Incremental Disfluency Detection for Spoken Learner English

Lucy Skidmore

Supervisor: Prof. Roger K. Moore

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy in Computer Science

in the

Department of Computer Science
December 7, 2022

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Lucy Skidmore

Signature: 

Date: 07.12.2022

In loving memory of my dad, Fred Skidmore.

Acknowledgements

Firstly I'd like to thank my Supervisor, Professor Roger K. Moore for all of his support, wisdom and generosity over the past four years. I am particularly grateful to him for continually fostering confidence in my capabilities, whether that be through inviting me along to participate in inspiring workshops, including me on interesting projects or supporting me to follow my interests in my research. The fact that I even considered pursuing a PhD is in large part thanks to Roger's encouragement. So thank you, Roger, you were right, I did it!

I would also like to thank my family and friends for their endless love and support while I have been busy working on this project. Thank you to my mom, late dad and sister for always cheering me on and being a source of joy and calm amidst the stressful environment of a PhD. Finally, thank you to my partner, Landon, for reassuring me at times when I didn't think I could keep going and most importantly being there for me every single day.

Abstract

Dialogue-based computer-assisted language learning (CALL) concerns the application and analysis of automated systems that engage with a language learner through dialogue. Routed in an interactionist perspective of second language acquisition, dialogue-based CALL systems assume the role of a speaking partner, providing learners the opportunity for spontaneous production of their second language. One area of interest for such systems is the implementation of corrective feedback. However, the feedback strategies employed by such systems remain fairly limited. In particular, there are currently no provisions for learners to initiate the correction of their own errors, despite this being the most frequently occurring and most preferred type of error correction in learner speech. To address this gap, this thesis proposes a framework for implementing such functionality, identifying incremental self-initiated self-repair (i.e. disfluency) detection as a key area for research. Taking an interdisciplinary approach to the exploration of this topic, this thesis outlines the steps taken to optimise an incremental disfluency detection model for use with spoken learner English. To begin, a linguistic comparative analysis of native and learner disfluency corpora explored the differences between the disfluency behaviour of native and learner speech, highlighting key features of learner speech not previously explored in disfluency detection model analysis. Following this, in order to identify a suitable baseline model for further experimentation, two state-of-the-art incremental self-repair detection models were trained and tested with a learner speech corpus. An error analysis of the models' outputs found an LSTM model using word embeddings and part-of-speech tags to be the most suitable for learner speech, thanks to its lower number of false positives triggered by learner errors in the corpus. Following this, several adaptations to the model were tested to improve performance. Namely, the inclusion of character embeddings, silence and laughter features, separating edit term detection from disfluency detection, lemmatization and the inclusion of learners' prior proficiency scores led to over an eight percent model improvement over the baseline. Findings from this thesis illustrate how the analysis of language characteristics specific to learner speech can positively inform model adaptation and provide a starting point for further investigation into the implementation of effective corrective feedback strategies in dialogue-based CALL systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Achieving conversational competence is an essential part of learning a language. Through conversational practice, a learner not only acquires skills in speaking and listening but also gains knowledge of social customs associated with the language they are learning. Having the opportunity for conversational practice is a high priority for learners [1], but it has historically been regarded as difficult to teach using automated systems [2]. As a consequence, language learning technologies available today typically lack the functionality required for the effective practice of conversation, focusing instead on the listening and speaking practice of isolated words and phrases [3, 4]. However, thanks to recent technological milestones in spoken dialogue system technologies alongside the global success of voice assistants such as Alexa and Google Assistant, there has been not only a technological shift but also an attitudinal shift towards the adoption of 'voice-first' technologies [5]. With this shift comes new opportunities for learners to engage in conversational practice. Imagine you are planning a trip to France, and you want to brush up on some common phrases to use when ordering at a restaurant. Or perhaps you have learned a new grammatical construct in your Japanese language class and part of your homework is to practice using it in full sentences. In these examples, a spoken dialogue system could take the role of a speaking partner, generating the required scenario to practice as well as providing learning support anytime, anywhere. This PhD thesis is rooted in this idea, with an aim to establish how state-of-the-art speech technology can be applied to further enhance the capabilities of spoken dialogue systems for language learning. In particular, this work explores how incremental disfluency detection models can be adapted for language learning data, in order to create more opportunities for learners to correct their

own errors when practicing with a spoken-dialogue system.

1.1 Thesis Aims

The overarching goal of the thesis was to build an incremental disfluency detection model that is suitably adapted for learner speech. With this goal in mind, the remainder of this thesis explores three main aims and ten research questions:

1. Identify the challenges specific to disfluency detection for learner speech through a comparative analysis of native and learner disfluency corpora.

    - RQ1. How do the corpora differ in terms of general linguistic features?
    - RQ2. How do the corpora differ in terms of disfluency features?
    - RQ3. How might the features identified in RQ1 and RQ2 impact disfluency detection models adapted for L2 speech?

2. Test two state-of-the-art incremental disfluency detection models to identify a suitable baseline model for further experimentation.

    - RQ4. Which model has the highest overall performance for L2 speech?
    - RQ5. What is the impact of the disfluency features discussed in RQ3 on the models' performance?
    - RQ6. What are the remaining challenges to address in an adapted baseline model?

3. Adapt the baseline model to improve its overall performance.

    - RQ7. Based on the findings of RQ6, how can the baseline model be improved?
    - RQ8. What is the impact of the adaptations outlined in RQ7 on overall performance?
    - RQ9. What is the impact of the adaptations outlined in RQ7 on the disfluency features established in RQ3?
    - RQ10. How does the final adapted model compare to models from other researchers?

Through the fulfilment of the above three aims, this thesis provides an empirical grounding of the differences between native and learner disfluency corpora, which is subsequently used as a framework for highlighting challenges, measuring model performance and developing model adaptations. The resulting incremental disfluency detection model adapted for spoken learner English forms the first step toward incrementality in dialogue-based CALL systems.

1.2 Thesis Outline

Taking an interdisciplinary approach to fulfil the above aims of this research project, the remainder of the thesis is split into five chapters:

Chapter Two first outlines the motivation for exploring incremental disfluency detection for spoken learner English. Through an introduction to the field of computer-assisted language learning (CALL) and subsequently dialogue-based CALL, corrective feedback is identified as an area for investigation. The strategies adopted in dialogue-based CALL systems to facilitate corrective feedback are analysed through the lens of 'pedagogical repair', revealing a missed opportunity for current systems to prioritise the self-initiated self-repair of learners. To address this gap, an incremental framework for corrective feedback that facilitates self-initiated self-repair is proposed, from which incremental disfluency detection is identified as a promising avenue for further research. Exploring this topic further, the linguistic structure, features and variation of self-initiated self-repairs (i.e. disfluencies) are outlined, followed by an overview of the approaches taken for their automatic detection. Alongside general trends in the field, the potential challenges pertinent to the task of incremental disfluency detection for learner speech are discussed, namely the impact of disfluency length, type and learner proficiency on detection scores as well as the absence of models trained using learner data.

Chapter Three further explores the challenges outlined in Chapter Two by means of a linguistic comparative analysis between two disfluency corpora: one for native speakers of English and one for learners of English. Several points of difference between the corpora are analysed, not only untangling the complex relationship between the features discussed above but also highlighting further features of interest: nested disfluencies, learner errors and disfluencies that co-occur with edit terms. In solidifying the differences between native and learner disfluency corpora, the results of this analysis provided motivation to train and test

disfluency models using learner speech.

Chapter Four details the evaluation of two state-of-the-art incremental disfluency detection models for use with learner speech. The models are trained and tested on both native and learner data and evaluated by their performance on the five 'difficult-to-detect' disfluency features established in Chapters Two and Three: length, type, nested, with-edit and with-error disfluencies. Due to its overall higher performance on learner speech as well as its lower number of false positive classifications of learner errors as disfluencies, a deep learning neural network model was selected as a baseline model for further experimentation.

In Chapter Five, five approaches to adaptation of the baseline model established in Chapter Four are explored: dataset lemmatization, using paralinguistic features as inputs, experimenting with character and word embeddings, separating edit terms from disfluency detection and using learners' prior proficiency scores as inputs to the model. Dataset lemmatization and edit term removal were shown to have the biggest impact on detection, with the final adapted model showing an 8.5% performance improvement on the baseline model and a 54% improvement on prior incremental models for learner speech. Analysis of the results highlights remaining challenges for disfluency detection in learner speech, namely the impact of learner errors and proficiency on model performance.

Chapter Six concludes the thesis, providing an overview of the work carried out and how the findings relate to the wider goal of applying speech technology to enhance the current capabilities of dialogue-based CALL systems. General limitations of the thesis are outlined, from which multiple avenues for further work are identified.

1.3 Thesis Contributions

The work described in this thesis is the first to address incremental disfluency detection of learner speech for use in dialogue-based CALL systems. Alongside the findings described above, the principal contributions of this thesis are:

- A systematic analysis of the corrective feedback strategies used by dialogue-based CALL systems. This analysis extends prior work on the features of dialogue-based CALL systems at large [6], devising and applying a new typological framework to analyse

the corrective feedback strategies used in dialogue-based CALL systems with a spoken modality.

- The establishment of an incremental framework for corrective feedback that prioritises self-initiated self-repair. This framework is based on the findings of the systematic analysis outlined above and provides a novel example of how incremental dialogue processing can be applied to a language learning setting. A summary of both this framework and the analysis above will be submitted for presentation at the Speech and Language Technology in Education (SLaTE 2023) workshop, with a paper titled 'An incremental framework for corrective feedback in dialogue-based CALL' .[1]

- The adaptation of the NICT-JLE Corpus (a corpus of learner speech labelled with disfluencies [7]—see Chapter Three for an overview) for use as training data for machine learning models. The outcome of this adaptation process has provided a new version of the dataset optimised for learner disfluency detection research, with added POS tags, disfluency re-labelling, prosodic feature derivation and data-splitting balanced by learner proficiency level. The adapted corpus and accompanying research paper describing the processes of adaptation titled 'Adapting the NICT-JLE Corpus for Disfluency Detection Models' has been shared online for public use.[2]

- The establishment of a baseline model for incremental disfluency detection of learner speech that is trained and tested using the adapted NICT-JLE corpus provides both a starting point for further work in the field as well as a new framework for fair replication and comparison across approaches in wider research.

- Three new disfluency features suitable for measuring disfluency detection model performance have been determined: nested, with-edit and with-error disfluencies. Model performance of all features can be tested in both non-incremental and incremental settings, with the latter being more pertinent to learner speech data.

- The application of learner proficiency level, lemmatization and laughter features for model improvement have previously not been tested in disfluency detection research.

[1]Funding for this publication has been provided by the University of Sheffield Postgraduate Research Student Publication Scholarship.

[2]https://github.com/lucyskidmore/nict-jle

The impact of the latter two on model performance is explored in a paper titled 'Incremental Disfluency Detection for Spoken Learner English' [8], presented at the 17th Workshop on Innovative Use of NLP for Building Educational Applications, held at the NAACL conference in July 2022.

1.4 Additional Publications

Alongside the contributions and publications outlined above, the following work has also been published during the course of this PhD:

- L. Skidmore and R. K. Moore. Using Alexa for Flashcard-Based Learning. In *Proc. Interspeech 2019*, 1846-1850, Graz, 2019.
- L. Skidmore and A. Gutkin. Does *A Priori* Phonological Knowledge Improve Cross-Lingual Robustness of Phonemic Contrasts? In *Proc. SPECOM 2020*, St. Petersburg, 2020.
- R. K. Moore and L. Skidmore. On the Use/Misuse of the Term 'Phoneme'. In *Proc. Interspeech 2019*, 2340-2344, Graz, 2019.

The first publication was presented as a poster at the Interspeech 2019 conference, summarising the researcher's MSc dissertation project which evaluated the feasibility of using Alexa as a tool for flashcard-based Japanese vocabulary learning. The second publication relates to the work undertaken as part of a three month internship as a 'Computational Linguist' at the Google AI Research Lab in London. The publication was co-written with Alexander Gutkin (part of the research team at Google) and reports the findings of the experimentation run by the researcher during the internship, presented online at SPECOM 2020. The final publication was also presented as a poster at the Interspeech 2019 conference, the work was conducted with Roger K. Moore and the researcher contributed to this work through the collation and data analysis of the use of the term 'phoneme' in archived Interspeech publications. Additionally, the researcher was responsible for the editorial coordination of the Dagstuhl Seminar 20021, titled 'Spoken Language Interaction with Virtual Agents and Robots (SLIVAR): Towards Effective and Ethical Interaction (Dagstuhl Seminar 20021)' [9].

Chapter 2

Toward Incrementality in Dialogue-based CALL

This chapter begins with an overview of CALL, after which it narrows focus to the subfield of dialogue-based CALL. Based on existing research in the field, corrective feedback in dialogue-based CALL systems is selected as an area worthy of further investigation. A typology of corrective feedback strategies applied in classrooms is presented and subsequently used as a framework to evaluate how corrective feedback in current dialogue-based CALL systems is implemented. From this analysis, the lack of opportunities for learners to correct themselves in current systems is identified, and a framework that accommodates such functionality is proposed. From this framework, incremental self-initiated self-repair detection is chosen as a necessary and currently unexplored component for dialogue-based CALL systems. With this motivation established, the formal definition of self-initiated self-repair, detailing its structure, features and variation is outlined. This is followed by a detailed overview of work on automatic self-initiated self-repair (also known as disfluency) detection. The main approaches to disfluency detection are explored, alongside a summary of approaches to dataset labelling, evaluation and challenges in the field. Approaches for both incremental and learner speech are analysed in detail, highlighting areas of particular interest for this research.

2.1 Computer-Assisted Language Learning

Computer-assisted language learning (CALL) is defined as "*the search for and study of applications of the computer in language teaching and learning*", [10] p. 1. Established in the 1960s, it is a long-standing field that continuously diversifies in tandem with the evolution of both technology and second language pedagogy. Examples of CALL technology range from using incidental tools such as machine translation engines [11] to intentional tools such as specifically designed language learning apps and software [3]. Compared to face-to-face teaching alone, the additional use of CALL technology not only improves learners' language proficiency [12] but has also been shown to increase motivation to learn [13, 14], learner self-confidence [14, 15] and satisfaction [16]. The interdisciplinary range of applications that CALL encompasses further fuels its diversification resulting in multiple perspectives from which CALL technology can be implemented and evaluated [17]. Human-computer interaction (HCI), second language acquisition (SLA), automatic speech recognition (ASR), natural language processing (NLP), and applied linguistics are just some of the many research areas that CALL intersects.

2.1.1 CALL in Context

From the first CALL programs performing text-based vocabulary drills [18] to contemporary studies in immersion learning using virtual reality (VR) technology [19], there have been a number of significant pedagogical and technological shifts within CALL research which can be broadly categorised into phases. This is visualised by Figure 2.4, which depicts a timeline from 1950 to the present day that marks out the boundaries of each CALL phase according to Warschauer [20], Bax [21] and Gimeno-Sanze [22]. It also marks commercial technology releases influential to CALL as well as example research projects (described below).

The first phase is known as 'Structural'[1] [20] or 'Restricted' [21] CALL, occurring between the 1960s and 1980s. The pedagogical motivation for this approach is rooted in Structural Linguistics, in which language is seen as a formal system made up of distinct units [24]. Mainframe computers were used in specialised language laboratories for vocabulary drill ac-

[1]This term was updated by Warschauer and Healey from their initial labelling of Behavioural CALL six years prior [23].

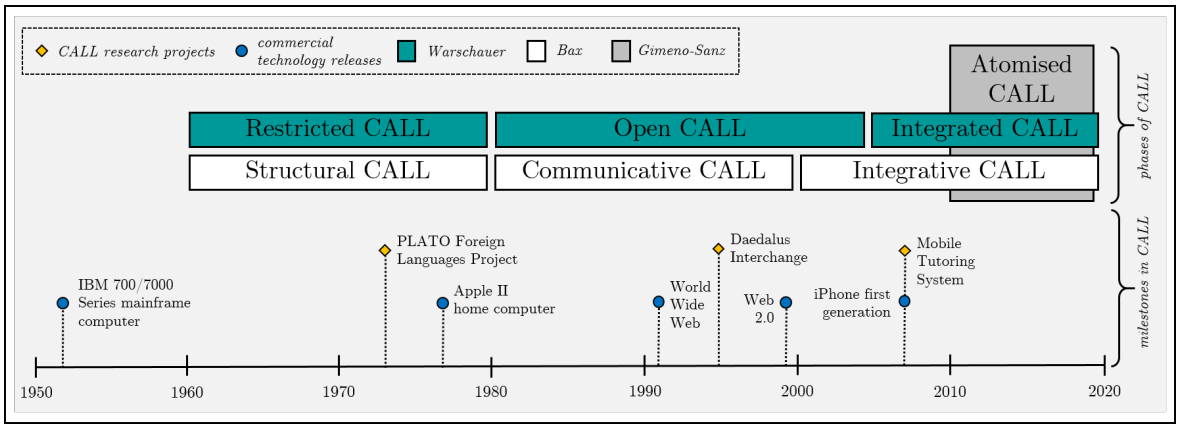


Figure 2.1: *History of CALL showing phases of CALL research according to Warschauer, Bax and Gimeno-Sanze, alongside a timeline of commercial technological releases and research projects related to CALL between 1950 and the present day.*

tivities to improve learners' *accuracy* [20] and were regarded as a supplement to classroom instruction, rather than its replacement [25]. The Illinois PLATO Foreign Languages Project [18] is an example of the Structural/Restricted approach using drill and practise exercises [10]. The project was an evolution of the 'Programmed Logic for Automatic Teaching Operations' computer-based education system developed in the 1960s on ILLIAC, the University of Illinois' mainframe computer [26].

The second phase is deemed 'Communicative' [20] or 'Open' [21] CALL. The former took place through the 1980s and 1990s, influenced by the approach of Communicative Language Teaching (CLT), whereby communicative competence is the main goal of language study [27]. The latter is argued to have continued through to the early 2000s, primarily due to the fact that communicative language teaching forms a part of, but not the whole picture of CALL between the 1980s and 2000s [21]. The term 'Open' is used to reflect the departure from the restricted nature of previous approaches and a more open attitude to the way computers are used for learning. Both interpretations of this phase describe opportunities for interaction, using personal computers (PCs) and later the internet for games, simulations and computer-mediated communication in order to improve learners' *fluency* [20]. An example of this approach can be seen in the investigation of a system called 'Daedalus Interchange', a network computer application that allowed students to engage synchronously in collaborative, written discussion on a chosen topic [28].

Phase three, starting in the 2000s, is deemed as 'Integrative' [20] or 'Integrated' [21] CALL. The former draws on a socio-cognitive view of language learning where learners are encouraged to integrate within a community. Multimedia tools and the internet are used by learners to carry out real-life tasks in order to improve their *agency* in the target language [20]. Integrated CALL, coined in 2003, is framed as "*an aim towards which we should be working*" [21], p.22, where the computer is so embedded into teaching and learning that it is almost an invisible technology—this process is coined 'normalisation' [21]. Phase three captures the ubiquity of computer technology in recent decades, a reality which was in part facilitated by the emergence of mobile-assisted language learning (MALL), a subfield of CALL which explores the use of mobile technologies such as mobile phones, media players, smartphones and tablet computers [29]. Taking CALL outside of the classroom, the portability, versatility and cost of these technologies, in addition to their ability to support multimedia and collaborative activities [30] allowed learners to engage with their language study anytime, anywhere. One example of MALL technology is seen in the investigation by Stockwell [31], who describes a mobile-based intelligent tutor system that generates vocabulary practise tasks personalised to learners' knowledge level.

Technology has continued to advance since the above phases were considered in the early 2000s. The widespread adoption and technological capabilities of smartphones have acted as a driver for 'integration' in CALL. With them came an abundance of commercial language learning applications ('apps'), a multi-billion dollar industry in its own right, as well as a rich topic of study. Smartphones not only enable the use of apps but also provide relatively low-cost access to high-speed internet. This in turn grants a learner with access to a vast array of online learning resources. This impact is directly linked to the current phase of CALL, coined in 2013, known as 'Atomised' CALL [22] which describes the field from 2010 onward. Gimeno-Sanze argues that this phase is an extension of phase three and thanks to the 'normalisation' of technology forecast by Bax [21], language learning has moved away from relying on standalone 'all-in-one' software and has become primarily "*needs-driven*" [22], pg. 1110, making use of the multiple technology-based tools and resources now available to learners. Examples of these tools and resources include language learning apps [1], podcasts [32], computer games [33] and other Web 2.0 technologies [34].

It is worth noting that although introduced chronologically, the phases described above

are not strictly confined to set time periods, and elements of all phases can be found across CALL research and development. The phases instead provide a framework to outline the general transformation of CALL from the 1960s to the present day. CALL systems have evolved from inaccessible tools used in university laboratories that teach small units of language to personalised tools used in everyday life that facilitate unlimited access to language learning resources. This evolution can be in part attributed to the general advancements in computer hardware in terms of portability, processing power and affordability. This is reflected in the commercial releases of the mainframe computer, desktop computer, and smartphone illustrated in Figure 2.4 where each release precedes a new phase of CALL. This evolution has been advanced further by the invention of the World Wide Web and Web 2.0 in particular, providing connectivity between CALL systems, facilitating real-time interaction between learners in a way that was not possible previously. With each technological milestone, the field has diversified and generated subfields, forming a vibrant and interdisciplinary landscape of research and development, reflecting Levy's statement that "*the nature of CALL at any particular time is, to a large degree, a reflection of the level of the development of the technology.*" [10], p.1. One such subfield—and the focus of this thesis—is dialogue-based CALL.

2.1.2 Dialogue-based CALL

As a consequence of the recent advances in automatic speech recognition and natural language processing, there has been huge growth in the commercial development of voice assistants such as Amazon Alexa, Google Assistant and Apple's Siri [5]. The widespread adoption of these technologies on smartphones, smart speakers and even in cars is evidence of a societal shift towards using spoken dialogue systems in everyday life. Echoing the pattern of general technical innovation leading to CALL applications described above, this shift has obvious ramifications for CALL in terms of facilitating conversational practise, a skill that is a high priority for language learners [1] that has been shown to be effectively improved using CALL systems [35]. This area of research forms part of an emerging subfield of CALL, known as 'dialogue-based' CALL.

Dialogue-based CALL concerns the application and analysis of automated systems that engage with a language learner through conversation. Routed in an interactionist perspective

Figure 2.2: *Screenshots from the POMY dialogue-based CALL system. From left to right, the scenarios include asking for directions, visiting a post office and going to the supermarket. Image Copyright ACL 1963–2022, licensed under CC BY-NC-SA 3.0.*

of second language acquisition (SLA) [36], dialogue-based CALL systems afford learners the opportunity for spontaneous production of the language they are practising. In a dialogue between a system and a learner, the system assumes the role of a conversational partner, providing the learner with the opportunity to practise their verbal communication skills at length, any time in an environment of low social risk. Dialogue with an automated agent has been a feature of CALL applications such as intelligent tutoring systems (ICALL) [37] and computer-assisted pronunciation training (CAPT) programs [38] since the 1980s, with additional strands of research emerging from the application of spoken dialogue systems (SDSs) [39] and chatbot technology [40]. Now emerging as a field in its own right, dialogue-based CALL systems today have the opportunity to exploit complex dialogue management and higher accuracy non-native speech processing, in turn facilitating more sophisticated and helpful conversational practise tools for language learners. Evaluations of dialogue-based CALL systems are reflective of the impact of CALL systems more generally—they have been shown to improve proficiency [35] and reduce speaking anxiety [41], as well as being enjoyable to learners [42].

