.edu/docs/LDC2004T12/SimpleMDE_V5.0.pdf

[11] E. Shriberg, A. Stolcke, D. Z. Hakkani-Tür, and G. Tür, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Communication*, vol. 32, pp. 127–154, 2000.

[12] M. Rei, G. K. O. Crichton, and S. Pyysalo, "Attending to characters in neural sequence labeling models," in *Proc. COLING 2016, 26th International Conference on Computational Linguistics*, 2016, pp. 309–318.

[13] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, 1992, pp. 517–520.

[14] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR Workshop*, 2013.

[15] C. Bucila, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 535–541.

[16] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohamadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Proceedings of INTERSPEECH*, 2018, pp. 3743–3747.

[17] C. Cieri, D. Miller, and K. Walker, "The Fisher Corpus: a resource for the next generations of speech-to-text," in *Proc. 4th International Conference on Language Resources and Evaluation (LREC)*, vol. 4, 2004, pp. 69–71.

[18] K. Knill, M. Gales, P. Manakul, and A. Caines, "Automatic grammatical error detection of non-native spoken learner english," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.

[19] D. Nicholls, "The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT," in *Proc. of the Corpus Linguistics 2003 conference; UCREL technical paper number 16.*, 2003. [Online]. Available: http://ucrel.lancs.ac.uk/publications/cl2003/papers/nicholls.pdf

[20] K. Knill, M. Gales, P. Manakul, and A. Caines, "Automatic grammatical error detection of non-native spoken learner English," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.

[21] E. Izumi, K. Uchimoto, and H. Isahara, "The NICT JLE corpus exploiting the language learners' speech database for research and education," *International Journal of The Computer, the Internet and Management*, vol. 12, no. 2, pp. 119–125, May 2004.

[22] Council of Europe, *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press, 2001.

[23] L. Chambers and K. Ingham, "The BULATS online speaking test," *Research Notes*, vol. 43, pp. 21–25, 2011. [Online]. Available: https://www.cambridgeenglish.org/Images/23161-research-notes-43.pdf

[24] A. Caines, D. Nicholls, and P. Buttery, "Annotating errors and disfluencies in transcriptions of speech," University of Cambridge, Computer Laboratory, UK, Tech. Rep. UCAM-CL-TR-915, Dec. 2017. [Online]. Available: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-915.pdf

[25] Y. Wang, J. H. M. Wong, M. J. F. Gales, K. M. Knill, and A. Ragni, "Sequence teacher-student training of acoustic models for automatic free speaking language assessment," *Proc. IEEE Spoken Language Technology Workshop 2018 (SLT)*, pp. 994–1000, 2018.