One example of the level of complexity that current systems can achieve is POMY [43, 44], an immersive language learning game where learners interact via speech with characters inside a 3D environment on the screen to carry out tasks such as visiting the post office to send a package. Figure 2.2 shows screenshots of the various tasks available to learners. Designed for Korean-speaking learners of English, the dialogue system is goal-oriented thanks to the task-based games that take place in an immersive environment. For example, in the post office scenario, learners are required to send a camera to England and, through a conversation

with the post office worker, have to ensure that the package is insured and delivered by the following week. During tasks, feedback is given on learners' morphological, grammatical and lexical errors by a 'tutor' character that accompanies learners throughout the game. The tutor can also provide hints to learners on how to engage with a task. Investigations into the effectiveness of POMY have shown that it improves learners' speaking rate and confidence levels [44] as well as satisfaction and interest in learning [45].

Another example of a contemporary dialogue-based CALL system is a text-based interaction game for English-speaking learners of Chinese [46]. Learners interact with pre-recorded videos of characters with the goal of form-focused practise of formulaic expressions. The setting and task are first described to the learner, after which they are required to play their role in the scene by selecting the appropriate expression in tandem with the scene being acted out in the video. Below is an example scenario (with English translations only) taken from [46].

> Situation: You are shopping at Raffles City Shanghai. A shop assistant walks up to you to offer help, but you don't need her help. While browsing, you see a T-shirt you like. You wonder how much it is and if it fits you.
>
> 1. Shop assistant: Hello, how can I help you?
> 2. Customer: I'm just looking
> 3. Shop assistant: Alright, please let me know if you need any help
> 4. Customer: How much is this T-shirt?
> 5. Shop assistant: It's 120 RMB.
> 6. Customer: Can I try it on?
> 7. Shop assistant Sure, the fitting room is over there

For utterances two, four, and six, learners are required to select from a choice of four options, with only one being correct. Feedback is provided to learners through the issuing of points for correct answers, as well as hints when the wrong answer is chosen. Findings from learner testing showed that after using the system, learners had stronger retention of the formulaic expressions both immediately after practise and also two weeks later [46].

The examples above show a snippet of the broad range of pedagogical and technical approaches to developing dialogue-based CALL systems. Differences can be seen in modality (e.g. spoken vs. written), the level of interactivity afforded to the learner (e.g. goal-oriented

and meaning-focused vs. highly constrained and form-focused), as well as instructional features such as the inclusion of feedback and gamification (see Bibauw et al. [47] for an overview of this diversity and how various features impact learning outcomes). With each system comes its own weaknesses specific to the desired learning task and tackling such challenges requires expertise across multiple domains. As it is not possible to explore all areas here, this thesis focuses specifically on how speech technology can be applied to improve the corrective feedback capabilities of dialogue-based CALL systems.

2.2 Challenges for Dialogue-based CALL: Corrective Feedback

Despite the advanced capabilities of contemporary dialogue-based CALL systems described above, an area where systems are still limited is in their ability to provide high-quality corrective feedback to learners. A recent investigation into the interactional, instructional and technological characteristics of dialogue-based CALL systems found that just under half of all systems that were evaluated provide corrective feedback [6], despite the fact that learning outcomes are almost twice as strong for the systems that include such functionality [48]. With these findings comes the motivation to explore both the methodological and functional limitations of dialogue-based CALL systems that employ corrective feedback.

2.2.1 Corrective Feedback in Teacher-Learner Dialogue

In a language learning environment, teacher-learner dialogue differs from dialogue between two native speakers. Firstly, there is a known imbalance of language capabilities between the speakers, with the learner having less knowledge of the target language than the teacher. For learners, this means they are more likely to produce errors in the target language and they may also struggle to comprehend the meaning of the teacher's speech. As a consequence, the teacher is required to accommodate for this knowledge imbalance through pedagogical techniques such as negotiation of meaning, intonation change and adjusting their rate of speech. These techniques fall under the umbrella of *corrective feedback*, a pedagogical approach that sits at the intersection of conversation analysis (CA) and second language acquisition (SLA). From the CA perspective, corrective feedback focuses on the application of the analysis frame-

work of 'repair' (the term given to the handling of errors in dialogue) in a language learning setting [49, 50]. From the SLA perspective, the focus is on how corrective feedback is used in practice in second language classrooms, and whether or not it is beneficial for language improvement. This is achieved through analyses of interactions between teachers and learners that are categorised and tested (see [51, 52, 53] for examples). Insights from this research provide an empirical basis for implementing specific corrective feedback strategies in classroom settings.

Figure 2.3 shows a taxonomy derived for this investigation that details the four instances of error correction in teacher-learner dialogue: 'prompting', 'reformulation', 'self-correction' and 'bridging the gap', a phrase coined specifically for this investigation. It is informed by typologies from both branches of research [49, 51, 54], capturing the dialogic structure of corrective feedback, general descriptions of when the corrections occur in practice as well as methods for implementing them. In addition, the terminology used in CA repair theory to describe a repair segment is adjusted for language learning—the standard terms *self* and *other* used in *self-initiated*, *self-completed*, *other-initiated* and *other-completed* are replaced with *learner* and *teacher*, respectively. In line with a concept referred to as 'pedagogical repair' [49], the framework applies only to errors in production made by the learner and does not address errors in learner comprehension or teacher errors. Additionally, the feedback provided by teachers is presumed to take place shortly after it occurs during conversation.

Self-correction describes the process of learner-initiated learner-completed repair, whereby the learner realises their own error and corrects it themselves. During self-correction, learners often mark their errors prosodically by cutting off the end of the erroneous word, stretching it out, or pausing, as well as lexically through the use of edit terms and error repetition (see later in this chapter for more detail on such processes). The other opportunity for learners to correct their own errors within this framework is through prompting, where the teacher not only indicates to the learner there is a problem in their utterance but also provides an opportunity for the learner to correct the error themselves. There are various methods that teachers use to achieve this: asking the learner to clarify what they mean (clarification request), repeating the same erroneous phrase back to the learner (repetition), coaxing the correction from the learner through the partial utterance of the corrected phrase (elicitation), providing clues related to the linguistic form of the desired correction (meta-linguistic clue)

Figure 2.3: *High-level taxonomy of corrective feedback strategies used in teacher-learner dialogue, with examples of methods for each error correction strategy for the learner-produced sentence "I wented to the cinema".*

and finally through using non-verbal actions such as the raising of an eyebrow (paralinguistic signal).

Bridging the gap and reformulation strategies occur when the teacher is the one to correct the learner's error. Opportunities for bridging the gap arise when a learner is aware that they have made an error but are unable to correct it themselves. This method is commonly referred to as a 'word search', i.e. a scenario when the speaker cannot think of the appropriate word to say. In a language learning setting, this can occur when a learner cannot think of the appropriate correction for an error, such as the example in Figure 2.3 but could also refer to a scenario where a learner cannot think of or does not know the appropriate vocabulary to use. Finally, reformation is used by teachers when the learner has not noticed their own error. Methods for doing so include repeating back a corrected version of the learner's erroneous utterance (recasts), overtly stating that what the learner said was incorrect and directly providing the correction (explicit correction), and finally explicit correction with a meta-linguistic explanation, which is the same as the previous method with the addition of the reason as to why the learner's utterance was incorrect.

A meta-analysis of 15 classroom-based studies that investigated the pedagogical effectiveness of the oral corrective feedback strategies described above showed that prompting approaches outperform reformulation, especially those that elicit free constructed responses from learners (i.e. responses that are not constrained by form or meaning) [51]. In addition, research indicates that learners not only prefer learner-completed repair (self-correction and prompting) over teacher-completed repair (reformulation and bridging the gap) [55] but also find opportunities to self-correct more motivating [56]. These preferences are also reflected in the behaviour of teachers, who have been shown to generally opt for a prompting first, reformulation second approach to feedback [57]. With both the pedagogical and preferential evidence pointing toward learner-completed repair as the more effective approach in teacher-learner dialogue, it seems that it would be beneficial for dialogue-based CALL systems to also prioritise such opportunities. However, in existing typologies of dialogue-based CALL systems [6], this level of granularity in regard to the kind of corrective feedback methods that are used does not exist, and so further investigation is required as to if and how such methods are currently being implemented.

2.2.2 Corrective Feedback in Dialogue-based CALL Systems

In order to better understand corrective feedback employed by dialogue-based CALL systems, the typology developed by Bibauw et al. [6] was expanded for this investigation to include further classifications relating to the strategies described above. The resulting modified typology was used to evaluate the distribution of corrective feedback methods in order to identify gaps in existing systems' capabilities. To carry out this task, the following data collection processes were followed:

1. The search methodology described by Bibauw et al. [6] was replicated to gather all papers related to dialogue-based CALL that have been published since 2017 (the end date of the study).

2. All dialogue-based systems (old and newly retrieved) labelled in the database with a 'primarily spoken modality' were reviewed and categorised according to their corrective feedback type and method, using the taxonomy described above in Figure 2.3.

The additional database of systems with categorical labelling is reported in Table 1 in the Appendix. Out of the 42 dialogue-based CALL systems surveyed with a primarily spoken modality, 27 had corrective feedback as a feature of the system and 26 provided details of the types and methods followed to implement it.[2] Ten of the systems surveyed employed more than one method of feedback and three out of the four feedback types were reported across the systems: prompting, reformulation and bridging the gap. Twelve of the systems gave delayed feedback (provided to learners after the completion of dialogue exercises) and for nine of these systems, this was the only form of feedback provided. The delayed feedback was communicated in various ways: onscreen transcriptions highlighting mispronounced words [38, 59, 60] and grammatical errors [61] as well as pronunciation scores given at word and sentence level [62, 63]. Additionally, pronunciation tips were generated for learners' common errors [64] as well as opportunities to hear back their own speech compared to an ideal pronunciation [65]. There was also use of visual feedback with simulations of mouth movements for learners to imitate [62, 66], and 'prosody transplantation' where the appropriate prosody for a given sentence was imposed onto the learner's speech for them to hear [67].

[2]The dialogue-based system 'Military Language Tutor' [58] used corrective feedback, but did not include any further details, and so was excluded from this analysis.

Looking at the strategies followed by the systems that applied immediate feedback, the most frequent approach was to combine the methodologies of prompting with reformulation or bridging the gap. Commonly these systems first provide the learner with the opportunity to correct themselves by using a clarification request. If the clarification request is unsuccessful, it is followed by a recast [68, 69, 70, 71]. This strategy has also been implemented through the use of repetition as the initial prompting method [43]. In place of recasts, some systems provide opportunities for bridging the gap instead through the generation of hints that learners can opt for following an unsuccessful clarification request [72, 73]. Following a similar approach but only using prompting methods, one system initiated correction with a clarification request, followed by a more explicit form of prompting such as elicitation [71]. The second most frequent correction approach is to apply explicit feedback both with and without metalinguistic information. The former is typically given at word level, realised through scored and corrected transcriptions for both grammar and pronunciation feedback that is generated as the learner speaks [74, 75, 69]. The latter provides feedback on a turn-by-turn basis, often through a 'pop-up' screen detailing metalinguistic information that has been generated based on a set of pre-defined error classifications [76, 77].

The remaining approaches to correction are standalone recasts, clarification requests due to system recognition errors and elicitation. Standalone recasts do not occur in combination with other methods and are applied with varying levels of implicity. These include the embedding of a recast into a system's response [78], providing additional prosodic emphasis to the corrected part of a recast [79] and generating a recast for the system's response whilst highlighting the learner's error in the conversational transcript [65]. Clarification requests that are due to system recognition difficulties differ from those that are combined with other methodologies in that in these cases the system is not able to identify issues in the learner input, either because the input is unintelligible for some reason, or the type of error produced by the learner has not been pre-defined by the system. In these cases, systems respond with phrases such as "*I don't understand, can you repeat that*" [77, 65, 71]. For the systems evaluated, these types of clarification requests are an additional fallback feature rather than the primary mode of correction for a system. Finally, two systems investigated reported elicitation as their only feedback method [80, 81]. In both cases, hints were generated by the systems in order to provide feedback on meaning (to progress the conversational trajectory),

rather than feedback on form.

The overview above has revealed the range of approaches followed by dialogue-based CALL systems to provide immediate oral corrective feedback. With such technology comes multimodal affordances not typically available during teacher-learner dialogue, such as on-screen transcriptions, visual feedback through highlighting and real-time proficiency scoring. Nevertheless, the majority of systems employ feedback strategies similar to those enacted by teachers—starting with implicit methods that encourage learner-completed repair before falling back to explicit forms of teacher-completed repair when necessary [57]. Despite this prioritisation of teacher-initiated learner-completed repair, there are no references to the affordance of learner-initiated learner-completed repair (i.e. self-correction) in any of the systems evaluated. This may in part be due to the nature of self-correction, in that it could be considered to be outside of the purview of corrective feedback, given that it is the learners themselves that are correcting their errors. However, as explored above, self-correction is an inherent part of the error repair process in conversation. It is the most frequently occurring correction type in speech [82] as well as being the most preferred type by learners [49, 83]. Through not addressing self-correction as part of the feedback process explicitly, dialogue-based CALL systems not only miss an opportunity to prioritise an additional form of learner-completed repair but also run the risk of self-corrections being misinterpreted as learner errors[3].

[3]Here an error is defined as the inaccurate use of language by a learner. An error may be related to a learner's pronunciation (phonological), word formation (morphological), word use (lexical) or sentence structure (grammatical). See [84] for an overview of errors in language learning.

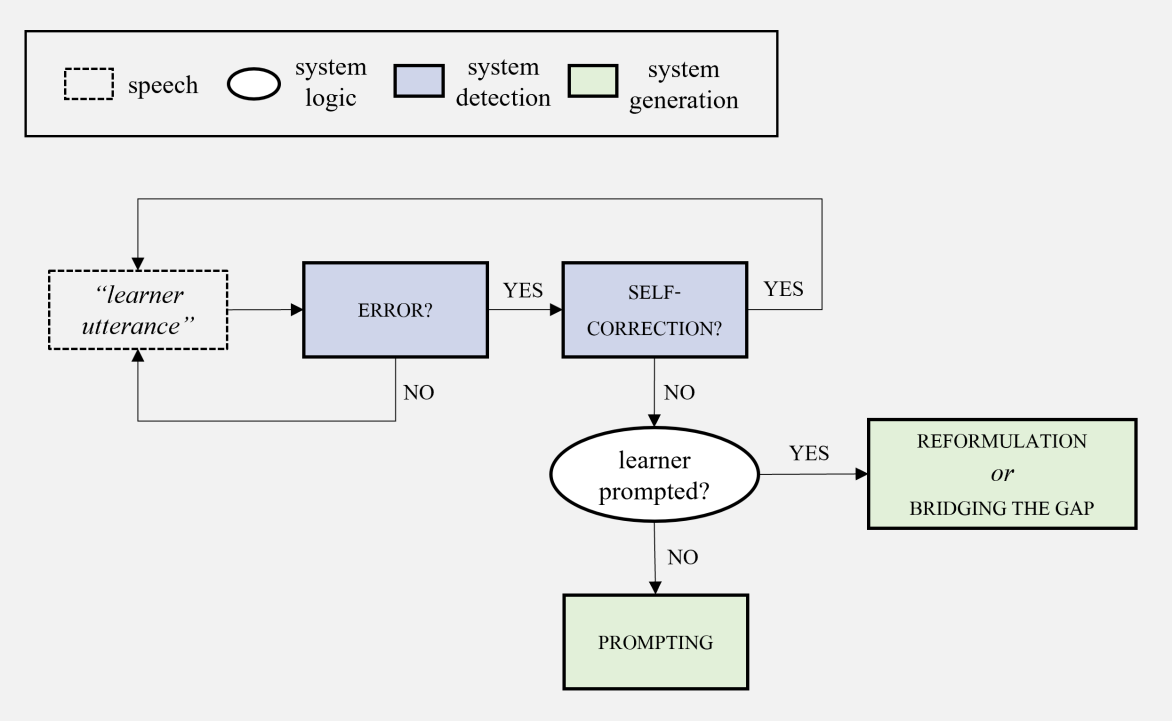


Figure 2.4: *Proposed incremental process for corrective feedback in a dialogue-based CALL system.*

2.2.3 An Incremental Framework for Corrective Feedback

Figure 2.4 shows a proposed framework for enacting corrective feedback in a dialogue-based CALL system. It has a similar structure to the majority of systems described above whereby learner-completed repair is prioritised over system-initiated repair, however, it differs in that it also provides the affordance of self-correction. The framework is designed to process a learner's utterance *incrementally*, i.e. word-by-word, in order to detect and act upon errors as they occur, rather than at the end of a given utterance. As the learner speaks, the system continuously detects whether or not an error has occurred in their utterance. If there is no error, the system continues to process the learner's speech. If an error is detected, the system first waits to see if the learner goes on to correct it (self-correction). If the learner corrects themselves, the system continues to process the learner's utterance as before. If the learner does not correct their error, the system goes on to do one of two things. If this is the first attempt by the system to correct the learner, it generates a prompt, again affording the

learner an opportunity to correct themselves (prompting). If after prompting the learner is unable to correct their error, the system does so through reformulation or bridging the gap methods.

The framework introduced above does not encapsulate all of the functionality required to implement a complete dialogue-based CALL system but instead focuses specifically on the key processes of pedagogical repair: error detection, self-correction detection and feedback generation. The feedback elements of the proposed framework would require either rule-based or model-trained generation of utterances primed on the error detection output as well as the pedagogical goals of the dialogue task itself. The addition of a self-correction module is what separates this framework from the previous approaches described above. Not only does the inclusion of the module force the system to 'wait', creating a window of opportunity for the learner to self-correct, but its output also determines the most effective strategy of corrective feedback to generate.

In a real-world setting, there are several other system elements beyond those described above that would impact the performance of such a dialogue-based CALL application. Firstly, poor quality ASR output may lead to the incorrect labelling of both learner errors and self-correction. Without accurate labelling of these speech features, the system would not be able to provide appropriate corrective feedback, potentially leading to confusion for the learner. This is more likely in settings of low-resource data such as learner speech where the rate of errors and mispronunciation is typically high, impacting the accuracy of ASR output. Additionally, the requirement for incrementality across the system as a whole, including the framework above as well as ASR, a dialogue manager and text-to-speech (TTS) modules may cause system latency. Such latency may impede the system's ability to provide corrective feedback to learners at the appropriate points in the conversation, again leading to ineffective and potentially confusing feedback for the learner. With this in mind, the framework introduced above can be considered as a starting point for the wider goal of fully effective incremental processing in dialogue-based CALL systems.

2.3 Automatic Disfluency Detection

Disfluency detection concerns the automatic labelling of self-initiated self-repairs (i.e. disfluencies) found in spontaneous speech. The majority of research in this area is applied with the aim of disfluency removal, transforming transcribed speech into a form more similar to written text. There are four main approaches to disfluency detection: parsing-based approaches, noisy-channel model approaches, sequence labelling approaches and encoder-decoder approaches. Following an introduction to the linguistic features of disfluencies, each approach to disfluency detection is addressed in detail below, alongside an overview of pertinent input features, evaluation metrics and open challenges.

2.3.1 Self-initiated Self-repair

As introduced in Section 2.2.1 above, 'self-initiated self-repair' is a term coined in conversation analysis research [82] describing the process whereby through 'self-monitoring', a speaker notices and 'repairs' a problem in their speech [85]. Such problems are a result of trouble during the psycholinguistic processes of utterance planning and production, broadly categorised as problems with planning informational structure and content, appropriateness of terms used and formulation of erroneous utterances [85]. The repair process can occur immediately within the same utterance (position one self-initiated self-repair) or in the utterance following an interlocutor's utterance (position three self-initiated self-repair) [86]. In line with the framework proposed in Section 2.2.3, this overview focuses on position one self-initiated self-repairs. Such repairs are preferred by speakers [82] and the most frequently occurring in conversation [87], making up approximately 6% of conversational speech [88].

(1) *I'd like a* [$\underbrace{\textit{coffee}}_{\text{reparandum}}$ + $\underbrace{\{\textit{uh}\}}_{\text{interregnum}}$ $\underbrace{\textit{cup of tea}}_{\text{repair}}$] *please*

Following the notation scheme devised by Shriberg [89], the example above shows the components of a self-initiated self-repair, where the speaker changes their request of "*coffee*" for a "*cup of tea*". Systematic in their structure, such repairs comprise of a reparandum phrase, optional interregnum phrase and repair phrase [85]. The + represents the 'interruption point', marking the end of a reparandum phrase prior to the onset of repair. Interregnum

phrases are either filled pauses, edit terms or discourse markers such as "*uh*", "*I mean*" and "*you know*". Beyond the lexical structure of the reparandum phrase, interregnum and repair phrase, prosodic features also play an important role in self-initiated self-repair. The interruption point is often marked prosodically with features such as silence as well as reparandum word cut-off and glottalisation [90]. Pitch differences between repair and reparandum phrases have also been reported, with emphasis particularly given to self-repairs that are initiated due to grammatically erroneous reparandum phrases [91]. Finally, physical acts of eye and gestural movements have been shown to have a systematic function within self-repair, correlating with preferences for self-initiated self-repair during word searches [92, 93].

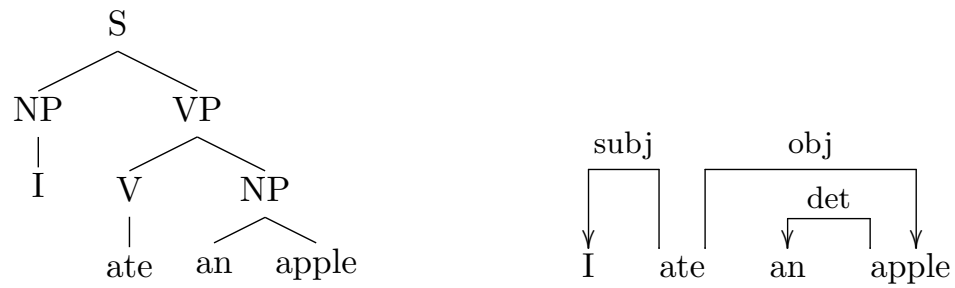
There are three structural sub-types of self-initiated self-repairs: repetitions, substitutions and deletions [89]. Example (2) below shows a repetition-type repair, where the reparandum phrase and repair phrase are identical. Example (1) above is a substitution-type repair, where the reparandum and repair phrases are different lexically but are easily understood to be substitutions of one another. Deletion-type repairs occur when the repair phrase does not resemble the reparandum phrase. Utterance-initial deletions such as the one shown in example (3) are referred to as restarts.

(2) *I'd like a* [$\underbrace{\textit{cup of}}_{\text{reparandum}}$ + $\underbrace{\textit{cup of}}_{\text{repair}}$] *tea please*

(3) [$\underbrace{\textit{Where}}_{\text{reparandum}}$ + $\underbrace{\{\textit{erm}\}}_{\text{interregnum}}$] *I'd like a cup of tea please*

In addition, self-initiated self-repairs can be simple or complex [89]. Simple repairs refer to the examples seen above, where there is only one instance of reparandum and repair within the overall repair structure. Complex repairs have multiple instances within one structure, also known as nested repairs. See example (4) below, where the repetition of the word "*like*" occurs within the reparandum phrase of the outer repair structure.

(4) [$\underbrace{\textit{I'd}\ [\ \overbrace{\textit{like}}^{\text{reparandum}}\ +\ \overbrace{\textit{like}}^{\text{repair}}\ ]}_{\text{reparandum}}$ + $\underbrace{\{\textit{erm}\}}_{\text{interregnum}}$ $\underbrace{\textit{I'd like}}_{\text{repair}}$] *a cup of tea please*



(a) *constituency parsing* **(b)** *dependency parsing*

Figure 2.5: *The labelling approaches for constituency and dependency parsing using the example sentence "I ate an apple".*

The frequency, distribution and linguistic features of self-initiated self-repairs have been shown to be dependent on a multitude of factors. Firstly, there is a proven relationship between the frequency of self-initiated self-repair and cognitive load [94], with less predictable words more likely to trigger instances of repair [95, 96]. Other examples of influencing factors include but are not limited to speech task type [97, 87, 98], speaker age and gender [99, 100], speech modality (i.e. human-computer interaction vs. human-human interaction) [101, 102] and the native language of the speaker [103]. The same is true for learner speech, which has higher overall rates of self-initiated self-repairs [104, 105] and filled pauses [106, 107] compared to native speech.

2.3.2 Approaches to Disfluency Detection

Text Parsing

Text parsing is the process of labelling sentences for their syntactic structure, and is applied for tasks such as grammar checking and question-answering [108]. There are two forms of parse-tree structures that can be applied to a given sentence: constituency structures and dependency structures. The former groups words into sub-phrases (i.e. 'constituencies') according to their grammatical function, whereas the latter labels the directional syntactic relationships (i.e. dependencies) between words. Figure 2.5 shows how constituency and dependency labelling differ for the example sentence "*I ate an apple*".

The presence of disfluencies in speech is a challenge for standard text-parsers. This is due to the fact that despite being syntactically similar, the reparandum and repair phrase of disflu-

encies do not always form a whole syntactic phrase. With this challenge came the motivation to eliminate disfluencies as part of the parsing process. The earliest work in this area integrated hand-crafted rules in order to delete reparandum phrases immediately upon detection during parsing [109, 110]. Other work identified disfluencies either prior to [88, 111, 112] or after [113] the parsing process. More recent iterations of this approach apply transition-based dependency parsing, where the standard transition operators used to label dependencies are augmented to include new operators for disfluency labelling [114, 115, 116, 117]. The most successful results, however, are seen in self-attentive neural constituency parsers for joint detection and constituency parsing. Modelled as a multi-task learning problem, syntactic information from parse-trees improves the accuracy of neural disfluency detection models achieving close to current state-of-the-art performance [118, 119].

The main limitation of parsing-based approaches is in the requirements for data labelled with both syntactic information and disfluencies. For the majority of work that takes such an approach, this is not a problem due to the fact that the most common corpus used for analysis, the Switchboard corpus (introduced in detail in Chapter Three), contains such labelling. However, when adapting disfluency detection models to new domains such as learner speech, there is no such dataset publicly available.

Noisy-channel Model

The noisy-channel approach to disfluency detection is adopted from statistical machine translation research. As described by Johnson and Charniak [120], here the original text containing disfluencies is considered the 'source language' and the text with disfluencies removed is considered the 'target language'. With the idea that fluent speech is passed through a 'noisy' channel to create disfluent speech, given this observed disfluent string Y, the aim is to uncover the underlying fluent string $\hat{X}$, expressed using Bayes Theorem as follows:

$$\hat{X} = \underset{X}{\operatorname{argmax}} \, \mathrm{P}(X|Y) = \underset{X}{\operatorname{argmax}} \, \mathrm{P}(Y|X) \cdot \mathrm{P}(X)$$

The resulting probability distribution is split into two parts: $\mathrm{P}(X)$ denotes the language model probability of a given fluent sentence X and $\mathrm{P}(Y|X)$ denotes the probability that the noisy-channel model generates disfluent sentence Y given fluent sentence X. Modelling in

such a way isolates the 'noise' (i.e. reparandum phrase) from the fluent sentence.

Model variations for this approach include combining multiple noisy-channel model outputs by a weighted sum [121], using a syntactic parser-based language model along with a Tree Adjoining Grammar (TAG) noisy-channel model [122, 120], training language models on large amounts of non-speech data [123] and adding context-sensitive conditional probabilities to the noisy-channel model [124]. The most successful outcome from these approaches is by using deep neural language models combined with hand-crafted features to re-rank the noisy-channel output [125].

The noisy-channel model approach relies on the assumption that the repair phrase is a 'rough copy' of the reparandum phrase, sharing structural and lexical similarities. However, as discussed by Zayats et al. [126], this framework is not as successful when detecting restart-type disfluencies, where the reparandum phrase is abandoned altogether. In addition, the most successful approaches that leverage additional language models create complex runtime dependencies [127] causing a higher computational load which is not preferable in an incremental setting where efficient time-to-detection is a functional priority. When applying the task to learner speech, it may additionally be the case that additional noise in the data such as grammatical errors is falsely identified as part of a reparandum phrase. This has not yet been shown however and requires further investigation.

Sequence Labelling

Born out of the success of statistical models for automatic speech and language processing seen through the 1980s and 1990s, sequence labelling is the most popular approach for disfluency detection research. Each word in a given sentence is labelled as either fluent or disfluent, where only the reparandum phrase is considered disfluent as shown in the example below.

Input sentence:	*I'd*	*like*	*a*	[*cup*	*of*	*tea*	{*uh*}	+	*coffee*]	*please*
General tags:	F	F	F	D	D	D	F		F	F
BIO tags:	O	O	O	BE	IE	EE	O		O	O

The example shows two common approaches to sequence labelling for disfluency detection: general disfluency tags and 'beginning-inside-outside' (BIO) tags [128]. The former is a

binary labelling approach which tags each word in a reparandum phrase of any length simply as *disfluent* (D) and the remaining words, including edit terms, as *fluent* (F). The latter approach delineates the position of words in a reparandum phrase as either at the *beginning* of a reparandum phrase (BE for 'begin edit'), *inside* the reparandum phrase (IE) or at the *end* of a reparandum phrase (EE). Single word reparandum phrases are given a separate label of *single edit* (SE) and the remaining words are labelled as *other* (O).

Early iterations of the sequence labelling approach combined hand-crafted rules with statistical models such as Hidden Markov Models (HMMs) and Decision Trees [129, 130, 131]. Other work used statistical approaches alone such as probabilistic language models to predict disfluent words from context [132] and Decision Trees using prosodic information for detection [133, 134]. Joint modelling was also a popular approach at this time, combining the task of disfluency detection with end-of-turn detection [135], sentence boundary detection [136] and punctuation detection [137] for use in spoken dialogue systems. As approaches to statistical modelling continued to improve, so did their application to disfluency detection, with work showing the success of conditional modelling through the use of Maximum Entropy Models (MEMs) and Conditional Random Fields (CRFs) [128, 138, 139, 140]. The latter were the most successful, and hence were the primary approach to disfluency detection for almost a decade until the mid-2010s [141, 142, 143, 144, 145]. Echoing the approaches outlined above, the current best performing sequence-labelling model makes use of neural networks, specifically a bi-directional Long Short-Term Memory network (LSTM) with word and part-of-speech (POS) embeddings as well as hand-crafted features as model inputs [126].

The advantage of bi-directional LSTMs as opposed to regular LSTMs or other neural networks is that they process both the backwards and forward context of any given input. This is particularly useful for disfluency detection, as the forward context of the repair phrase can often provide useful information for accurately labelling the reparandum phrase due to the syntactic relationship between the two. However, similarly to other recurrent neural network (RNN) structures, LSTMs struggle with long-range dependencies. This is due to the fact that during the backwards pass of model training, the multiplications required to backpropagate the error signal through the model's hidden layers can cause the resulting gradients to be driven to zero when sequences are long [108]. This is often referred to as the 'vanishing gradient problem' and in the case of disfluency detection, it results in poorer performance

for disfluencies with longer reparandum phrases. In addition, the best performing sequence labelling models rely on numerous hand-crafted features causing high dimensional sparse data which is undesirable for model training. Finally, detecting disfluencies through individual word labelling requires that each word be modelled independently. Although some context is captured through RNN model structures, such models can cause illegal label sequences. This in turn requires further elements to be added to the model pipeline such as inter linear programming (ILP) which applies local and global constraints to the output sequence [128].

Encoder-Decoder

The most recent and successful approach to disfluency detection is seen in encoder-decoder model architectures. The difference between these architectures compared to the approaches described above is that they allow for processing an entire sentence input at once. Also known as 'end-to-end' or 'sequence-to-sequence' models, such approaches are commonly used for tasks where capturing the long range dependencies across whole sentences is required, such as neural machine translation (NMT). This is achieved through the inclusion of what is known as an 'attention mechanism', which assigns learned weights to model states non-sequentially, thereby avoiding the vanishing gradient problem seen in RNNs. This is achieved either by adding an attention layer to RNN pipelines [146] or, more successfully, using a standalone attention mechanism, also known as a 'Transformer' model [147]. Both approaches have been applied to disfluency detection, with examples of the former modelled as an NMT problem [148, 149] and the latter using an adapted transformer model [150] as well as modelling disfluencies jointly with tasks such as ASR [151] and punctuation prediction [152].

The most successful encoder-decoder approaches for disfluency detection are those that are built on top of BERT models—'Bidirectional Encoder Representations from Transformers' [153]. The BERT model is a transformer-based language model, trained using unsupervised methods on very large datasets of billions of words. BERT is then applied as a baseline language model to be fine-tuned for downstream NLP tasks. This process is known as 'transfer learning' [108]. The current state-of-the-art approach for reparandum phrase detection uses a BERT model, achieving an F-score of 92.2 [154]. Further examples of BERT models being used for disfluency detection include joint modelling with constituency parsing [119] and punctuation detection [155] as well as domain and data augmentation for smaller on-device

approaches [156].

Aside from their high performance, the success of BERT models can also be attributed to the simplicity and universality of the input features that can be used to transfer to multiple NLP tasks. However, the requirement of transformer models for whole sentence inputs is not well suited to incremental approaches, which are restricted by their left-to-right operability, and the task of combating this is non-trivial (addressed further in Section 2.3.5 below). In addition, such language models trained on very large amounts of text scraped from the internet have been shown to promote negative bias against underrepresented populations in the data (see [108] for an overview). This mismatch in data type between written text of likely majority native speakers of English and transcribed speech from learners of English may cause similar problems.

2.3.3 Input Features

The model input features for the various approaches described above can be largely split into two categories: lexical and prosodic. For lexical features, up until the success of BERT, the vast majority of models relied on word pattern match features in order to capture the linguistic structure of disfluent phrases. Such features include the distance to a repeated word or bigram in a given window as well as POS tags and binary features such as whether or not a word is preceded by an edit term. Additionally, language model probabilities have been used as features themselves [120, 123], conceptualising repair starts as a type of deviation from fluency that could be identified by a low probability output from a language model trained on speech without disfluencies [143]. With the further development of neural network approaches came the use of static word embeddings, where each word has its own unchanging vector representation such as GloVe [157] and word2vec [158] (see Chapter Five for more detail on these approaches). Such embeddings were commonly included in combination with the pattern match and language model features described above [126].

Prosodic features are less commonly explored, often due to the fact that not all disfluency datasets provide audio files for experimentation. It is therefore more likely for work that experiments with ASR output to also include such prosodic information as model inputs. The majority of prosodic features used in such research are built on early work investigating

Table 2.1: *F-scores for reparandum phrase detection (Frm) according to modelling approach for all existing non-incremental disfluency detection models that have been trained and tested with the Switchboard Corpus.*

Approach	Model	*Frm*
Text parsing	Charniak & Johnson 2001 [88]	0.76
	Kahn et al. 2005 [111]	0.78
	Wu et al. 2015 [114]	0.85
	Tran et al. 2017 [117]	0.78
	Wang et al. 2017 [116]	0.88
	Lou et al. 2019 [118]	0.89
	Lou & Johnson 2020* [119]	0.91
Noisy-channel	Johnson & Charniak 2004 [120]	0.80
	Zwarts & Johnson 2011 [123]	0.84
	Lou & Johnson 2017 [125]	0.87
Sequence labelling	Zayats et al. 2014 [143]	0.83
	Ferguson et al. 2015 [145]	0.85
	Zayats et al. 2016 [126]	0.86
Encoder-decoder	Wang et al 2016 [149]	0.87
	Dong et al. 2019 [150]	0.89
	Bach & Huang 2019* [154]	0.92
	Wang et al. 2020* [162]	0.91
	Rocholl et al. 2021* [156]	0.92

*Models that use BERT.

reliable predictors of the interruption point of repairs [90, 159, 160] such as pauses in speech (represented as the presence of pauses, number of pauses in a given phrase and the duration of pauses) and pitch (represented as the fundamental frequency as well as the distance of a word's average pitch to a speaker's pitch floor). Prosodic features have been included in text parsing, sequence labelling and encoder-decoder approaches showing improved performance compared to lexical approaches alone (see [111, 117, 145, 151, 161] for examples). The impact of prosody on state-of-the-art approaches using BERT models, however, is not yet known.

2.3.4 Performance Evaluation

The work conducted by Charniak and Johnson [88] was the first to formalise methods for disfluency detection performance evaluation. The first aspect of this formalisation was to introduce the use of the Switchboard Corpus [163] for model training and evaluation. Elaborated in further detail in Chapter Three, the Switchboard Corpus contains transcripts and

audio files of telephone conversations, human-labelled for disfluency features among other linguistic phenomena. The labelling process follows Shriberg's notation scheme outlined above in Section 2.3.1, labelling reparandum phrase, interregnum phrase and repair phrase as well as preserving the structure of any complex repairs. Following this labelling schema, Charniak and Johnson introduced the evaluation metric *Frm* for disfluency detection, which is the F-score for a model correctly identifying a word being part of the reparandum phrase of the repair [88]. As is the case for the majority of disfluency detection, attention is given only to reparandum phrases for subsequent removal and so the detection of repair phrases is not measured.

Table 2.1 shows the performance of all of the above-mentioned studies that report the *Frm* for the Switchboard corpus as defined by Charniak and Johnson [88]. It is worth noting that there are some differences between approaches, with some choosing to exclude partial words and edit terms from the dataset (see [118, 156] for more detailed comparisons between models). As can be seen, there are strong outcomes across the various approaches, however in recent years the most successful are those from text parsing and encoder-decoder approaches, especially those that leverage BERT models (marked by an asterisk in the table) [119, 162, 154, 156].

Beyond *Frm*, many studies go on to evaluate the model performance with error analyses. Firstly, disfluency type has been shown repeatedly to impact detection performance, with repetitious instances of disfluencies being shown to be significantly easier to detect than non-repetitious instances [116, 143, 126, 150, 162]. Deletion sub-types have been shown to be the hardest to detect [151]. Other features of disfluencies shown to reduce model performance include reparandum length [127] and the presence of partial words [118]. Errors from ASR transcripts also pose a challenge, with models trained on such transcripts seeing a significantly lower performance than those trained using human transcriptions [115, 117, 151]. Finally, features of the sentence containing the disfluency can also cause disfluencies to go undetected, most notably longer sentences [155] and sentences with grammatical errors [127].

2.3.5 Incremental Disfluency Detection

As described above, the standard approach to disfluency detection is built around the aim of disfluency removal and with the later approaches, in particular, sentences are processed as

a whole either using bi-directional LSTMs in sequence labelling or through encoder-decoder approaches. However, there is a subset of disfluency detection research that is focused on processing disfluencies incrementally, retaining the disfluent speech. This approach, commonly adapted for use in spoken dialogue systems is justified for two reasons, the first being that disfluencies themselves have a linguistic function [164] and retention of their structure is often required for resolving meaning [165]. Take this example from Core and Schubert [110]:

Have the engine [take the oranges to Elmira + {um I mean} take them to Corning]

Here it is clear that "*them*" from the repair phrase is referring to "*the oranges*" in the reparandum phrase. However, in a conventional disfluency detection approach, the reparandum phrase containing the detail about the oranges would be removed, which is an undesirable outcome in a conversational system. The second reason shares the perspective of incremental dialogue processing at large; dialogue is inherently incremental and as such, dialogue processing tasks should function within an incremental framework [166]. For the application of disfluency detection, that is to detect disfluencies at the point of repair onset, rather than after the completion of a learner's utterance. There are fewer studies exploring incremental detection compared to the non-incremental approaches above, however, interest has been building in recent years with the motivation to incorporate live ASR streaming models directly on-devices [167].

The trajectory of incremental approaches broadly follows that of non-incremental detection, with text parsing [168, 169] and sequence labelling [170, 171] forming some of the first iterations of research. The functionality of these early systems can be considered incidentally incremental thanks to their left-to-right operability, but as argued by Hough and Purver [172], these systems are not explicitly optimised for incremental processing. The first system that is considered to be intentionally optimised for incremental detection is that developed by Zwarts et al. [173]. This approach prioritised model responsiveness (i.e. accurately detecting a disfluency as soon as possible after its occurrence) and adapted Johnson and Charniak's [120] noisy-channel model with a Tree Adjoining Grammar (TAG) to operate incrementally. Inspired by this approach, Hough and Purver implemented a pipeline of Random Forest classifiers that modelled the detection of reparandum start, repair start and interregna separately

Table 2.2: *F-scores for reparandum phrase detection (Frm) and repair start detection (Frps) according to modelling approach for all existing incremental disfluency detection models that have been trained and tested with the Switchboard Corpus.*

Approach	Model	*Frm*	*Frps*
Noisy-channel	Zwarts et al. 2010 [173]	0.78	-
	Hough & Purver 2014 [172]	0.78	-
	Purver et al. 2018 [86]	0.78	0.85
Sequence labelling	Hough & Schlangen 2015 [174]	0.71	-
	Hough & Schlangen 2017 [175]	0.60	0.72
	Shalyminov et al. 2018 [176]	0.75	0.82
	Rohanian & Hough 2020 [177]	0.74	0.81
Encoder-decoder	Rohanian & Hough 2021 [179]	0.76	0.85

using probability-based input features derived from trigram language models [172]. The same model was adapted to include partial words in the training data, leading to a marginal increase in performance (0.781 compared to 0.779) [86]. Hough and Schlangen applied an RNN with word embeddings and POS tags as input features, using one model as a multi-class sequence labeller [174]. Modelling tasks jointly has also been explored, using LSTMs for both disfluency detection and utterance segmentation [175] extended further to multi-task learning with utterance segmentation, POS tagging and language modelling [176, 177].

Echoing the outcomes of general disfluency detection research, the most recent explorations of incremental approaches apply fine-tuned BERT language models. As touched on above, the encoder-decoder transformer architecture of BERT models requires whole sentence inputs which is a non-trivial task for incremental systems that are developed with left-to-right operability. As it stands, there are two approaches to this problem. The first, inspired by work on incrementalising transformer architectures for other NLP tasks [178] is to combine unfinished utterances with predictions generated from a GPT-2 language model to form full sentences to be used as inputs [179]. The second is to train a BERT model to decide whether or not to wait for further context before outputting a disfluency prediction [167].

Table 2.2 shows the utterance-final results of the intentional incremental systems trained and tested using the transcribed version of the Switchboard corpus. As well as the *Frm* score defined for general disfluency detection research, *Frps* is also commonly reported for repair start detection (i.e. the F-score of the correct identification of the first word of the repair

phrase). As the results show, the *Frm* results are generally lower for incremental approaches compared to non-incremental approaches. In addition, despite increasing performance for neural models used for sequence labelling and encoder-decoder approaches, noisy-channel models remain the state-of-the-art for *Frm*. Other measures of incremental detection performance for accuracy and latency include 'time-to-detection' (how many words are processed following a disfluency before detection) and 'delayed accuracy' (within a window of n words, how often are disfluencies correctly identified) [123] as well as 'edit-overhead' (the number of unnecessary edits to the labelling output structure) [180]. These measures are typically task-specific and are not consistently reported across papers so are not included here.

In the same way as non-incremental approaches, incremental detection has been shown to be impacted by repair sub-type and reparandum length [86, 179]. An additional challenge for incremental approaches is that they are typically designed for use in real-time settings, using unsegmented, live transcripts generated by ASR models. The output of such models inevitably contains model errors, and this 'noisy' data has been shown to significantly reduce model performance—the strongest performing models on human transcripts only achieve *Frps* scores of 0.6 and *Frm* of 0.5 on data produced by ASR models (see [179] for a more detailed overview of the impact of such processes on recent model performance). Looking next to the reduced performance of incremental approaches compared to non-incremental approaches, it can be in part attributed to the restriction of left-to-right operability imposed on incremental frameworks. As a consequence of this restriction, incremental approaches rely strongly on successful repair start detection. The application of BERT-based encoder-decoder models with generated inputs seeks to address this issue [179], however, overall performance remains behind the state-of-the-art score established by Hough and Purver [172]. With the above in mind, it may be of benefit to explore additional methods of boosting repair start detection. Exploring the application of prosodic features as model inputs seems an obvious next step, given both their overall positive impact on non-incremental approaches and linguistic salience immediately prior to repair start.

2.3.6 Disfluency Detection for Learner Speech

Currently, there have been only three approaches to disfluency detection for learner speech. The first involves the adaptation of an incremental transition-based joint dependency parsing and disfluency detection model [181]. For this approach, 'Redshift', an open-source incremental parsing model trained on the Switchboard Corpus [169] was tested on the BULATS corpus, which comprises transcriptions of the speaking tests of English learners from a variety of first language backgrounds [182].[4] Reparandum phrase detection performance was found to be low for the learner speech corpus (*Frm* of 0.48) and also a positive correlation was found between learner proficiency level and disfluency detection accuracy. Although not tested explicitly, this led to the hypothesis that learner errors may have a negative impact on detection performance. The second and third approaches to detection stem from the first, with the aim of applying disfluency detection for the downstream task of grammatical error detection and correction. Approaching the task non-incrementally and using flattened disfluency structures (i.e. removing complex nesting labels), two further approaches to disfluency detection in learner speech have been explored. The first is a sequence labelling approach using a bi-directional LSTM with word embeddings, POS tags and hand-crafted features [183] and the second is an encoder-decoder model leveraging word embeddings alongside acoustic features using an attention mechanism [184]. These models were also trained using the Switchboard corpus and tested on learner speech. Both the BULATS corpus and the NICT-JLE corpus were used, (the latter consists of transcriptions of speaking tests of Japanese learners of English, the details of which are expanded on in Chapter Three). The bi-directional LSTM model was more successful for disfluency detection specifically, with *Frm* scores of 0.80 and 0.64 for the NICT-JLE and BULATS corpora, respectively, compared to *Frm* scores of 0.64 and 0.54 for the same corpora tested on the end-to-end model. Echoing the results with native speech explored above, in both approaches, using ASR transcription outputs for testing data led to a poorer performance of disfluency detection models.

Much like the majority of approaches described previously, the above three models were designed to convert ASR output from learners' speech to a form that is more similar to written text through reparandum phrase detection and subsequent deletion and so are not

[4]This corpus was provided by Cambridge Assessment English and is not available for public use.

entirely suitable for the problem defined for this thesis. Although the first approach operates incrementally, it can be considered an 'incidental' approach similar to the early work in this area. This limitation of the model together with its particularly poor performance provides further motivation to explore alternative approaches to incremental detection of disfluencies in learner speech. For all three approaches, models were trained using native speech data. With the knowledge outlined above that disfluency behaviour seen in learner speech differs from that of native speakers, it seems likely that using learner speech for training and testing would lead to improved model performance, especially for lower proficiency learners. The BULATS corpus used in those studies is not publicly available, however, there is potential for the NICT-JLE corpus to be adapted for such purposes. It is therefore of interest to establish the suitability of the NICT-JLE corpus for model training. Finally, the presence of learner errors was cited across all studies as a potential cause of poor model performance. These observations along with evidence of a similar impact in the native speech approaches discussed above [127] provide another point of further investigation.

Dialogue-based CALL has been shown to be an important emerging field considering the recent trends in CALL research toward machine learning-based solutions. In particular, the success of speech recognition and natural language processing technologies have facilitated the sophistication of spoken dialogue systems, such that a version of the technology is now commercially available on devices such as smartphones and smart speakers. Dialogue-based CALL has the opportunity to leverage these recent developments to create systems that better accommodate the nature of teacher-learner dialogue, in turn facilitating more meaningful opportunities for language learners to practise their conversation skills. One of the many areas of interest is that of corrective feedback. Taking an interdisciplinary approach to evaluating corrective feedback in dialogue-based CALL systems, findings revealed a mismatch in the handling of learner errors between teacher-learner dialogue and system-learner dialogue. This mismatch shows that existing dialogue-based CALL systems do not have the appropriate functionality to facilitate learner self-correction. To address this issue, a framework for such functionality was proposed and incremental self-initiated self-repair detection for learner speech was identified as a key area for further research. The linguistic structure, features and variation of self-initiated self-repairs were defined, followed by an overview of the main approaches to their automatic detection. Various challenges for disfluency detection were highlighted, namely the impact of reparandum phrase length and non-repetitious disfluencies on model detection across all approaches, the restriction of left-to-right operability for incremental frameworks and finally the impact of learner errors on model performance. From these findings the main goal of the thesis was introduced: to build an incremental disfluency detection model that is suitably adapted for learner speech.

Chapter 3

A Comparative Analysis of L1 and L2 Disfluency Corpora

As illustrated in the previous chapter, the vast majority of disfluency detection research is built for native speech and it is not clear how contemporary incremental approaches would perform using a learner corpus. With the knowledge that native (L1) and learner (L2) disfluency behaviour differ in a variety of ways such as frequency of occurrence and distribution of disfluency type [185, 186], it needs to be established if and how these differences are represented in existing corpora for disfluency detection. With this comes the aim to identify the challenges specific to disfluency detection for learner speech through a linguistic comparative analysis of two available corpora that contain labelled disfluency structures—one for native speakers of English and one for learners of English. Findings from this evaluation identified several points of difference between L1 and L2 disfluency behaviour. The subsequent analysis investigates how these features of disfluency relate to one another, followed by an exploration of how speaker proficiency further impacts these features in the learner speech corpus. These differences are considered in the context of existing linguistic research in order to understand how they may impact disfluency detection models. From this analysis, five key disfluency features are identified as potential difficult-to-detect cases.

3.1 Introducing the Corpora

Two corpora were chosen for this investigation: the Switchboard corpus (L1 data) [163] and the National Institute of Information and Communications Technology Japanese Learner English (NICT-JLE) corpus (L2 data) [7]. The corpora were chosen due to their similarity and therefore comparability as both contain transcriptions of similar conversational tasks as well as labelled disfluency features. Although there are other conversational corpora for both L1 and L2 data, at the time of investigation the corpora stated above were the only two with labelled disfluency features that were available for public use[1]. See Table 2 in the Appendix which provides an overview of corpora that were considered and subsequently ruled out for use in this research.

The Switchboard corpus is made up of 2,400 telephone conversations (approximately 260 hours of speech) among 534 speakers from across the United States, covering 70 pre-defined conversation topics. Speaker pairing was determined to ensure no two speakers would engage in conversation more than once, and no speaker would speak on a certain topic more than once. For the analysis described in this chapter, the disfluency-tagged subset of Switchboard dialogues was used [188], which contains 650 transcribed and POS-tagged conversations (approximately 70 hours of speech) labelled with edit terms and disfluencies and follows the standard division into train (80%), heldout (10%) and test (10%) sets as established by Johnson and Charniak [122]. As described in Chapter Two, the Switchboard corpus is the most popular corpus for current disfluency detection research for native speech and is commonly used as a benchmark for model evaluation.

The NICT-JLE corpus consists of 1,281 English oral proficiency tests (approximately 300 hours of speech) of Japanese-speaking learners of English. The Standard Speaking Test (SST) is made up of various conversational tasks. At the beginning and end of the test, the learner is encouraged to engage in 'small-talk' style open dialogue with the assessor, covering topics such as the weather, what plans the learner has for after the assessment and so on. The majority of the test is made up of two activities taken from a pre-defined set according

[1]Since this investigation, The KIT Speaking Test Corpus (KISTEC) [187] has been released. The KIT corpus contains transcriptions of Japanese-speaking learners of English engaging with a computer-based English speaking test. The disfluency tagging follows the same approach as the NICT-JLE corpus. Given its similarity to the NICT-JLE corpus, as well as the fact that the data recorded is from human-computer interaction, it is of interest to use this corpus in any future research relating to L2 disfluency detection.

Table 3.1: *General linguistic features of the Switchboard and NICT-JLE corpora*

	Switchboard	NICT-JLE
total words	746290	1165785
total utterances	102169	178934
vocabulary size	16810	13499
logTTR	0.85	0.68
average utterance length (SD)	7.30 (3.61)	6.51 (3.27)

to the learner's proficiency: a role-play scenario and a picture description task. The corpus provides transcriptions of the test labelled for disfluencies, edit terms as well as 'non-verbal sounds' such as laughter and silence. Each file also contains meta-data relating to the learners' prior English proficiency, nationality and gender. A subset of files (167) contain additional annotation of morphological, grammatical and lexical errors. Presently, the NICT-JLE corpus is currently the only publicly available dataset of learner speech labelled with disfluencies. Existing disfluency detection research has only used the grammatical error labelled subset of the corpus for model evaluation (see [183, 184]) and there are currently no studies that use the NICT-JLE corpus for model training.

Table 3.1 compares a range of general linguistic features of the two corpora. The NICT-JLE corpus is the larger of the two, with a higher number of both words and utterances (11165785 words and 178934 utterances compared to 746290 words and 102169 utterances). However, the Switchboard corpus has a larger vocabulary size than the NICT-JLE corpus (16810 compared to 13499). This is reflected in the values for the Type-token ratio (TTR): a measure for 'lexical richness' taken by dividing the vocabulary size by the total number of words in a given segment of text [189]. The closer the TTR value is to one, the higher the lexical richness of a given text. Here the logTTR is reported (logV/logN), as it better accounts for the sample size variation seen between the corpora [190]. The logTTR is 0.85 for the Switchboard corpus and 0.69 for the NICT-JLE corpus, indicating that the Switchboard corpus is the more lexically rich of the two. Finally, the average utterance length in the Switchboard corpus is just under one word longer than the average utterance length in the NICT-JLE corpus (7.30 compared to 6.51).

Table 3.2: *Disfluency features of the Switchboard and NICT-JLE corpora*

	Switchboard	NICT-JLE
total disfluency phrases	22820	64346
disfluency phrases per 100 words	3.06	5.52
total disfluency instances	26597	87887
disfluency instances per 100 words	3.56	7.54
disfluency phrase average reparandum length (SD)	1.61 (1.13)	2.05 (1.71)
disfluency instance average reparandum length (SD)	1.58 (1.12)	1.62 (1.08)
total nested disfluencies	5767	34587
nested disfluencies/total disfluency instances	21.68	39.35
total non-repetitious disfluencies	13153	40075
non-repetitious disfluencies/total disfluency instances	49.45	45.60
total edit terms	53410	162718
total interregna	5896	27327
edit terms per 100 words	11.04	16.79
interregna/total edit terms	8.00	14.67
disfluency instances with interregna/total disfluency instances	22.17	31.09

3.2 Disfluencies in the Switchboard and NICT-JLE Corpora

3.2.1 Comparing General Disfluency Features

The approaches followed to define the disfluency attributes of the NICT-JLE and Switchboard corpora are briefly explained here, starting with the differentiation of disfluency phrases and disfluency instances:

$$\text{disfluency phrase: } \underbrace{[\textit{I want } \overbrace{[\textit{to + to}]}^{\text{inner disfluency}} \textit{ go} + \textit{I wanted to go}]}_{\text{outer disfluency}}$$

$$\text{inner disfluency instance resolved: } [\textit{I want to go + I wanted to go}]$$

The above disfluency phrase contains two disfluency instances: the inner disfluency instance and the outer disfluency instance. For all disfluency phrases, disfluency instances are resolved from the inside out, following the labelling guidelines for the Switchboard corpus, defined by Meteer et al. [188]. The inner disfluency instance, in this example a simple one-word repetition, has a reparandum phrase length of one. For the outer disfluency instance, the reparandum phrase length is counted as if the inner disfluency is resolved and so has a

Table 3.3: *Percentage distribution of disfluency instances according to reparandum phrase lengths for the Switchboard and NICT-JLE corpora, showing little difference between the two.*

	% Distribution	
`rm` Length	Switchboard	NICT-JLE
1	66.2	63.7
2	21.1	22.5
3	6.8	8.2
4	3.1	3.1
5+	2.7	2.6

Table 3.4: *Percentage distribution of disfluency instances per disfluency phrase for the Switchboard and NICT-JLE corpora, showing that the NICT-JLE corpus has a higher rate of nested disfluencies of size two or more.*

	% Distribution	
Instances	Switchboard	NICT-JLE
1	88.6	78.7
2	9.9	15.6
3	1.2	3.9
4	0.2	1.2
5+	0.1	0.6

length of four. A disfluency phrase that contains one disfluency instance is considered to be non-nested, whereas a disfluency phrase that contains more than one disfluency instance (as above) is labelled as nested.

Table 3.2 compares a variety of disfluency features found in the Switchboard and NICT-JLE corpora. As the table shows, there is a higher number of both disfluency phrases and instances in the NICT-JLE corpus. On examination of their frequency per 100 words, it can be seen that disfluency phrases occur in the NICT-JLE corpus almost twice as often as they do in the Switchboard corpus (5.52 occurrences per 100 words compared to 3.06), with disfluency instances occurring over twice as frequently (7.54 occurrences per 100 words compared to 3.56).

For disfluency instances, the average reparandum phrase length is comparable between the corpora: the average length in the NICT-JLE corpus stands at 1.62 words compared to 1.58 words in the Switchboard corpus. This is also reflected in the frequency distribution of

reparandum phrase lengths, as shown in Table 3.3, with the NICT-JLE corpus having only a marginally higher proportion of disfluency instances with a reparandum length of two or more (36.4% in the NICT-JLE corpus compared to 33.7% in the Switchboard corpus). Disfluency phrases as a whole, however, have a higher average length in the NICT-JLE corpus compared to the Switchboard corpus—2.05 and 1.61, respectively.

This difference in disfluency phrase length can partly be explained by the higher rate of nested disfluencies occurring in the NICT-JLE corpus. As shown in Table 3.2, over a third of disfluency instances (39.35%) in the NICT-JLE corpus are also nested—approximately twice as many as those in the Switchboard corpus (21.68%). Table 3.4 summarises the distribution of these disfluency phrases according to the number of nested instances per phrase, where disfluency phrases without any nesting are considered to have a size of one. As is shown, the NICT-JLE corpus has a higher proportion of disfluency phrases with two or more instances than those in the Switchboard corpus.

Looking next to non-repetitious disfluency instances, the Switchboard corpus has marginally higher rates than the NICT-JLE corpus (49.45% and 45.60%, respectively). Edit terms occur more frequently per 100 words in the NICT-JLE corpus compared to the Switchboard corpus (16.79 and 11.04, respectively). When considering the proportion of edit terms that are interregna (i.e. part of a disfluency instance), the NICT-JLE corpus is once again higher than the Switchboard corpus with a rate of 14.67%, almost double that of the 8.00% seen in the Switchboard corpus. Finally, almost a third of disfluency instances contain an interregnum in the NICT-JLE corpus compared to the Switchboard corpus (31.09% and 22.17%, respectively).

The general statistics described above compare each disfluency characteristic as separate features, however, disfluencies can contain several of such features. For example, consider the example below:

disfluency phrase: $[I + [I\ work + \{um\}\ I\ work]]$ *for five or six days*

inner disfluency instance resolved: $[I + I\ work]$ *for five or six days*

The inner disfluency instance is labelled as repetitious, with-edit, nested, and has a reparandum phrase length of two. With the inner instance resolved, the outer disfluency

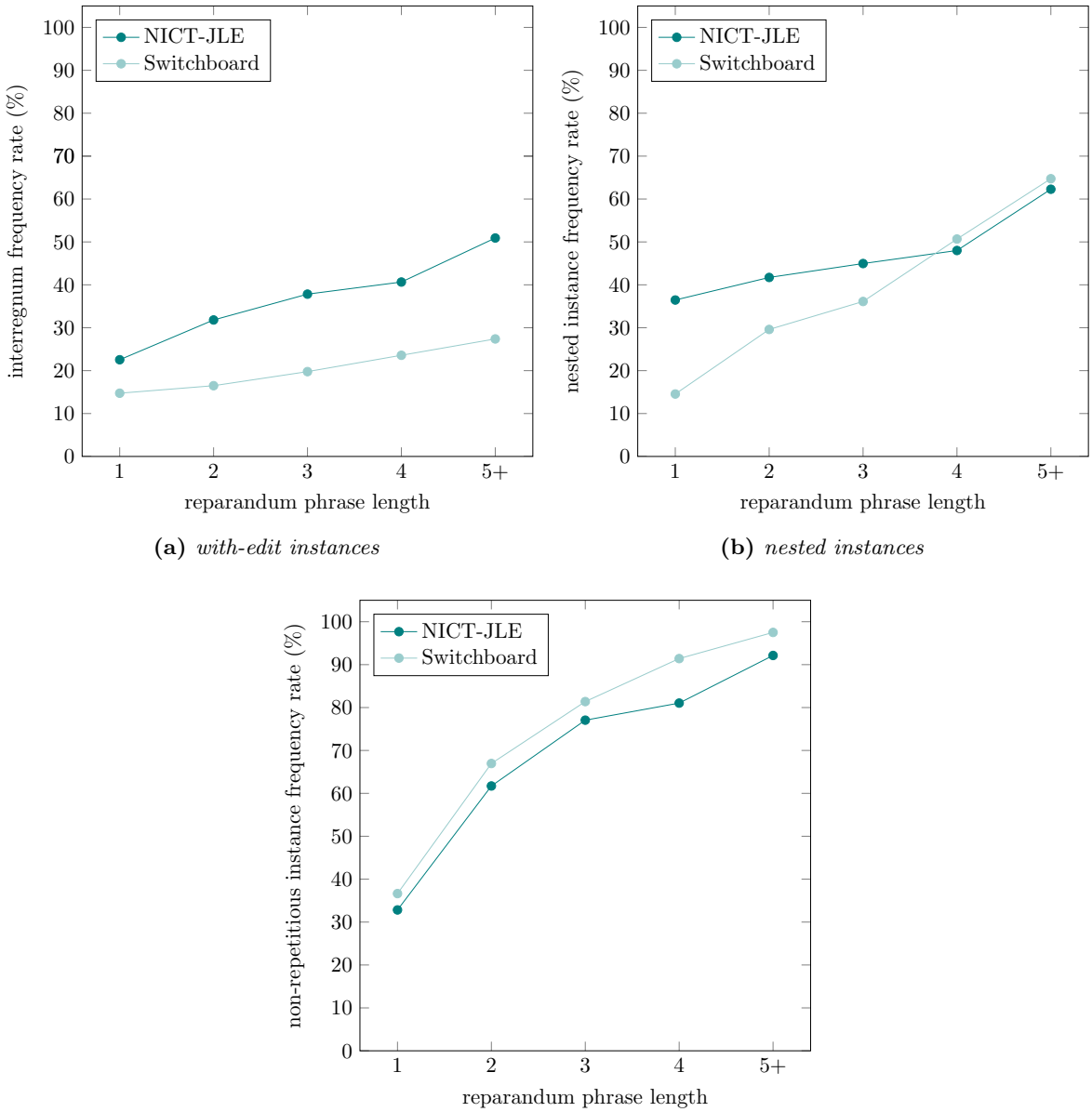
Table 3.5: *Percentage distribution of disfluency features per disfluency instance for the Switchboard and NICT-JLE corpora.*

	Frequency rate (%)	
Disfluency Type	Switchboard	NICT-JLE
repetitious	38.9	26.7
repetitious with edit	5.9	7.0
non-repetitious	26.7	18.1
non-repetitious with edit	6.9	8.8
repetitious (nested)	5.2	16.5
repetitious with edit (nested)	0.6	4.2
non-repetitious (nested)	13.2	11.5
non-repetitious with edit (nested)	2.7	7.2

instance is non-repetitious and nested and has a reparandum phrase length of one. Based on this information, the following analysis considers the relationships between disfluency features and how they differ across corpora.

Figure 3.1 shows the relationships between reparandum phrase length and the frequency rate of with-edit, nested and non-repetitious disfluency instances in both the Switchboard and NICT-JLE corpora. For all features across both corpora, as reparandum length increases so does the likelihood that a disfluency instance contains such features. For with-edit instances depicted in Figure 3.1a, the correlation is stronger for the NICT-JLE corpus, with approximately 50 percent of disfluency instances of length five or more containing an edit term. For nested disfluency instances depicted in Figure 3.1b, the correlation with reparandum length is stronger for the Switchboard corpus. This is due to the fact that the NICT-JLE corpus has a higher frequency of nested instances for reparandum phrase lengths between one and three, with rates becoming comparable between both corpora for reparandum phrase lengths of four and higher. Finally, Figure 3.1c shows a strong correlation between reparandum length and non-repetitious disfluency, with over 90% of disfluency instances with reparandum lengths of five or more being non-repetitious for both corpora.

Table 3.5 shows the co-occurrence rates of nested, repetitious, non-repetitious and with-edit instances for both the Switchboard and NICT-JLE corpora. As the distribution shows, the majority of disfluency instances in the Switchboard corpus are standalone repetitious or non-repetitious, i.e. they are neither nested nor co-occur with edit terms. This group makes



(a) *with-edit instances*

(b) *nested instances*

(c) *non-repetitious instances*

Figure 3.1: *The percentage of disfluency instances that are nested, non-repetitious and with-edit according to reparandum phrase length for the NICT-JLE and Switchboard corpora, showing a positive correlation between length and all three features.*

up approximately two-thirds (65.6%) of all instances, compared to the 44.8% seen in the NICT-JLE corpus. Of the nested instances, the NICT-JLE corpus has almost three times as many repetitious disfluencies that are part of a nested phrase compared to the Switchboard corpus (16.5 compared to 5.2, respectively). The rate of both repetitious and non-repetitious nested instances that co-occur with edit terms is low for both corpora, although it is more so for the Switchboard corpus. For non-repetitious instances that are nested, however, the Switchboard has marginally higher rates (13.2% compared to 11.5% in the NICT-JLE corpus).

3.2.2 Learner Errors in the NICT-JLE Corpus

Language errors are an inherent part of learner speech, occurring at a much higher rate than in native speech. As described earlier, 167 of the transcript files in the NICT-JLE corpus contain tags for learners' morphological, grammatical and lexical errors (phonological errors are not annotated). The examples below taken from the NICT-JLE corpus reflect how errors co-occur with disfluency structures. The examples are labelled for disfluencies and words in bold highlight learner errors.[2]

(1) My computer [**use** + {er} is used] by [**all family** + my family]

(2) She [[**wanted shopping** + **wanted shop**] + {er} wanted to go shopping]

(3) [[I don't + **I'm not have watching movie**] + I don't have **no** time to **watch movie**]

The examples illustrate how learner errors can occur in all parts of the disfluency phrase. The first example shows an example of an 'error-type' disfluency [185], where the error in the reparandum phrase is repaired to its correct form. The second example shows the presence of errors in a nested disfluency phrase, where the learner's first attempt at correcting the initial error fails and is followed by a second, correct repair phrase. The third example shows a failed attempt to fully correct an error, where the repair phrase for the outer disfluency instance remains erroneous. There are approximately 11 instances of annotated learner errors per 100 words in the NICT-JLE corpus. However, the actual number of errors present in the corpus is likely to be higher as errors that occur in the reparandum phrase are not annotated.

[2]The same examples are used in the conference paper for BEA 2022 outlined in the Introduction of this thesis [8].

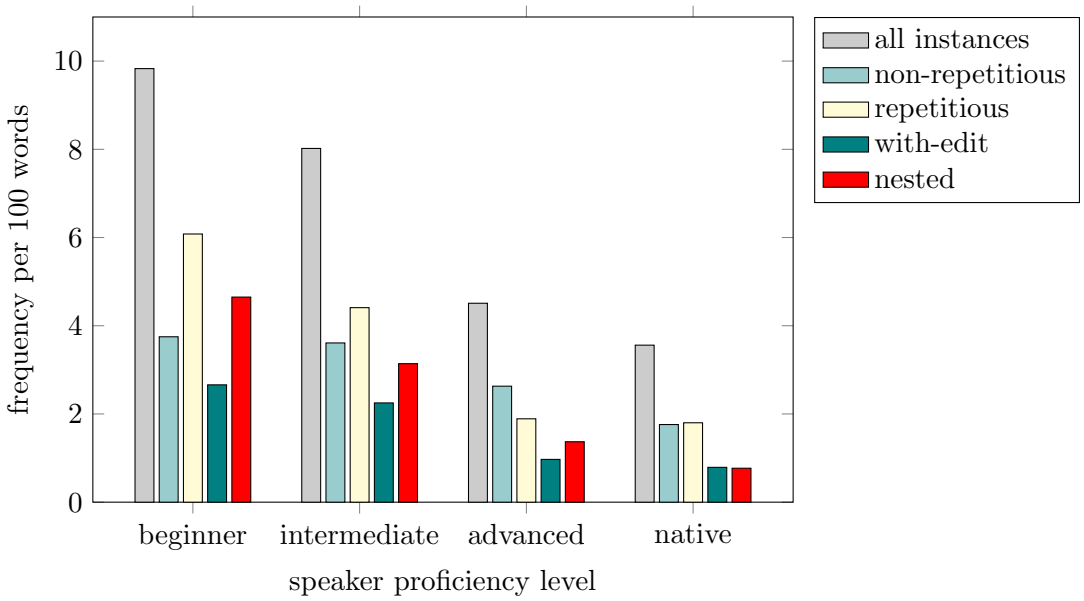


Figure 3.2: *Frequency per 100 words of disfluency instances, subsequently broken down into non-repetitious, repetitious, with-edit and nested disfluency instances according to speaker proficiency level.*

3.2.3 Impact of Speaker Proficiency on Disfluency Behaviour

Figure 3.2 shows the frequency per 100 words of all disfluency instances, and subsequently, repetitious, non-repetitious, with-edit and nested instances according to speaker proficiency level. The 'beginner', 'intermediate', and 'advanced' levels refer to speakers from the NICT-JLE corpus at SST levels one to three, four to six and seven to nine, respectively. The 'native' group refers to all speakers in the Switchboard corpus. As Figure 3.2 shows, the frequency of all disfluency instances decreases as speaker proficiency level increases. This is also reflected in the features of disfluencies. Repetitious disfluency instances show the steepest decline between beginner and advanced proficiency levels, with frequency levelling off between advanced and native speakers whereas the frequency of non-repetitious disfluencies shows a smaller slope of decline as proficiency increases. Notably, Figure 3.2 shows the change of frequency distribution of repetitious and non-repetitious disfluency instances. The ratio of repetitious to non-repetitious disfluency instances is at its highest for beginner proficiencies and marginally higher for intermediate proficiencies. For advanced speakers, the distribution switches and non-repetitious disfluencies are shown to be more frequent until finally, the distributions are approximately equal for native speakers.

3.3 Considerations for Incremental Disfluency Detection Models for Learner Speech

As can be seen from the data described above, there are some key differences between the NICT-JLE and Switchboard corpora in terms of linguistic complexity and disfluency behaviour. The Switchboard corpus is more lexically complex across measures of vocabulary size, utterance length and logTTR. These figures are reflective of second language acquisition research that determines both vocabulary size and average utterance length as predictors of learners' speaking proficiency [191, 106], where a larger vocabulary equates to a higher speaking proficiency. However, it is important to note that language proficiency may not be the only cause of increased vocabulary size as although similar, the Switchboard and NICT-JLE corpora are not perfectly comparable. Firstly, the NICT-JLE corpus covers a smaller amount of conversational topics than the Switchboard corpus—22 and 70 respectively, which may be a factor in the reduced vocabulary size of the NICT-JLE corpus. Furthermore, task type for learners has been shown to have an impact on vocabulary size [192]. For example, picture describing tasks (which constitute approximately a third of each conversation in the NICT-JLE corpus) have been shown to elicit less frequent words, and therefore wider vocabulary, from learners [193]. From a modelling perspective, having a lexically 'simpler' dataset may allow the disfluency detection model to generalise better. This is not only beneficial for learner speech, but also for specific speaking assessment tasks such as role-play scenarios.

Comparing the disfluency characteristics of both corpora, the NICT-JLE corpus has overall higher rates of disfluency in terms of both disfluency phrases, instances and edit terms. This reflects previous findings on the self-repair behaviour of learners which attributes the lower degree of learners' language 'automatisation' to the increased number of mistakes and hesitations found in their speech [104, 105]. The same behaviour is seen for filled pauses, which have higher rates in learner speech compared to native speech [106, 107]. It has also been shown that learner proficiency level influences the disfluency features of learner speech—as proficiency increases, the distribution of features becomes more similar to that of a native speaker [185]. For example, the number of repetitions in learner speech has been shown to be correlated with linguistic knowledge, where repetitious disfluency rates decrease as proficiency increases [107]. These findings are corroborated by the NICT-JLE corpus data shown

in Figure 3.2, and may also be a factor as to why the same trend is seen for nested disfluency instances. The impact of these findings on model performance has mixed consequences. Prior incremental disfluency detection research has shown that repetitious disfluency instances are easier to detect than non-repetitious disfluency instances [172]. Although the overall discrepancy between repetitious disfluencies in the Switchboard and NICT-JLE corpus is minimal, it may be particularly beneficial for disfluency detection in beginner and intermediate learners' speech, given the higher rates of repetitious instances seen in these groups. However, given that it is not currently clear how the co-occurrence of nesting and with-edit features impacts model performance, the high rate of repetitious repairs without such features in the Switchboard corpus may also be beneficial for detection.

Findings from the data also revealed the impact of reparandum length on disfluency behaviour. Prior research has shown the degradation of detection performance as reparandum length increases, citing the difficulty that RNNs often face due to the vanishing gradient problem [172]. Although the corpora are comparable in terms of reparandum length distribution, it is clear from the data explored in Figure 3.1 that a secondary influential factor is reflected in the correlation between reparandum phrase length and complex disfluency features such as non-repetitious, nested and with-edit disfluency instances. As the data show, disfluency instances with longer reparandum phrases and interregna may pose a particular issue for models trained and tested using the NICT-JLE corpus given their high prevalence in the dataset.

The presence of learner errors in the NICT-JLE corpus is also of specific interest as they are often cited as a contributing factor to the difficulty of NLP tasks for learner language data such as POS tagging [194] and parsing [195]. Additionally, as discussed in Chapter Two, such errors have been hypothesised (but not yet tested) to be detrimental to the performance of disfluency detection models [181, 183, 184]. Prior research has shown that learner speech tends to contain a higher rate of 'error-type' disfluencies (where the error forms all or part of the reparandum phrase and is subsequently corrected in the repair phrase) compared to other types [185]. However, as illustrated above, this is not the only way that errors appear in learner speech. They can occur both outside and inside of disfluency phrases and for the latter, can occur in any or all parts of the disfluency structure. Due to the limited scope of learner error labelling in the NICT-JLE corpus, it is not known whether the proportion

of error-type disfluencies is higher or lower than the frequency of errors occurring elsewhere in learner speech, and as a consequence, it is unclear if such errors will impede disfluency detection. Existing research on disfluency detection using the NICT-JLE corpus focuses on removing disfluency for the purpose of grammatical error detection and correction and as a result, does not explore the impact of learner errors on detection performance itself [183, 184]. It is therefore of interest to explore this relationship in further detail.

3.4 Limitations

In the NICT-JLE corpus, disfluency types are either labelled as repetitious or non-repetitious. However, in the Switchboard corpus, a further distinction is made in the non-repetitious category between deletion and substitution disfluency types. As a result, the comparison conducted between the two corpora is limited to the scope of the NICT-JLE corpus labelling. The same can be said for the approach to learner error labelling as previously indicated above. Not only is labelling limited to a subset of files (167 out of 1281), but it is also limited in that any errors that occur in the reparandum phrase are not annotated. This in turn shrinks the scope for analysis. Nevertheless, the analysis that was possible for both disfluency type and learner error behaviour has provided ample insights for what follows in Chapters Four and Five.

The NICT-JLE corpus is additionally limited in that it only contains Japanese-speaking learners of English. With the knowledge that learners' native language can influence factors such as disfluency frequency [196], it would be valuable to analyse data from learners with varied language backgrounds. However, as it stands, there are currently no such corpora that are publicly available to carry out this kind of analysis. Despite the limited scope of the NICT-JLE corpus in terms of L1 variation, the parallels seen between its data and existing research on learner disfluency behaviour are supportive of it being used as a proxy for learner speech at large.

Finally, there are additional individual differences beyond proficiency that influence disfluency behaviour in learner speech that were not considered in this analysis, for example, learners' sociolinguistic background [197], preference for self-repair [198] and L1 disfluency frequency [196]. For this investigation, the aim was to establish the differences between learner

and native disfluency corpora more broadly and as such, these inter-group differences fell out of the scope of this investigation. However, the fact that there is variability of this nature within learner speech motivates the necessity for much broader data collection procedures for any future corpora that are developed.

This chapter introduced the Switchboard and NICT-JLE corpora for use in incremental disfluency detection. Through a comparative analysis of linguistic and disfluency features, several points of difference between the corpora were found. The Switchboard corpus was found to be more lexically diverse with higher rates of non-repetitious disfluencies, whereas the NICT-JLE corpus has considerably higher disfluency rates overall as well as a higher rate of disfluencies that are nested and co-occur with edit terms. The frequency of such features was shown to be compounded by reparandum phrase length, with the correlation between edit term frequency being particularly high for the NICT-JLE corpus. Furthermore, the NICT-JLE corpus contains a high rate of learner errors both inside and outside of disfluencies and a wider variability of disfluency distribution due to the impact of learner proficiency on disfluency behaviour. Through an exploration of how these differences may impact model performance, five features of interest were identified: reparandum length, nesting, non-repetitious disfluencies, disfluencies with edit terms and learner errors. These features can be used as starting points for investigating the suitability of existing incremental disfluency detection models adapted for learner speech.

Chapter 4

Establishing a Baseline Model

With a solid grounding in the differences between the disfluency behaviour seen in the Switchboard and NICT-JLE corpora as well as potential points of difficulty for models trained on learner speech, the subsequent aim of this research is to establish a suitable baseline model for experimentation. In order to achieve this, two state-of-the-art incremental disfluency detection models are trained and tested on both the NICT-JLE and Switchboard corpora. An error analysis of the models' outputs is carried out, exploring how the difficult-to-detect cases identified in the previous chapter (reparandum length, nesting, non-repetitious disfluencies, disfluencies with edit terms and learner errors) impact model performance. Due to its overall higher performance on the NICT-JLE corpus as well as its lower number of false positive classifications of learner errors as disfluencies, an LSTM sequence-labelling model with HMM decoder was selected as the baseline model for further experimentation.

4.1 Introducing the Models

The Strongly Incremental Repair detection (STIR) model [172, 86] achieves current state-of-the-art performance for incremental detection on the Switchboard corpus. The STIR model follows a noisy channel approach to disfluency detection following [120, 173], where disfluencies are treated as a deviation from an underlying 'fluent' phrase. Components of disfluencies (reparandum start, interregnum, repair start and repair end), as well as edit terms outside of disfluency phrases, are independently detected using a pipeline of Random Forest classifiers, where each classifier has 20 trees with a maximum depth of four nodes, its

Table 4.1: *The incremental sequence labelling process adopted by the STIR and DEEP models, demonstrated with the example sentence "His professor felt very uh very happy".*

time step	input trigram	incremental label output
1	`<s> <s> his`	his
2	`<s> his professor`	his professor
3	`his professor felt`	his professor felt
4	`professor felt very`	his professor felt very
5	`felt very uh`	his professor felt very {uh}
6	`very uh very`	his professor felt [very {uh} very
7	`uh very happy`	his professor felt [very {uh} very] happy

own error function and specific combination of information-theoretic input features. These features are derived from trigram language models for both words and POS tags and trained on a 'clean' version of the Switchboard training set, where all disfluencies and edit terms have been removed. Weighted Mean Log probability [199], entropy and Kullback–Leibler divergence are some examples of the features implemented in the individual classifiers (see supplementary materials in [86] for an overview of all of the features).

Following on from work on RNNs for incremental disfluency detection [174], the DEEP Disfluency (DEEP) model [175] uses an LSTM architecture combined with an HMM decoder for sequence labelling and achieves close to state-of-the-art results on the Switchboard corpus. Modelled as a joint task with utterance segmentation, the DEEP model uses trigrams of gensim word2vec embeddings [200] of length 50 (trained on a 'clean' version of the Switchboard training set following the same approach as the STIR model), POS tags and word timings (time stamps indicating the start and end of each word) as input features to label reparandum starts, repair starts and edit terms. The network has a hidden layer of 50 nodes and an output layer of size 10. Negative log likelihood (NLL) is used as the cost function and stochastic gradient descent is used over the parameters and word embeddings. The learning rate is set to 0.005 and L2 regularisation is applied with a weight of 0.0001. Viterbi decoding is used on the LSTM softmax output layer combined with timings windows of each word, following [201]. This hand-crafted Markov model is used to eliminate any illegal tag sequences from the LSTM output in the final tag sequence.

Both models process sequences incrementally in a window of nine words which encap-

sulates the repair start and the eight words prior. Features from the trigram $w_{i-2}...w_i$ are used as inputs to the models. The outputs are updated incrementally with the models' best hypotheses as each new word in the sequence is processed. Table 4.1 shows how an example sentence would be correctly labelled at each time step. The trigrams at time steps one and two contain padding tags at the beginning of the sentence. Up until time step four, all words are labelled as fluent, with the first non-fluent word classified at time step five for the edit term "*uh*". Time step six is the first point in the utterance that indicates a possible disfluency with the repetition of the word "*very*". It is at this point that the model should label the repair start and 'look-back' for the reparandum start. For the STIR model, this is achieved through local detection with back-tracing, where the repair start classifier precedes the reparandum start classifier in the pipeline. If a repair start is classified, the system then performs a backwards search of up to eight words and labels the most likely position of the reparandum start. For the DEEP model, there is no manual look-back process. The position of the repair start is instead encoded directly in the tag set and labelled as `repair start-n` where `n` denotes the location of the reparandum start relative to the repair start—in the case of the example in Table 4.1 it would be `rps-1` as interregna are not counted in this labelling process. After both the repair start and reparandum start have been identified, the subsequent words after the repair start are processed in order to identify the repair end. For the STIR model, this is done using a final repair end classifier in the pipeline. For the DEEP model, the repair end is tagged during the Viterbi decoding process.

Aside from the similarities outlined above in terms of model inputs, outputs and labelling conventions, the STIR and DEEP models take markedly different approaches to disfluency detection. In its separate modelling of each part of a disfluency phrase, STIR takes an individualised approach to detection with specific information theoretic input features separately tailored for each classifier. The DEEP model takes a broader approach, with one classifier to label the disfluency phrase as a whole using word embeddings and POS tags as generic input features. The way that each model ensures the elimination of illegal tag sequences also varies: the STIR model controls this through the order of the classification pipeline whereas the DEEP model uses the HMM decoder to find the best legal sequence of tags.

4.2 Testing the STIR and DEEP Disfluency Detection Models

The aim of this experiment was to understand how the STIR and DEEP models perform with the NICT-JLE corpus compared to the Switchboard corpus in order to *(i)*, identify the better performing model for L2 incremental disfluency detection, *(ii)*, test presumptions about model performance outlined in Chapter Three and subsequently *(iii)*, identify areas for further improvement to the chosen model for testing in later chapters. This section first outlines the methodology followed to fulfil these aims. Following this, the results from testing are presented and discussed thereafter, taking the attributes of both the models and corpora as well as previous research into account. Finally, implications for further experimentation are identified.

4.2.1 Methodology

To ensure a fair comparison between model outputs, both the models and the corpora selected for this experiment required adaptation. The NICT-JLE corpus is provided as individual transcripts with html-style tags of 1281 English learners' oral proficiency tests. These transcripts were processed in order to match the formatting and labelling conventions used in the Switchboard corpus. Firstly, all tags relating to learner profile details (such as age, gender, nationality, proficiency level and time spent in an English-speaking country) were removed. Additionally, all tags relating to interview structure (pre-defined task descriptions and task order) were also removed. Finally, any utterances made by the examiner, utterances that contain Japanese, and utterances that had redacted parts for anonymisation purposes were removed, as well as all paralinguistic information such as 'laughter', 'sigh' and non-verbal actions such as 'points' or 'shakes head'.

The Switchboard and NICT-JLE corpora differ in both formatting style and disfluency labelling approach. In the NICT-JLE corpus, only the reparandum part of the disfluency is labelled for detection (i.e. reparandum start, middle and end) and edit terms and interregnum are not differentiated. In the Switchboard corpus all parts of the disfluency phrase are labelled (i.e. reparandum start, middle and end, repair start, middle and end, interregnum and edit terms outside of disfluency phrases). Due to these differences, the tags for both datasets were adapted for this experimentation. The revised tag set includes reparandum start, middle

Example Sentence 1

	his	*professor*	*felt*	[*very*	{*uh*}	*very*]	*happy*
Word Index	0	1	2	3	4	5	6
Switchboard	f	f	f	rms id=5	i id=5	rps id=5 rpn id=5	f
NICT-JLE	-	-	-	R	F	-	-
Revised tags	f	f	f	rms id=5	i id=5	rps id=5	f

Example Sentence 2

	[*I*	*can't*	*do*	{*ah*}	[*I*	*I*]	*couldn't*	*do*]	*my*	*work*
Word Index	0	1	2	3	4	5	6	7	8	9
Switchboard	rms id=4	rm id=4	rm id=4	i id=4	rps id=4 rms id=5	rp id=4 rps id=5 rpn id=5	rp id=4	rpn id=4	f	f
NICT-JLE	SC	SC	SC	F	R	-	-	-	-	-
Revised tags	rms id=4	rm id=4	rm id=4	i id=4	rps id=4 rms id=5	rps id=5	f	f	f	f

Figure 4.1: *Examples of the tagging approaches used in the Switchboard and NICT-JLE corpora as well as the final revised tag set for the sentences "His professor felt very uh very happy" (repetitious disfluency) and "I can't do ah I I couldn't do my work" (nested disfluency), taken from the NICT-JLE corpus.*

and end and repair start labels only. It also separates interregnum and edit terms. Here, hesitation fillers such as "you know" and "I mean" are also labelled as either interregnum or edit terms. Figure 4.1 provides an example of how disfluencies are tagged using this revised tag set alongside the original tags used in the Switchboard and NICT-JLE corpora. Two examples of disfluencies are given, the first being a repetitious disfluency and the second being a nested disfluency. The tag formatting of the revised set was adapted to match the Switchboard corpus as closely as possible, as the code written for the STIR and DEEP models was already compatible with this corpus. Words that are not part of the disfluency structure are labelled as `f` (fluent). Interregnum are labelled as `i` and edit terms outside of disfluencies are labelled as `e`. Reparandum phrase starts are labelled as `rms` and words that are either in the middle or end of a reparandum phrase are labelled as `rm`. Repair phrase starts are labelled as `rps`. Additionally, each disfluency instance is assigned an identification number (ID) relating to the positional word index of the repair start. As shown in Example Sentence 2 in Figure 4.1, words can be assigned multiple tags in the case of nested disfluencies; the word "I" at index position four inside the repetitious repair indicates the repair phrase start for the outer disfluency instance as well as the reparandum phrase start for the inner repetitious disfluency instance.

To adapt the Switchboard corpus to this revised tag set, all instances of repair phrase middle (`rp`) and repair phrase end (`rpn`) tags were removed from the corpus and replaced with fluent tags where necessary. The NICT-JLE corpus required a higher degree of processing and re-formatting. Firstly, edit terms (originally labelled as `F`, meaning 'filler', in the NICT-JLE corpus) were further delineated as either interregnum (`i`) or edit terms (`e`) and all other non-disfluent words were labelled as fluent (`f`). Reparandum phrases (labelled in the NICT-JLE corpus as either `R` for repetitions or `SC`, meaning 'self-correction', for disfluencies) were re-labelled as either `rms` or `rm`. Repair phrase starts (`rps`) were added as a new label. ID numbers for all instances of disfluencies were also added.

Finally, the adapted NICT-JLE dataset was tokenized and subsequently tagged with parts of speech using Stanford's left3words MaxentTagger [202]. Following the same conventions as the Switchboard corpus, contracted forms of words such as "I've" and "can't" were split by tokenization and re-merged after POS tag labelling to form compounded POS tags. In the case of the two examples, the POS tags for these words are `PRPVBP` and `MDRB`, respectively.

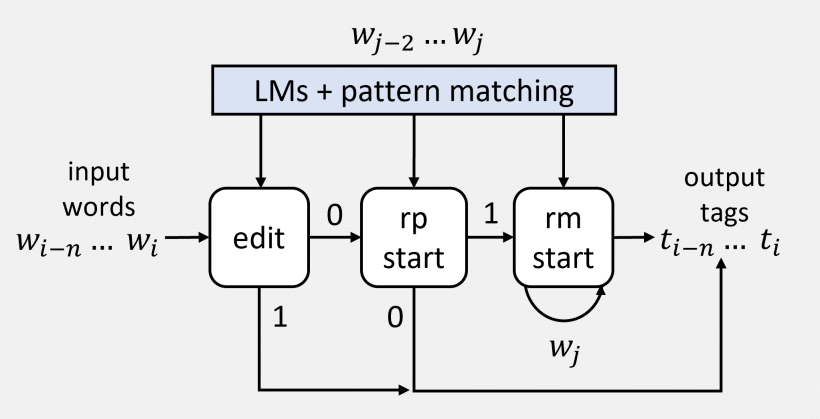
The corpus was split with 80% of the data for training, 10% for heldout and 10% for testing, matching the structure of the Switchboard corpus. In light of the impact of learner proficiency level on disfluency behaviour explored in Chapter Two, the speech data were balanced across the train, heldout and test sets to ensure equal distributions of SST proficiency levels. Additionally, all speech data in the test set were restricted to files that have learner error tags to allow for future analysis. See Table 3 and Table 4 in the Appendix for an overview of the disfluency metrics and proficiency level distribution associated with each set. The procedures described above resulted in a significantly adapted version of the NICT-JLE corpus. As noted in Chapter One, with the addition of POS tags, a wider set of disfluency labels, equal distribution of data according to learner proficiency and the restriction of error-tagged data to the test set, the adapted corpus is not only more closely aligned with the Switchboard corpus but also better optimised for automatic L2 disfluency detection research. Further adaptions to the NICT-JLE corpus relating to the paralinguistic and meta-features of the corpus are described in Chapter Five.

Based on the changes described above, both the STIR[1] and DEEP[2] models were adapted from their original source code. For the DEEP model, the 'disfluency-only' model (without utterance segmentation) was used. In addition, the HMM decoder was updated in two ways: timings were removed as an input option to the decoder as the NICT-JLE corpus does not provide timing information and repair end tags were removed from the final tag set defined for the HMM. For the STIR model, the repair end classifier was removed from the pipeline. The structure of the adapted models are depicted in Figure 4.2. For experimentation, the adapted versions of the STIR and DEEP models were trained and tested on the adapted Switchboard and NICT-JLE corpora individually. All of the model parameters described in Section 4.1 were kept the same. To ensure replicable results, the same random seed was used for all data preparation (word embeddings, language models etc.) and training procedures.

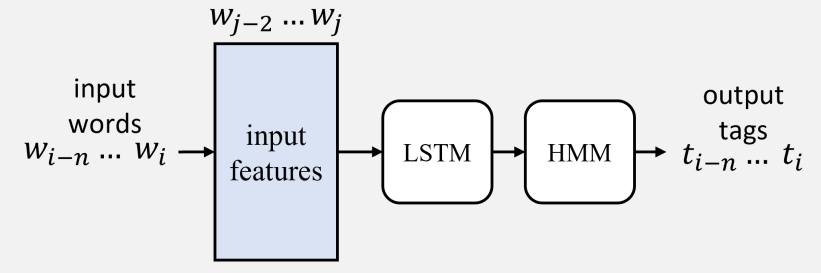
4.2.2 Results

Table 4.2 depicts the overall performance of the STIR and DEEP models trained and tested on the Switchboard and NICT-JLE corpora. Following the standard approach taken for

[1] https://bitbucket.org/julianhough/STIR/src/master/
[2] https://github.com/clp-research/DEEP_disfluency



(a) *STIR model*

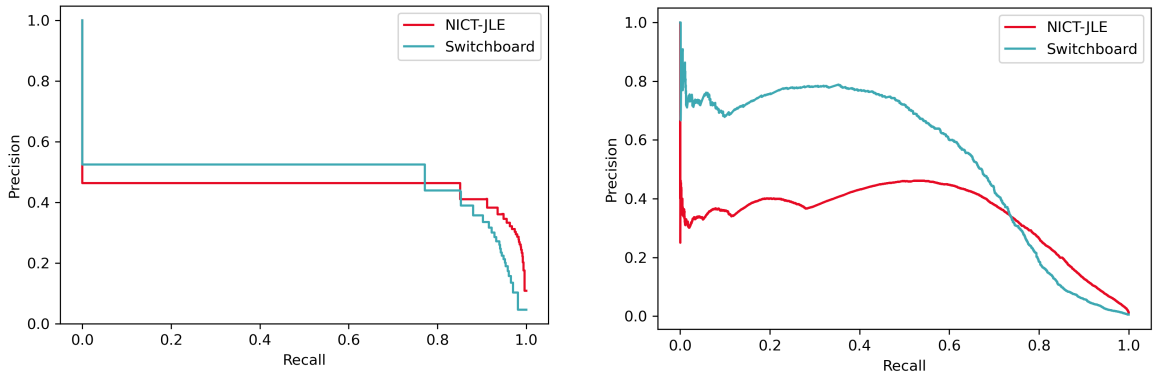


(b) *DEEP model*

Figure 4.2: *Diagrams of the adapted STIR and DEEP model structures used for experimentation.*

Table 4.2: *Precision, recall and F-score results for repair start (rps), reparandum phrase (rm) and edit term (e) detection for the STIR and DEEP models, trained and tested on the Switchboard and NICT-JLE corpora.*

	STIR Models						DEEP Models					
	Switchboard			NICT-JLE			Switchboard			NICT-JLE		
	rps	*rm*	*e*	*rps*	*rm*	*e*	*rps*	*rm*	*e*	*rps*	*rm*	*e*
Precision	0.69	0.67	0.94	0.63	0.57	0.99	0.83	0.75	0.93	0.72	0.65	0.98
Recall	0.79	0.75	0.93	0.81	0.74	0.99	0.65	0.59	0.90	0.76	0.69	0.91
F-score	0.73	0.70	0.94	0.71	0.65	0.99	0.73	0.66	0.91	0.74	0.67	0.95

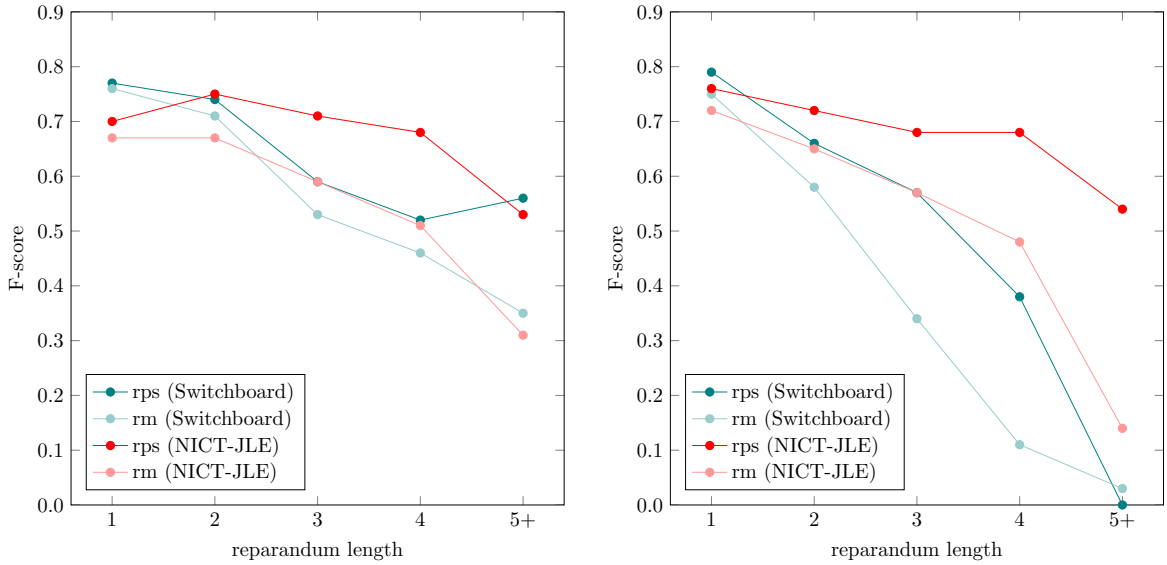


(a) *STIR models* **(b)** *DEEP models*

Figure 4.3: *Precision-Recall curves for repair start detection for the STIR and DEEP models trained and tested on the Switchboard and NICT-JLE corpora.*

evaluating incremental disfluency detection, the models' performance is measured in terms of precision, recall and F-score for repair start (*rps*), reparandum phrase (*rm*) and edit term (*e*) detection, with utterance final labels reported. The DEEP model has the highest scores for *Frps* and *Frm* detection in the NICT-JLE corpus (0.74 and 0.67, respectively) compared to the STIR model (0.71 and 0.65). Looking more closely, it can be seen that this is a result of the STIR model's imbalanced recall and precision scores, with considerably lower figures for the latter compared to the DEEP model. In regard to edit terms, the STIR model has the best detection rates for the NICT-JLE corpus with an F-score of 0.99. Across both models and datasets, *Frps* consistently outperforms *Frm* and the gap between *Frm* and *Frps* detection is narrower for the STIR models, especially for the Switchboard corpus.

Figure 4.3 shows the Precision-Recall (PR) curves for each of the STIR and DEEP models trained and tested on the Switchboard and NICT-JLE corpora. As described above, the



(a) *STIR models*

(b) *DEEP models*

Figure 4.4: *F-score results for repair start (rps) and reparandum phrase (rm) detection according to reparandum phrase length for the STIR and DEEP models, trained and tested on the Switchboard and NICT-JLE corpora.*

STIR models use binary classification to detect repair start and reparandum start whereas the DEEP model uses a multiclass classification approach to label the whole disfluency. With these labelling approaches, only the repair start labels have associated model probability scores available for PR curve calculation and so reporting is restricted to repair phrase start detection. As the calculations for PR curves are based on the presumption of binary classification, the PR curves reported for the DEEP model are the result of micro-averaging the prediction outputs for the eight classes related to repair start (`rps-1`, `rps-2` and so on) excluding the fluent and edit term classes. As seen in the figure, the STIR models show consistent performance across the native and learner speech domains. The STIR models skew towards higher recall, with the imbalance between recall and precision slightly higher for the NICT-JLE corpus, reflecting the results above in Table 4.2. The PR curves for the DEEP models show different outcomes depending on the corpus domain also reflected in the above results in Table 4.2, with higher precision scores across thresholds for the Switchboard corpus compared to the NICT-JLE corpus.

Table 4.3: *F-score results for repair start (Frps) and reparandum phrase (Frm) detection of the STIR and DEEP models, for disfluency instances with reparandum phrase lengths of two or more.*

	STIR Models				DEEP Models			
	Switchboard		NICT-JLE		Switchboard		NICT-JLE	
	Frps	*Frm*	*Frps*	*Frm*	*Frps*	*Frm*	*Frps*	*Frm*
length 2+	0.66	0.60	0.72	0.61	0.60	0.46	0.70	0.59

How does reparandum phrase length impact the detection of disfluency instances?

Figure 4.4 compares *Frps* and *Frm* for the models tested on both corpora according to reparandum length. Figure 4.4a shows the results of the Switchboard and NICT-JLE corpora trained and tested on the STIR models and Figure 4.4b shows the same for the DEEP models. As the graphs show, in general, model performance decreases as reparandum length increases. In addition, reflecting the overall results described above, *Frps* consistently outperforms *Frm*. The performance gap between these two measures generally increases with reparandum length and is particularly detrimental for the DEEP models. Overall, the STIR models are the least impacted by reparandum length and with the exception of one word reparanda, the models trained using the NICT-JLE corpus generally show higher results across lengths. Performance is particularly poor for the DEEP model trained on the Switchboard corpus, showing the steepest decline in performance and particularly low scores for *Frm* detection. Table 4.3 summarises this data further and shows the F-score performance for all disfluency instances with reparandum phrases that are two words or longer. As can be seen, for both corpora, the STIR models have higher F-scores.

How does disfluency type (repetitious vs. non-repetitious) impact the detection of disfluency instances?

Table 4.4 summarises the performance results across models and corpora for repetitious and non-repetitious disfluency instances. As the model tags do not differentiate between repetitious and non-repetitious disfluencies in the adjusted NICT-JLE corpora, precision could not be calculated, and therefore only recall scores are reported. Reflective of the overall results above, the STIR models have higher recall scores for both conditions. The

Table 4.4: *Recall results for repair start (rps) and reparandum phrase (rm) detection of the STIR and DEEP models, for repetitious and non-repetitious disfluency instances.*

	STIR Models				DEEP Models			
	Switchboard		NICT-JLE		Switchboard		NICT-JLE	
	Rrps	*Rrm*	*Rrps*	*Rrm*	*Rrps*	*Rrm*	*Rrps*	*Rrm*
repetition	0.98	0.96	0.94	0.89	0.81	0.79	0.86	0.83
non-repetition	0.59	0.53	0.68	0.59	0.47	0.37	0.63	0.52

Table 4.5: *F-score results for repair start (Frps) and reparandum phrase (Frm) detection of the STIR and DEEP models, for nested and non-nested disfluency instances.*

	STIR Models				DEEP Models			
	Switchboard		NICT-JLE		Switchboard		NICT-JLE	
	Frps	*Frm*	*Frps*	*Frm*	*Frps*	*Frm*	*Frps*	*Frm*
nested	0.74	0.65	0.68	0.56	0.72	0.59	0.65	0.52
non-nested	0.73	0.71	0.70	0.66	0.73	0.68	0.73	0.67

recall rates of repetitious instances are considerably more successful than the recall of non-repetitious instances across both models and corpora. This is especially true for the STIR models, with reparandum start recall reaching 0.98 for the Switchboard corpus and 0.94 for the NICT-JLE corpus. For non-repetitious disfluencies, the models trained on the NICT-JLE corpus outperform those trained on the Switchboard corpus.

How does disfluency nesting impact the detection of disfluency instances?

Table 4.5 reports the impact of nesting on the disfluency detection F-scores across the models and corpora. As can be seen, models show a higher recall for non-nested disfluency instances across both corpora. The degradation of detection performance for *Frps* is minimal between the nested and non-nested instances in all cases apart from the DEEP model trained on the NICT-JLE corpus which shows a considerable gap (0.73 and 0.65 for non-nested and nested, respectively). When considering *Frm*, the detection of non-nested instances is higher than nested instances across all models and corpora, most significantly so for models trained on the NICT-JLE corpus.

Table 4.6: *F-score results for repair start (Frps) and reparandum phrase (Frm) detection of the STIR and DEEP models, for with-edit and without-edit disfluency instances.*

	STIR Models				DEEP Models			
	Switchboard		NICT-JLE		Switchboard		NICT-JLE	
	Frps	*Frm*	*Frps*	*Frm*	*Frps*	*Frm*	*Frps*	*Frm*
with-edit	0.75	0.70	0.77	0.69	0.65	0.54	0.72	0.62
without-edit	0.73	0.70	0.69	0.63	0.75	0.69	0.74	0.69

Table 4.7: *Comparing edit term recall with the average recall performance gap for repair start and reparandum phrase detection showing that as edit term recall declines, the performance gap between with-edit and without-edit disfluency instances increases.*

	STIR Models		DEEP Models	
	NICT-JLE	Switchboard	NICT-JLE	Switchboard
performance gap	0.02	0.06	0.09	0.10
edit term recall	0.99	0.94	0.91	0.87

How do interregna impact the detection of disfluency instances?

Table 4.6 shows the impact of interregna on the detection F-scores of disfluency instances across models and corpora. As can be seen in the results, the presence of interregna impacts the STIR and DEEP models in opposite ways. For the STIR models, the disfluencies with interregna have either the same (for in the Switchboard corpus) or a higher F-score than those without. The difference is especially noticeable for the NICT-JLE corpus, (0.77 *Frps* and 0.69 *Frm* for with-edit compared to 0.69 *Frps* and 0.63 *Frm* for without-edit). When considering the DEEP model, however, the performance for disfluency instances with interregna is lower, with the impact being most prevalent for the Switchboard corpus (0.65 *Frps* and 0.54 *Frm* for with-edit compared to 0.75 *Frps* and 0.69 *Frm* for without-edit). Table 4.7 shows the average performance decrease (the average of *Frps* and *Frm* recall score for the with-edit group minus the average of *Frps* and *Frm* recall score for the without-edit group) for each model dataset combination in ascending order. Also reported is the associated overall edit term recall score for each model combination. As can be seen, there may be a correlation between edit term recall and the influence of interregna on disfluency instance detection—the higher the recall of edit terms, the smaller the degradation of performance for with-edit disfluency instances.

Table 4.8: *F-score results for repair start (Frps) and reprandum phrase (Frm) detection of the STIR and DEEP models for disfluency instances preceded by a learner error.*

	STIR Model		DEEP Model	
error position	*Frps*	*Frm*	*Frps*	*Frm*
rps only	0.68	0.63	0.72	0.63
rps-1 only	0.40	0.36	0.49	0.45
rps and *rps-1*	0.50	0.45	0.59	0.52
any *rps-1*	0.46	0.41	0.55	0.49

How do errors in learner speech impact the detection of individual disfluency instances?

Table 4.8 reports the impact of learner errors in the NICT-JLE corpus on the DEEP and STIR models' detection rates. Four conditions of disfluency instances are reported: those that co-occur with a learner error at repair start only, those that co-occur with a learner error at one word prior to the repair start only, those that co-occur with errors at both repair start and one word prior to repair, and those that co-occur with an error at one word prior to repair start. It can be seen from the results that the co-occurrence of learner errors with disfluency instances impacts detection performance across both models, especially for instances where there is an error immediately prior to the repair start. The STIR model is particularly impacted, with F-score as low as 0.36 for reparandum phrase detection when co-occurring with a learner error immediately prior to repair onset. Disfluency instances that co-occur with learner errors at repair start are the least impacted, however, the performance remains lower than the overall performance reported in Table 4.2.

4.2.3 Discussion

To summarise the above results, the DEEP model shows the best performance for the NICT-JLE corpus, outperforming the STIR model as a result of its low precision. In addition, models trained on the NICT-JLE corpus glean higher edit term detection accuracy compared to those trained on the Switchboard corpus, although all model combinations score highly. Reparandum length was shown to have a negative impact on model performance, however, the severity of the impact was less for the models trained on the NICT-JLE corpus. Results

confirmed that non-repetitious and nested disfluency instances prove harder to detect for both models, whereas the accurate detection of instances with-interregna was model-dependent. Finally, the presence of learner errors in the NICT-JLE test set impacts model performance, for both the STIR and DEEP models, with the STIR model output suffering to a higher extent. The causes and implications of these results are discussed in further detail below.

First looking at the overall performance of the models and corpora, one contributing factor as to why the DEEP model shows the best performance for the NICT-JLE corpus may be dataset size. As explored earlier in Chapter Three, NICT-JLE is the larger corpus, with over three times as many instances of disfluency compared to the Switchboard corpus. The impact of this is particularly evident in the results for the DEEP models, where the recall scores for repair start and reparandum phrase detection for the NICT-JLE corpus are considerably higher than those for the Switchboard corpus (0.76, 0.69 and 0.65, 0.59, respectively). The impact of the NICT-JLE corpus' larger dataset may be compounded further by the fact that it has a lower lexical richness compared to the Switchboard corpus. With a higher number of instances and smaller variation to learn from in the NICT-JLE corpus, it may be the case that disfluency instances more closely resemble one another, making it easier for a model to generalise. This may also explain why despite the fact that the STIR model trained on the Switchboard corpus is the better performing model overall, the recall results are higher for the NICT-JLE corpus than the Switchboard corpus in the STIR models.

Looking next to precision in the overall results, the poor outcomes for the STIR model tested on the NICT-JLE corpus may be in part to do with the relationship found between false positive classification and learner errors in the NICT-JLE corpus. As described in Chapter Three, the STIR model approach relies on the presumption that the underlying phrase of a disfluency is 'fluent'. The language models that are used for probability-based feature extraction are trained on 'clean' datasets without disfluencies, with the idea that any repair start encountered by the model will output low probability scores making them easy to detect. This approach is likely very effective for error-type disfluency instances in the NICT-JLE corpus, and this may be another contributing factor as to why the recall scores for the STIR model trained on the NICT-JLE corpus are so high. However, as explored previously, there are instances of learner errors in the NICT-JLE corpus that do not go on to be repaired. In the STIR model, these errors will also produce low probability and

will therefore be more likely to be incorrectly identified as part of a disfluency phrase. The particularly poor performance of the STIR model when detecting disfluency instances that co-occur with learner errors supports this theory. Based on these findings, it seems that the noisy-channel model approach to disfluency detection adopted by the STIR model may not be suitable for learner data. It is worth noting that the DEEP model also shows poor performance in this area but to a lesser degree than the STIR model. Although the DEEP model does not use probabilistic language model features explicitly, it does use trigrams of word embeddings which likely encode some semantic information into the model. In addition, during validation and testing, the embedding vectors for previously unseen words are initialised at random. Learner errors that result in non-existent words such as "*wented*" are likely to be treated as unseen words and as a consequence exacerbate the problem.

The structured look-back system that is implemented in the STIR models is a likely factor as to why they are the better performing models across reparandum lengths. As also discussed in [175], this approach outperforms that of left-to-right LSTMs, which generally struggle with labelling longer phrases as a consequence of the vanishing gradient problem. In addition, because the DEEP model does not classify edit terms separately as STIR does—which subsequently removes them from the classification pipeline—these edit terms add extra length to the utterance, further exacerbating the issue of diminished detection accuracy over longer phrases. When comparing the corpora performances, reparandum length is less of an issue for the NICT-JLE corpus. These results contradict the predictions made in Chapter Three, which supposed that disfluency instances with long reparandum phrases may be more detrimental to detection in the NICT-JLE corpus due to them being more likely to occur with difficult-to-detect disfluency features. These results instead provide further evidence to support the above theory correlating reduced dataset lexical richness with higher disfluency detection rates.

The results for repetitious and non-repetitious disfluency instances reflect that of prior research discussed in Chapter Two, showing that repetitious disfluency instances are easier for models to detect in general, especially for models such as STIR that use language model probabilities as input features. Looking at the non-repetitious instances, as discussed in Chapter Three, the lower rate of such instances seen in the NICT-JLE corpus may be a contributing factor as to why detection performance for such instances was higher compared

to the Switchboard corpus.

Looking at the impact of edit terms on disfluency detection, results showed that their usefulness is model-dependent. For the STIR model, disfluency instances with interregna perform marginally better on the Switchboard corpus and significantly better on the NICT-JLE corpus. These results suggest that for the STIR model, high rates of edit terms in training data like that seen in the NICT-JLE corpus not only lead to very high edit term detection performance in general (as seen in the overall results in Table 4.2) but also to improved disfluency detection, especially for repair start. The opposite effect is true for the DEEP models. The impact is lesser for the NICT-JLE corpus compared to the Switchboard corpus, again potentially due to the higher rate of edit terms seen during training, but generally speaking, the DEEP models show a significant reduction in reparandum phrase detection for both datasets. This adds further to the discussion above relating to the impact of edit terms and the DEEP model performance as reparandum length increases. From these results, it is clear that the handling of edit terms is an area of interest for this work and warrants further experimentation.

The outcomes for nested disfluency instances confirm their negative impact on model performance as was hypothesised in Chapter Three. The results show that this is especially the case for reparandum phrase detection. The poor performance for nested disfluency detection in the NICT-JLE corpus may be a reflection of its higher frequency of such disfluency instances, exacerbated by their co-occurrence with non-repetitious and with-interregna instances. However, the impact is not as severe for the STIR model tested on the NICT-JLE corpus. As discussed above, the look-back classification approach used in the STIR model may be the reason for this, especially with the knowledge that the inclusion of edit terms leads to STIR model performance. These results highlight the need to focus on reducing the impact of nesting in future experimentation.

The analysis above reveals the strengths and weaknesses of both models when trained and tested using learner speech. Thanks to its individual classifiers and 'look-back' tagging pipeline, the STIR model is particularly successful in disfluency instances closely related to reparandum phrase detection: nested and with-edit instances as well as instances with longer reparandum phrases. With its broader approach to classifying disfluency phrases as a whole, the DEEP model is better able to handle learner errors. The particularly poor

results of the STIR model in this area suggest that noisy-channel approaches are not suitable for learner speech with high rates of learner errors. Furthermore, despite the STIR model outperforming the DEEP model on measures of disfluency length, with-edit, nested and non-repetitious repairs, the DEEP model shows the highest overall performance for disfluency detection with the NICT-JLE. This result suggests that a model's ability to handle learner errors has a higher impact on overall performance than the other difficult-to-detect cases and as such should be a strong focus of further inquiry. Finally, lexical complexity seems to play a role in model performance across all disfluency types, with reduced lexical complexity showing to be particularly advantageous in the DEEP model.

In terms of the models' F-scores for disfluency detection (i.e. reparandum phrase and repair phrase start detection), the DEEP model showed the highest performance on the NICT-JLE corpus and was therefore chosen as the baseline model for further experimentation in Chapter Five. Beyond its higher performance, there are other benefits to using the DEEP model as a baseline for adaptation. Firstly, the architecture of the model is much simpler than that of STIR. With one classifier (plus an HMM decoder) and one set of input features, modifications can be implemented much more efficiently. For example, this could include the addition of further input features or integration of beneficial aspects of the STIR model, such as the high-accuracy edit term detection module. An additional benefit is that such a model, like many other sequence-tagging approaches outlined in Chapter Two, can easily be extended for further research endeavours to other joint tasks pertinent to dialogue processing such as utterance segmentation (as is done in the original version of the DEEP model [175]), POS-tagging and learner error detection.

4.3 Limitations

Although the corpora tested above were adapted to aid comparison, two features remain that may have impacted model performance. The first is POS tags, which have been annotated by hand for the Switchboard corpus and tagged automatically for the NICT-JLE corpus. It is presumed here that automatic tagging, especially when using a tagger that is not adapted for learner speech, is likely to have had a negative impact on the models trained and tested on the NICT-JLE corpus. The second difference is dataset size. As discussed earlier, it

is likely that the NICT-JLE corpus' larger size in part contributed to model performance. However, the differing sizes of the NICT-JLE and Switchboard corpora make it difficult to control for size when considering the effects of other features such as nesting frequency and reparandum length. These factors alongside the comparability issues regarding both task and speaker variation previously discussed in Chapter Three highlight again how the corpora tested here are a proxy for learner and native speech and future analysis would benefit from using data that has been controlled for features such as speaker count, dataset size and activity type. Nonetheless, comparing the performance of models trained and tested on the NICT-JLE corpus with those trained and tested on the Switchboard corpus allowed for a deeper understanding of the effects of the models versus the effects of the data. This in turn facilitated a more precise targeting of the key challenges for incremental disfluency detection adapted for learner speech specifically.

Experiments carried out on two start-of-the-art incremental disfluency detection models found that an LSTM-based model with HMM decoder is the better-suited model for learner speech. Findings from an error analysis carried out on the model outputs suggest that this is due to the NICT-JLE corpus' large size and reduced lexical richness being particularly advantageous in a deep learning setting, as well as the LSTM model's lesser reliance on a noisy channel approach to disfluency modelling, reducing the number of false positives attributed to learner errors in the data. All five features that were previously identified as potential difficult-to-detect cases were confirmed as such for learner speech, three of which (nested, with-edit and with-error instances) have not been explored in previous research. The results also confirmed the interrelated nature of the disfluency features tested and provide several areas for further investigation. All of these features can be considered as challenges to address, however, given their particularly poor results, with-error disfluencies remain the most difficult disfluencies to detect accurately.

Chapter 5

Approaches to Model Adaptation

With the aim of improving the overall performance of the baseline model established in Chapter Four, five approaches to model adaptation are introduced: dataset lemmatization, using paralinguistic features as inputs, experimenting with character and word embeddings, tackling edit term detection prior to disfluency detection and, using learners' English proficiency levels as input features. This is followed by an overview of the methodology followed to test each adaptation. Performance results of the final model as well as each model iteration are analysed in the context of the five difficult-to-detect disfluency cases encountered by the baseline model. An ablation study is also carried out in order to identify the individual impact of each of the adaptations. In addition, the model performance is reviewed according to learners' proficiency level. The model is tested on out-of-domain data and the overall performance of the adapted model is compared to existing research, setting a new state-of-the-art benchmark for incremental disfluency detection for learner speech.

5.1 Opportunities for Model Improvement

5.1.1 Lemmatization

Lemmatization is a text normalisation technique whereby words are reduced to their morphological 'root' (or 'lemma') form. For example, when lemmatized, the words "went", "goes", and "going" are all reduced to the singular present tense verb "go". Normalising the text in this way not only reduces the overall vocabulary size of the text but also lowers its lexical complexity and POS tag variation.

With the results of experimentation in Chapter Three and Chapter Four revealing a possible correlation between dataset lexical complexity and model performance, along with the knowledge that repetitious disfluencies are easier to detect for existing models, it was hypothesised that normalising the NICT-JLE corpus may be a simple solution to improving baseline performance. The advantage of lemmatization over other techniques such as stemming is that the root forms remain whole linguistic units i.e. the lemma will almost always be a real word. This allows for more accurate POS-tagging and word embedding representation when creating the training dataset.

5.1.2 Pauses and Laughter

As discussed in Chapter Two, cues for disfluency come from a variety of sources. Physical movements such as hand gestures [203] and eye gaze [93] as well as prosodic cues such as word cut-offs and sound stretching all contribute to the production of self-repair. The NICT-JLE corpus includes a subset of such features, tagged in the dataset as 'non-verbal sounds'. With the knowledge that several prosodic cues tend to occur immediately prior to repair onset (i.e. the point at which disfluencies are detected in an incremental framework), using such features may prove more beneficial to incremental systems compared to non-incremental approaches. The features explored for this work are pauses in speech, laughter, and also laughed speech.

Incorporating pauses as model input features has shown to be successful in previous disfluency detection research. They have been encoded implicitly as filter bank outputs [184] explicitly with binary indicators of their presence or absence [145, 138], and also generated as predictions from text [161]. The disfluencies of non-native speakers are more likely to contain pauses compared to native speakers [204], suggesting that such pauses may be a more reliable indicator of disfluency in learner speech than native speech. Due to the positive impact of prosodic input features in non-incremental settings outlined above, it seems likely that including pauses as input features will have a similar effect on incremental model performance.

It is well understood that laughter serves a pragmatic function in conversation beyond reacting to something humorous [205]. For example, research has shown laughter to be used as a tool for continuing the ongoing trajectory of talk [206] as well as dealing with embarrassing moments during interviews [207] and doctor-patient interactions [208]. This is also the case for language learners, who use laughter to pre-empt a problematic action

[209] during uncertainty [210] and after making an error [211]. Considering the relationship between learner errors and self-repair and the interview-like nature of the SLT test used in the NICT-JLE corpus, it is reasonable to assume that laughter likely plays an important role for learners during self-repair. In fact, in an analysis of English proficiency interviews for UK universities, laughter was found to co-occur with disfluency in three ways: *(i)*, on its own between the reparandum and repair phrase, *(ii)*, along with other indicators of disfluency such as pauses and partial words, and *(iii)*, as laughed speech during either the repair phrase or the whole disfluency phrase [212]. Laughter has been used as input for dialogue act classification models [213], however, its value as a feature for incremental disfluency detection in learner speech has not yet been explored.

5.1.3 Out of Vocabulary Words

A common challenge when modelling learner language data is the high number of unknown words found in datasets, namely due to the inclusion of transliterated proper nouns and learner errors [194]. An example of the former in the NICT-JLE corpus is "hackikomae", which refers to a famous meeting place in Tokyo in front of Shibuya train station. This is something well-known to those living in Tokyo but may not be likely to appear in pre-trained datasets. Morphological errors made by language learners lead to the presence of non-existent words in learner corpora such as "camed", "advices" and "broked" found in the NICT-JLE corpus. The types of instances described above pose a challenge to word embedding models as infrequent or erroneous words are less likely to have appeared in their training data. As a consequence, words are treated as being 'out of vocabulary and are typically assigned a vector of random values rather than a learned embedding. Although the baseline model used in this work updates embeddings as part of the objective function during training, this is not likely to be as representative as a model that can better capture the similarity between a mistake such as "camed" and the word "came".

Two approaches to this problem are explored here. The first is to compare the performance of using large pre-trained word embedding models with models trained on the NICT-JLE dataset. The baseline model already implements the latter, and it can be argued that doing so may help tackle the unknown word issue, as grammatical errors that occur frequently

such as "camed" would be encoded into the embedding model. However, more recent work in disfluency detection—including those applied to the NICT-JLE corpus [183, 184]—has shown the advantages of using larger embeddings trained on millions of words of text. It is therefore of interest to establish if including such embeddings is also beneficial in an incremental setting or if prioritising the modelling of unknown words leads to better performance. The second approach is the integration of character embeddings into the model. As seen in disfluency detection [183] and POS-tagging [194] for learner data, character-level embeddings are concatenated with word embeddings as another way to handle unknown words. Including characters in this way allows the linguistic similarity between examples such as "camed" and "came" to be more closely modelled.

5.1.4 Edit Terms

Looking at prior disfluency detection research, there are various approaches taken to edit term detection. One approach has been to use the presence of edit terms as a binary input feature in the models [126, 172]. However, other work has commented on the limitations that including edit terms in this way brings in real-world settings, as many pre-built ASR models automatically filter out edit terms [175]. Work on disfluency detection using the NICT-JLE corpus specifically removes edit terms from the dataset altogether, taking the view that edit term detection can be considered a separate task that can be handled prior to disfluency detection [183, 184].

Following the results from experimentation in Chapter Four, it was hypothesised that for the baseline model, it is not necessarily the presence of edit terms that degrades performance but rather the increased distance between repair start and reparandum start for disfluencies with interregna that cause detection difficulties. In fact, the presence of edit terms may be a useful predictor specifically in L2 settings due to the fact that interregna occur almost twice as frequently in the NICT-JLE corpus compared to the Switchboard corpus. Therefore, it seems likely that retaining information about the presence or absence of edit terms *without* including them directly in the LSTM tagging model will be beneficial to overall performance.

5.1.5 Speaking Proficiency Level

As shown in Chapters Two and Three, there are various ways in which a learner's speaking proficiency may impact disfluency behaviour, including disfluency instance structure, disfluency instance frequency, interregnum frequency and learner error frequency. Having established in Chapter Four that with-edit and with-error disfluency instances negatively impact the baseline model performance, it is reasonable to assume that lower proficiency speakers with high rates of disfluency are likely to be more challenging. In addition, the NICT-JLE corpus does not have an equal distribution of speaker proficiencies, with the majority of speakers at the intermediate SST levels four, five and six (see Table 4 in the Appendix for an overview of this distribution). This may cause a bias in the model and negatively impact the disfluency detection performance for beginner and advanced speakers in the corpus. With both of the above observations in mind, it was hypothesised that including speaker proficiency level explicitly as an input feature may help the model to better generalise to the variation described above.

5.2 Testing Model Adaptations

The aim of this experiment was to apply the approaches described above in order to improve the performance of the baseline model established in Chapter Four. The adaptations to the model described below were first tested consecutively in order of appearance, with each new adaptation building on the best performing model of the previous iteration. This was followed by an ablation study, where each adaptation was combined with the baseline model and tested individually. Throughout the tests, the same hyperparameters as the previous experimentation were used, with a learning rate of 0.005, L2 regularisation applied to the parameters with a weight of 0.0001. The random seed for each iteration of the testing was kept the same.

5.2.1 Methodology

The NLTK WordNet Lemmatizer was used for dataset lemmatization [214]. Following the POS tag formatting requirements for the NLTK Lemmatizer, all POS tag variations of nouns, verbs, adjectives and adverbs previously assigned by the Stanford POS-tagger [202] were

manually reduced to one tag per POS category: `NN`, `VB`, `JJ` and `RB`, respectively. Only words with these tags were lemmatized, leading to an approximate 20% reduction in vocabulary size (6181 words reduced from 7590) and a 30% reduction in POS tags (49 tags reduced from 70). The training set of the lemmatized data was then used to create the word embedding model used for input features.

Silence and laughter features were derived automatically from the 'non-verbal sound' tags in the NICT-JLE corpus transcripts and transformed into one-hot vectors suitable for model consumption. Each word was assigned a vector that indicated the presence (1) or absence (0) of a preceding short pause, long pause, non-verbal laughter, or if the current word itself was laughed. Following the approach of the baseline model for words and POS tags, the silence and laughter data for the preceding two words were also included, resulting in a concatenated vector of size 12. In order to find the best combination of these features, this experimentation compared the individual impact of including silence, non-verbal laughter and laughed speech on the model, as well as subsequent combinations of all three features.

For word and character embeddings, three models were explored: GloVe [157], word2vec [158] and fastText [215]. GloVe uses global corpus statistics to generate ratios of probabilities from co-occurrence matrices. The word2vec model follows a skip-gram with negative sampling approach and uses the weights of a classifier trained to predict the likelihood of two words occurring near each other in a text. Finally, fastText is an extension of word2vec which is better able to handle unknown words thanks to the inclusion of n-gram subword models. For this experimentation, pre-trained versions of each model with embedding sizes of 300 are compared with word2vec and fastText models trained on the NICT-JLE training set with embedding sizes of 50, 100 and 200. The pre-trained embedding size was set to 300 because this was the only embedding size that was publicly available across all three models. It was decided to start the embedding size of the dataset-trained models at 50 which is the original size used for the baseline model, matching that of previous research [175].

In order to remove edit terms from the LSTM part of the model, an edit term classifier was added to the front of the pipeline. The edit term classifier implemented in the STIR model from Chapter Four was used, due to its high detection F-scores for the NICT-JLE corpus (0.99). Figure 5.1 pictures the updated structure of the model. Instead of including edit terms in the LSTM, the edit term classification is isolated by detecting and subsequently

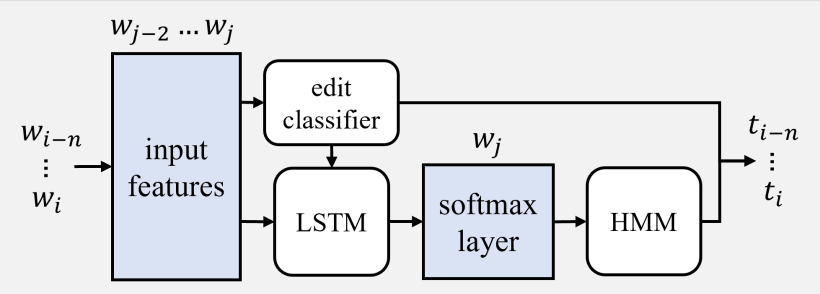


Figure 5.1: *Diagram of the adapted DEEP model structure used for experimentation.*

Table 5.1: *Precision, recall and F-score results for repair start (rps) and reparandum phrase (rm) detection for the baseline model and subsequent model adaptations. Each adaptation is added incrementally, including all preceding model iterations.*

Model iteration (added incrementally)	Repair phrase start (*rps*)			Reparandum phrase (*rm*)		
	P	R	F	P	R	F
baseline	0.72	0.76	0.74	0.65	0.69	0.67
+ lemmatization	0.72	0.81	0.76	0.65	0.74	0.69
+ pauses & laughter	0.72	0.80	0.76	0.66	0.74	0.70
+ char embeddings	0.74	0.80	0.77	0.68	0.73	0.71
+ edit terms	0.75	0.82	0.78	0.70	0.76	0.73
+ learner level (final model)	0.76	0.81	0.79	0.71	0.76	0.74

removing edit terms from the pipeline. The output from the classifier is fed to the LSTM as a one-hot vector indicating the presence or absence of a preceding edit term for the current word and the two words prior. Removing the edit terms in this way reduces the tag-set of the LSTM and HMM from ten to nine. Edit terms are then fed back into the pipeline for the final incremental detection output. This experimentation tested two models: one which receives the edit term classifier output and one which does not.

Proficiency levels were derived from the NICT-JLE transcripts, represented as one-hot vectors of size nine, reflecting the number of levels. Due to the more 'global' nature of these features, relating to speaker features as opposed to isolated word-by-word features, these vectors did not include data about the previous two words. For evaluation, the model was tested with and without the presence of proficiency level features.

Table 5.2: *Performance results for repair start (rps) and reparandum phrase (rm) detection for difficult-to-detect cases across all model adaptations. Each adaptation is added incrementally, including all preceding model iterations.*

Model iteration (added incrementally)	length 2+ F-score	non-rep Recall	nested F-score	with-edit F-score	with-error F-score
			Repair phrase start (*rps*)		
baseline	0.70	0.63	0.65	0.72	0.55
+ lemmatized	0.70	0.79	0.68	0.74	0.59
+ pause & laughter	0.71	0.79	0.69	0.75	0.59
+ char embeddings	0.73	0.79	0.69	0.75	0.60
+ edit terms	0.75	0.81	0.73	0.80	0.59
+ learner level (final model)	0.76	0.80	0.73	0.79	0.60
			Reparandum phrase (*rm*)		
baseline	0.60	0.52	0.52	0.62	0.49
+ lemmatized	0.59	0.72	0.52	0.63	0.53
+ pause & laughter	0.61	0.72	0.53	0.64	0.53
+ char embeddings	0.62	0.72	0.55	0.64	0.53
+ edit terms	0.65	0.75	0.58	0.72	0.53
+ learner level (final model)	0.66	0.75	0.58	0.71	0.54

5.2.2 Results

Table 5.1 shows the precision, recall and F-score results for repair start and reparandum phrase detection of the baseline model and the subsequent best performing models for each of the testing iterations described above. For a full overview of the results for all testing iterations, see Table 9 in the Appendix. As the results show, with each adaptation the overall F-scores for repair start and reparandum phrase detection improve, with the one exception of pauses and laughter where the F-score for repair start remains the same. The final adapted model has an overall performance of 0.79 for *Frps* and 0.74 for *Frm*, showing a good improvement for repair start detection and a strong improvement for reparandum phrase detection on the initial baseline F-scores of 0.74 and 0.67, respectively. The final adapted model uses a lemmatized dataset with fastText word and character embeddings of size 50 derived from the NICT-JLE corpus, incorporating all silence and laughter features, edit term features derived from an edit term classifier as well as learner proficiency features as inputs.

Out of all of the model adaptations, the model results for lemmatization show the largest

overall performance boost thanks to its high recall, leading to an approximate 0.02 increase for both *Frps* and *Frm*. These results, extrapolated further in Table 5.2, show the detection performance of each adaptation in relation to the five difficult-to-detect disfluency cases explored in previous chapters. For ease of interpretation, the impact of reparandum length is summarised as the F-score performance of all disfluency instances with a reparandum phrase of two words or longer, and the impact of learner errors is summarised as the F-score performance of all disfluency instances with an immediate preceding learner error. Following Chapter Four, non-repetitious disfluency instance detection is measured by recall, and the remaining cases are measured by F-scores.

When looking at difficult-to-detect cases, lemmatization shows improvement across all measures apart from disfluency instances with reparandum lengths of two or more. This is especially true for non-repetitious disfluencies and disfluencies occurring with learner errors. The further inclusion of pause and laughter features shows a small overall improvement for overall *Frm*, due to an increase in prediction precision. Looking again at the difficult-to-detect cases, this is particularly impactful for the *Frm* score of disfluency instances with reparandum lengths of two or more. Character and word embeddings trained on the NICT-JLE corpus lead to an increase in overall model precision across both repair start and reparandum phrase detection, showing small improvements across all difficult-to-detect measures apart from with-edit disfluency instances. Removing edit terms from the processing pipeline increases overall recall of the model and is particularly impactful for reparandum phrase detection. This is seen to be especially beneficial for nested disfluency instances and disfluency instances that co-occur with edit terms. Finally, the inclusion of learner level information leads to higher overall model precision, primarily benefiting longer disfluency instances as well as disfluency instances that co-occur with learner errors.

Table 5.3 reports the results of the ablation study on each of the adaptations described above. A baseline model with each adaptation was trained and tested to better understand the individual impact of the features. As the results show, the removal of edit terms from the pipeline lead to the highest improvement over the baseline, with an increase of 0.03 for *Frps* and 0.05 for *Frm*. Lemmatization also has a high impact with a performance increase of 0.02 for both *Frps* and *Frm*. The remaining features of pauses and laughter, character embeddings and learner level all show increased recall at the cost of reduced precision across

Table 5.3: *Results from an ablation study on each of the adaptations derived for the final model. Precision, recall and F-score results for repair start (rps) and reparandum phrase (rm) detection are reported.*

Model (baseline + feature)	Repair phrase start (*rps*) P	R	F	Reparandum phrase (*rm*) P	R	F
baseline	0.72	0.76	0.74	0.65	0.69	0.67
+ lemmatization	0.72	0.81	0.76	0.65	0.74	0.69
+ pauses & laughter	0.70	0.78	0.74	0.64	0.71	0.67
+ char embeddings	0.69	0.80	0.74	0.63	0.73	0.68
+ edit terms	0.74	0.81	0.77	0.69	0.76	0.72
+ learner level	0.70	0.79	0.74	0.64	0.72	0.68

Table 5.4: *Precision, recall and F-score results for repair start (rps) and reparandum phrase detection (rm) according to learner proficiency level.*

Proficiency level	Repair phrase start (*rps*) P	R	F	Reparandum phrase (*rm*) P	R	F
beginner	0.84	0.84	0.84	0.81	0.80	0.81
intermediate	0.80	0.82	0.81	0.76	0.77	0.77
advanced	0.77	0.74	0.75	0.73	0.68	0.70

both *Frps* and *Frm* compared to the baseline, with only small increases in *Frm* for character embeddings and learner level.

Table 5.4 summarises the precision, recall and F-scores of the final adapted model for repair start and reparandum phrase detection according to learner proficiency level. Following the same approach as in Chapter Three, the 'beginner', 'intermediate', and 'advanced' levels refer to speakers from the NICT-JLE corpus at SST levels one to three, four to six and seven to nine, respectively. As the results show, as learner proficiency increases, both *Frps* and *Frm* decrease. Precision and recall are both negatively impacted as proficiency increases however the decline is more pronounced for recall, especially that of reparandum phrase detection.

Table 5.5 shows results for the Switchboard corpus tested on the final adapted model. The baseline model reported is taken from the DEEP baseline model for the Switchboard corpus in Chapter Four. The 'tested-only' model reflects the performance of the final adapted model reported in Table 5.1, tested using the Switchboard Corpus test set. Due to their unavailability in the Switchboard corpus, feature values for the pause and laughter and learner level inputs were set to 0. For the 'trained and tested' model, a new model was trained using

Table 5.5: *Precision, recall and F-score results for repair start (rps) and reparandum phrase detection (rm) for the Switchboard corpus both tested on the final adapted model and trained and tested on using the final model settings. The baseline model is the DEEP model from Chapter Four trained and tested using the Switchboard corpus.*

Model	Repair phrase start (*rps*)			Reparandum phrase (*rm*)		
	P	R	F	P	R	F
baseline	0.83	0.65	0.73	0.75	0.59	0.66
tested-only	0.57	0.74	0.65	0.50	0.65	0.57
trained and tested	0.83	0.67	0.74	0.77	0.62	0.68

Table 5.6: *F-scores for reparandum phrase (Frm) detection for incremental and non-incremental disfluency detection across the NICT-JLE, BULATS and Switchboard corpora.*

Model	Incremental?	Corpus	*Frm*
Adapted model	✓	NICT-JLE *	0.74
Lu et al. 2019	-	NICT-JLE *	0.80
Moore et al. 2015	✓	BULATS	0.48
Hough & Purver 2014 (STIR)	✓	Switchboard	0.78
Hough & Schlangen 2017 (DEEP)	✓	Switchboard	0.60
Lou & Johnson 2020	-	Switchboard	0.91

*The proficiency test files used for the test set in this research are a subset of those used by Lu et al. [183] (128 out of 167).

the Switchboard corpus, with pause and laughter as well as learner level features excluded. As the results show, although the 'tested-only' model does not lead to improvements over the baseline, it does show promising performance for a full zero-shot approach. The model both trained and tested with the Switchboard corpus shows an improvement of 0.01 for *Frps* and 0.02 for *Frm* compared to the baseline.

Table 5.6 summarises the reparandum phrase detection results of the final adapted model alongside other comparable models. As no previous work has been done specifically on incremental disfluency detection for the NICT-JLE corpus, multiple models approaching disfluency detection have been reported. The Lu et al. model [183] is the closest for comparison due to the fact that it uses the NICT-JLE corpus for testing. However, there are various differences between the model developed by Lu et al. and the models for this research. Firstly, the test set used for this research is a subset of those used by Lu et al (128 out of 167). In addition, detection is handled non-incrementally with flattened disfluency structures (i.e. nesting re-

moved) and the model itself is trained using the Switchboard corpus. As can be seen in Table 5.6, Lu et al.'s model outperforms the model developed for this chapter in terms of *Frm* score. The only other available work in this area that focuses on learner speech is Moore et al.'s joint parsing and disfluency detection model [181]. Operated incrementally, the model is trained on the Switchboard corpus and tested using the BULATS corpus, a collection of transcribed and disfluency-labelled recordings from Cambridge English Assessment's Business Language Testing Service. This corpus was provided to the researchers and is not publicly available. As the results show, the adapted model from this work significantly outperforms Moore et al.'s model.

At the time of writing, the STIR model from Hough and Purver has the current state-of-the-art utterance-final results for incremental disfluency detection on native speech [172]. As the results show, the adapted model from this work is lower than but not significantly behind the performance of the STIR model. Also included here is the DEEP model developed by Hough and Schlangen [175], as this is the model that the structure of the adapted model is based on. As can be seen, the model adapted for this research has a higher *Frm* score than the original version of the DEEP model built for native speech. Finally, the current state-of-the-art model for utterance-final non-incremental detection developed by Lou and Johnson [216] is reported for wider context.

5.2.3 Discussion

To summarise the above results, the adapted model developed here leads to an overall F-score improvement of 0.05 for repair start detection and 0.07 for reparandum phrase detection. Results from the ablation study showed that the removal of edit terms from the pipeline has the biggest impact on overall performance, followed by lemmatization. Detection improvement is seen across all difficult-to-detect cases, and the model adaptations are particularly advantageous for non-repetitious disfluency instances. Despite improvement from lemmatization as well as the inclusion of learner proficiency information, the detection scores for disfluency instances that co-occur with learner errors remain relatively low compared to the other difficult-to-detect cases. Results also show that learner proficiency impacts model performance. The adapted model showed moderate performance when tested on out of domain

data, providing a starting point for the further exploration of its zero-shot capabilities. Additionally, training a model for the Switchboard corpus using edit term removal, lemmatization, and character embeddings showed improvement over the baseline. Finally, the adapted model outperforms prior incremental approaches for disfluency detection in learner speech, setting a new benchmark score for further improvement.

These results not only show the overall improvement of the adapted disfluency detection model compared to the baseline model but also highlight the impact of the adapted model on difficult-to-detect disfluency cases. Lemmatization proved to be a simple and effective way to leverage the syntactic parallelism between reparandum and disfluency phrases by reducing the linguistic complexity of disfluency structures. The example below illustrates how this was particularly advantageous for non-repetitious and with-error disfluency instances.

She [*go to* + *goes to*] *the department store.*

↓

She [*go to* + *go to*] *the department store.*

As shown by the example, lemmatization not only reduces the non-repetitious disfluency instance to a repetitious (and therefore easier to detect) disfluency instance, but it also minimises the impact of noun case and verb tense learner errors. However, this may not always be the case for learner errors that result in words that do not exist such as "camed". Such words are more likely to be processed as 'unknown' and therefore go unlemmatized. It would be of interest in future work to evaluate the performance of lemmatizers on non-word learner errors such as these as well as explore how lemmatizers could be adapted to better suit learner speech. More broadly, the simple approach of lemmatization provides motivation for further methods of reducing the lexical complexity of datasets from other domains for disfluency detection.

The impact of pauses and laughter as well as character embeddings on disfluency instances with reparandum lengths of two or more shows the relationship between model precision and reparandum phrase detection performance. That is, in the final adapted model, the inclusion of these features allows the model to better identify the beginning of the disfluency phrase. The impact of pauses and laughter in particular highlights the value of non-lexical features

in settings where data is 'non-typical'. In this case, learner speech with a high rate of errors where relying solely on textual information can be detrimental to performance. To date, this is the first work to include laughter as a feature for incremental disfluency detection in a language learning setting and provides motivation to explore the combination of other non-lexical indicators of disfluency such as gestures and eye-gaze.

The smallest embedding size of 50 trained on NICT-JLE outperformed the models using large pre-trained embeddings, confirming that in this setting, prioritising the modelling of unknown words is a better approach. This may not always be the case, however. As discussed in Chapter Three, the range of topics covered in the NICT-JLE corpus is limited. In addition, the speaker group is homogeneous (Japanese learners of English), which is not always true for learner corpora which often contain speakers from a wide variety of first language backgrounds. It may be that in more general applications of disfluency detection for learner speech, large pre-trained embeddings are more suitable and the appropriate approach would need to be addressed on a case-by-case basis.

The results of edit term classification and subsequent removal from the processing pipeline confirmed the presumptions outlined in Chapters Three and Four that the negative influence of edit terms on detection performance was due to their impact on reparandum phrase length. And in fact, thanks to their high frequency in the NICT-JLE corpus, they were useful as an indicator of disfluency. This outcome further supports the inclusion of prosodic and gestural features that occur systematically prior to repair onset.

As predicted, the inclusion of speaker proficiency levels improved detection for with-error disfluency instances. Despite the limited or even negative impact on other difficult-to-detect cases, the overall improvement to the model seen when including speaker proficiency level shows the importance of prioritising approaches that tackle the issue of learner errors. It is worth noting, however, that using proficiency scores is only viable in certain use cases where access to such information is possible. For example, in a scenario where disfluency detection is used for automatic proficiency scoring, this would not be feasible.

The findings from the ablation study confirm the individual impact of both lemmatization and the removal of edit terms from the pipeline on model performance. The results also reveal the value of combining the adaptations in the final model; individually, the adaptations to include pauses and laughter, character embeddings and learner level features make little to

no impact when tested individually in the ablation study, however, when looking at the adaptations combined in the final model, performance is 0.02 higher across both *Frps* and *Frm* compared to the best-performing model (baseline + edit terms) in the ablation study.

When evaluating model performance according to speaker proficiency, the results are initially counter-intuitive—the detection performance for advanced learners is considerably lower than for beginner and intermediate learners. These results can be better understood by looking back to Figure 3.2 in Chapter Three. Firstly, the figure shows that the frequency of disfluency instances decreases as proficiency increases. With the knowledge that the final adapted model prioritises recall, it is understandable that the model will perform better on speakers with higher numbers of disfluencies as there will be proportionally fewer false positives. In addition, lower proficiency speakers also have a higher rate of repetitious disfluency instances. With the results from Chapter Four confirming that repetitious disfluencies are easier to detect, it is again understandable that this impact is reflected in the results for beginner and intermediate learners. From this, it is reasonable to presume that it may be beneficial for future work to separate data and models according to learner proficiency. In addition, considering the close similarities of disfluency behaviour seen between advanced learners and native speakers, it may even be feasible for these speakers to be modelled together.

The results of testing the adapted model on the Switchboard corpus provide a starting point for expanding the model's application to other speech domains. Although the model adaptations described above had a higher positive impact on model performance for learner speech, results still showed such adaptations to be beneficial for native disfluencies. This finding is further corroborated by the fair performance of the 'tested-only' (zero-shot) model, especially given that laughter, pause and proficiency features were set to zero at test time. Further investigating the zero-shot capabilities of the model across domains as well as other L2 datasets (such as the KISTEC corpus introduced in Chapter Three [187]) is of interest for future work.

5.3 Limitations

Without audio files available in this corpus, the instances of speakers' pauses and laughter were derived from the transcripts rather than as a result of automatic detection from audio.

Similarly to the way that ASR output has been shown to reduce model performance [183], it is possible that when detected automatically, input features of laughter and silence may only be marginally impactful for model improvement. However, with automatic detection comes access to audio files, in turn providing the opportunity for a much richer representation of prosodic features than the binary input features tested here.

Additional limitations can be found in the experimental methodology for this investigation. As discussed in previous chapters, it is clear that the difficult-to-detect disfluency cases are interdependent—reparandum phrase length impacts the likelihood of nesting and interregnum frequency, for example. Separating the impact of each iteration as they relate to individual disfluency features is insightful, but likely does not paint the whole picture of model performance. In the same way, testing the model adaptations in a different order may have yielded different outcomes. For example, if lemmatization was incorporated into the model later on in the iteration process, it may not have had such an impact on overall model performance. However, as the main aim of the research reported in this chapter was to improve the model performance, the discussion of the results above can be seen as a springboard for further investigation and not simply as definitive observations on the isolated impact of these adaptations.

With the integration of all of the adaptations, the final model is relatively complex and corpus-specific, and features such as pauses and laughter information as well as learner proficiency level may not be readily available in other corpora. With recent research papers published applying an incrementalised version of transformer-based language models such as BERT for disfluency detection [179], it may be of value to explore such model architecture for disfluency detection in learner speech. This is especially true given that BERT uses byte-pair encoding of subword units, a tokenization approach that is similar to lemmatization (see [217] for an overview).

This chapter explored adaptations to the baseline model developed in Chapter Four. The most successful approach was found through the application of a lemmatized dataset with fastText word and character embeddings of size 50 derived from the NICT-JLE corpus, silence and laughter features, edit term features derived from an edit term classifier as well as learner proficiency features as inputs. These adaptations led to a significant increase in performance for both repair start and reparandum phrase detection. The findings reveal lemmatization and the removal of edit terms to be effective approaches to model improvement and also show the advantages of using non-lexical features in settings where transcript data is 'non-typical', in this case, non-native data with a high rate of errors. Results also corroborate the relationship between disfluency frequency, repetitious disfluency instance rate and disfluency detection performance previously established in Chapter Four. Testing of the final model on the Switchboard corpus showed promising results for use of the model in zero-shot settings. In particular, the adaptations of lemmatization, character embeddings and removal of edit terms were shown to be effective in the native speech domain. Although positive improvements have been made to the baseline model across various difficult-to-detect disfluency types, nested disfluency instances and disfluency instances that co-occur with learner errors remain the biggest challenges for disfluency detection.

Chapter 6

Conclusion

The work outlined in this thesis provides an introduction to the investigation of incremental disfluency detection of learner speech. Laying the foundations for further investigation, various aspects of this topic have been explored. Firstly, a framework for a spoken dialogue system with such functionality was proposed in Chapter Two. Informed by an analysis of the prior strategies of corrective feedback in dialogue-based CALL, the framework not only provided an empirical grounding for the work of this thesis but also contextualised how incremental disfluency detection could be applied to language learning settings. Through the comparative analysis of native and learner speech disfluencies in Chapter Three, three additional challenges for disfluency detection in learner speech were hypothesised: nested disfluencies, disfluencies with edit terms and learner errors. The impact of such features on detection were confirmed in Chapter Four through the analysis of two state-of-the-art incremental disfluency detection models trained on both native and learner speech. There were two additional outcomes of this testing: *(i)*, the establishment of a train, heldout and test set of the NICT-JLE corpus to be used for future model training and evaluation and *(ii)*, the establishment of a baseline model to be used for further adaptation. Results from such adaptation detailed in Chapter Five showed a good improvement of model performance over the baseline. In particular, error analysis revealed that using a lemmatized dataset is a simple but effective approach to improve model performance, especially so for non-repetitious disfluencies. Promising results were also seen for other areas of adaptation, providing motivation for the further exploration of paralinguistic signals of disfluency such as laughter, silence as well as other multimodal features. Though significant improvements were seen from model

adaptation, several challenges for incremental disfluency detection in learner speech persist, namely the impact of learner errors on model performance.

6.1 General Limitations and Future Work

The research topic of this thesis was approached from an applied machine learning perspective, evaluating how previously established state-of-the-art approaches to incremental disfluency detection could be adapted for use with learner speech. The investigations were linguistically grounded with experimentation and model evaluation was rooted in the differences found between native and learner disfluencies. However, as described in Chapter Three, there are various other approaches to model performance evaluation that were not explored as part of this investigation. Model latency and accuracy are key components for evaluating incremental systems [123, 180] and are equally as valuable in a dialogue-based CALL setting, allowing for the generation of timely, accurate and therefore likely more effective feedback. This approach to evaluation is generally reserved for systems that are already part of a broader dialogue system architecture, with access to information such as timestamps from ASR output [176, 179, 167]. The approach taken for this work can instead be seen as a precursor to such approaches, with the long term goal of evaluating various aspects of incremental disfluency detection as part of a larger dialogue-based CALL system that implements corrective feedback.

As discussed in Chapters Two and Three, the structure and features of disfluencies may vary greatly. The approaches developed in Chapters Four and Five are designed to detect all of these variations within one model. However, as the performance results of the five difficult-to-detect cases showed, certain types of disfluencies are much more difficult to detect than others, most notably non-repetitious repairs and repairs that co-occur with learner errors. Seeing such model performance variation across disfluency types motivates the question of whether certain disfluency types should be modelled separately. For example, Ostendorf et al. split disfluency modelling between repetitious (the most easily detectable type of disfluency structure) and non-repetitious disfluency types, leading to an overall improvement of disfluency detection compared to approaches that modelled all variations together [142]. Moreover, the framework proposed in Chapter Two only requires disfluencies related to

learner errors to be attended to by the system. With the above factors in mind, it would be of interest to explore the impact of modelling certain disfluency types independently. Firstly, non-repetitious and repetitious disfluencies but also with-error disfluency types and without-error disfluency types. For the latter, drawing on the taxonomies of both native [85] and learner repair [185, 218], two sub-types of repair are of interest: 'error' repairs which relate to lexical, syntactic and phonetic issues during speech, and 'appropriateness' repairs which relate to the manner of expression. Having such a distinction within a model may not only lead to higher overall disfluency detection rates as suggested by previous studies [142] but may also reduce the negative impact of learner errors on detection shown in Chapters Four and Five.

An additional benefit of the modelling approach described above is its usefulness for feedback generation. Labelled outputs for disfluency type could be used by feedback generation models to determine whether to provide feedback on form (for error-type repairs) or meaning (for appropriateness-type repairs). The success of cross-functionality such as this within a dialogue-system framework has been reported in recent work on multi-task learning of disfluencies, part-of-speech (POS) tags, utterance segmentation and language modelling, where in a live setting such universal models outperform the tasks modelled individually [177]. The joint learning of POS tagging and language modelling are of particular interest in a dialogue-based CALL system. For the former, the shared representation of features may help alleviate the negative impact attributed to POS taggers trained on native data discussed in Chapter Four [194]. The inclusion of a language model in such a framework would not only aid disfluency detection as shown in prior research [176, 177] but would also be useful as an additional input feature for feedback generation, especially in highly predictable conversational tasks such as role-play scenarios and picture descriptions. With the most recent work on incremental disfluency detection leveraging BERT models using language generated from GPT-2 [179], there is an obvious opportunity here to integrate such a model with the multi-task approaches outlined above. Within the context of a dialogue-based CALL system designed for corrective feedback, such a system would use a fine-tuned BERT model for the joint labelling of POS tags, learner errors and disfluencies combined with a GPT-3 language generation model, the outputs of which would be used as inputs for both the BERT model itself (as done by Rohanian and Hough [179] using GPT-2) as well as the feedback generation

model. However, given that BERT is trained using written text scraped from the internet, it is unclear how such models may perform on learner data. Therefore the first step prior to developing the full incrementalised model described above would be to test the baseline performance of a BERT model fine-tuned with the adapted NICT-JLE corpus.

As discussed in Chapters Three, Four and Five, the limited features of the NICT-JLE corpus restricted the investigation of this thesis in some areas, most notably, the limited labelling of learner errors. As a result, there remain roadblocks not only to the joint modelling of disfluencies and errors but also to understanding the full impact of learner errors on disfluency detection rates. In order to address this gap and achieve the road map of future work outlined above, there is a clear motivation to develop a 'gold-standard' corpus of learner dialogues to be used not only for disfluency detection but for dialogue-based CALL applications more generally. Based on the discussions of limitations throughout this thesis, as well as the requirements of the model proposed above, there is a multitude of desired features for such a dataset. Firstly, in order to account for the influence of factors such as modality [101, 102], speaker's native language [103], learner proficiency [185] and task type [97, 87, 98], the ideal dialogues created for this corpus would involve a range of activity types (such as the picture description task and role-play scenarios in the NICT-JLE corpus) carried out by learners of English from a broad range of countries using voice assistants, on-screen virtual environments and embodied robots. The provision of time-aligned transcriptions (both orthographic and phonetic), and audio and video files of the learners in dialogue would provide the opportunity for both lexical and paralinguistic features to be modelled. Transcriptions would need to be labelled with disfluencies following the notation scheme of Shriberg [89], with additional labelling for with-error and other disfluency sub-types. Additionally, POS tags, learners' pronunciation and grammatical errors as well as their corrections would also need to be labelled. To carry out such a task from scratch would of course require a significant amount of time and resources and so may not be feasible in the immediate future, however, there are alternative routes to achieving this goal. For example, there has been some recent success in applying data augmentation techniques to generate artificial data for disfluency detection [219, 220] and these techniques could be followed to create an equivalent set of data that reflects learner speech. Furthermore, the NICT-JLE corpus itself could be adapted to better accommodate the requirements for disfluency detection. Through the addition of

gold-standard POS tags, the extension of disfluency labelling to include repair phrases as well as disfluency categorisations for with-error and repetitious types, and the additional labelling of errors that occur as part of the reparandum phrase.

Finally, looking at the framework that was proposed in Chapter Two, there are multiple ways it can be further improved. Firstly, the framework is one-directional, where the learner's speech is the only source of error, with no opportunities for the learner to initiate repair due to problems in their comprehension. Furthermore, the system focuses on same-turn repair, whereas in conversation self-initiated self-repair can happen across turns [82]. These limitations are in line with the concept of 'pedagogical repair' [49], however, the restrictions reflect the fact that the framework only captures a small part of the conversational behaviour seen in teacher-learner dialogue and could be expanded further. From the reverse perspective, using teacher-learner dialogue practices as a basis for developing dialogue-based CALL systems does not account for the additional pedagogical affordances granted by automated systems. For example, depending on the modality of the system used in practice, the framework proposed in this thesis could be enhanced by features such as live transcriptions that incrementally highlight errors [61] or the addition of delayed meta-linguistic feedback that is generated from the error and disfluency data detected during dialogue. Using the highlighted areas of improvement for the goal of developing a successful dialogue-based CALL system, it is important to not only maximise the benefits of computer-assisted features but also facilitate the qualities of human dialogue that are beneficial for effective conversational practice.

6.2 Final Remarks

This thesis has investigated incremental disfluency detection for spoken learner English, fulfilling its aim to explore how speech technology research can be applied to enhance the current capabilities of dialogue-based CALL systems. The results of this investigation have been formalised into three research papers, one of which has already been published [8], another which is currently in review and a third which is in preparation for submission. A new benchmark of performance for incremental disfluency detection of learner speech has been set, along with an adapted version of the NICT-JLE corpus that can be used by the wider research commu-

nity. Findings from this thesis have not only generated multiple avenues of research for the specific task of incremental disfluency detection in learner speech but have also contributed to the wider goal of moving toward incrementality in spoken dialogue systems.

References

[1] F. Rosell-Aguilar, "Autonomous language learning through a mobile application: a user evaluation of the busuu app," *Computer Assisted Language Learning*, vol. 31, no. 8, pp. 854–881, 2018.

[2] M. Levy and G. Stockwell, *CALL dimensions: options and issues in computer assisted language learning*. Mahwah, NJ: Lawrence Erlbaum, 2006.

[3] S. Loewen, D. Crowther, D. R. Isbell, K. M. Kim, J. Maloney, Z. F. Miller, and H. Rawal, "Mobile-assisted language learning: A Duolingo case study," *ReCALL*, vol. 31, no. 3, p. 293–311, 2019.

[4] G. Lord, "'I don't know how to use words in spanish': Rosetta Stone and learner proficiency outcomes," *The Modern Language Journal*, vol. 99, no. 2, pp. 401–405, 2015.

[5] Voicebot.ai, "UK Smart Speaker Consumer Adoption Report," 2021.

[6] S. Bibauw, T. François, and P. Desmet, "Discussing with a computer to practice a foreign language: research synthesis and conceptual framework of dialogue-based CALL," *Computer Assisted Language Learning*, pp. 1–51, 2019.

[7] 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, 2004.

[8] L. Skidmore and R. Moore, "Incremental disfluency detection for spoken learner English," in *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, (Seattle, Washington), pp. 272–278, Association for Computational Linguistics, July 2022.

[9] L. Devillers, T. Kawahara, R. K. Moore, and M. Scheutz, "Spoken Language Interaction with Virtual Agents and Robots (SLIVAR): Towards Effective and Ethical Interaction (Dagstuhl Seminar 20021)," *Dagstuhl Reports*, vol. 10, no. 1, pp. 1–51, 2020.

[10] M. Levy, *Computer Assisted Language Learning: Context and Conceptualisation*. Oxford: Clarendon Press, 1997.

[11] S.-C. Tsai, "Using Google Translate in EFL drafts: a preliminary investigation," *Computer Assisted Language Learning*, vol. 32, no. 5-6, pp. 510–526, 2019.

[12] M. Sharifi, A. R. AbuSaeedi, M. Jafarigohar, and B. Zandi, "Retrospect and prospect of computer assisted English language learning: a meta-analysis of the empirical literature," *Computer Assisted Language Learning*, vol. 31, no. 4, pp. 413–436, 2018.

[13] N. Azzouz Boudadi and M. Gutiérrez-Colón, "Effect of gamification on students' motivation and learning achievement in second language acquisition within higher education: a literature review 2011-2019," *The EUROCALL Review*, vol. 28, no. 1, pp. 40–69, 2020.

[14] L. Dong, S. Jamal Mohammed, K. Ahmed Abdel-Al Ibrahim, and A. Rezai, "Fostering EFL learners' motivation, anxiety, and self-efficacy through computer-assisted language learning- and mobile-assisted language learning-based instructions," *Frontiers in Psychology*, vol. 13, pp. 899557–899557, 2022.

[15] K. Hava, "Exploring the role of digital storytelling in student motivation and satisfaction in EFL education," *Computer Assisted Language Learning*, vol. 34, no. 7, pp. 958–978, 2021.

[16] Y. Zhonggen, Z. Ying, Y. Zhichun, and C. Wentao, "Student satisfaction, learning outcomes, and cognitive loads with a mobile learning platform," *Computer Assisted Language Learning*, vol. 32, no. 4, pp. 323–341, 2019.

[17] G. Stockwell, *Computer Assisted Language Learning: Diversity in Research and Practice*. Cambridge: Cambridge University Press, 2012.

[18] R. S. Hart, "The Illinois PLATO Foreign Languages Project," *CALICO Journal*, vol. 12, no. 4, pp. 15–37, 1995.

[19] K. Yamazaki, "Computer-assisted learning of communication (CALC): A case study of Japanese learning in a 3D virtual world," *ReCALL*, vol. 30, no. 2, pp. 214–231, 2018.

[20] M. Warschauer, "Technological change and the future of CALL," in *New perspectives on CALL for second and foreign language classrooms* (S. Fotos and C. Brown, eds.), pp. 15–25, Mahwah, NJ: Lawrence Erlbaum Associates, 2004.

[21] S. Bax, "CALL — past, present and future," *System*, vol. 31, pp. 13–28, 2003.

[22] A. Gimeno-Sanz, "Moving a step further from 'Integrative CALL'. What's to come?," *Computer Assisted Language Learning*, vol. 29, no. 6, pp. 1102–1115, 2016.

[23] M. Warschauer and D. Healey, "Computers and Language Learning: An Overview," *Language Teaching*, vol. 31, pp. 57–71, 1998.

[24] P. H. Matthews, *A short history of structural linguistics*. Cambridge ; New York: Cambridge University Press, 2001.

[25] S. Fotos and C. Brown, *New Perspectives on CALL for Second and Foreign Language Classrooms*. Mahwah, NJ: Lawrence Erlbaum Associate, 2004.

[26] D. L. Bitzer and P. G. Braunfeld, "Computers, teaching machines, and programmed learning - computer teaching machine project: PLATO on ILLIAC," *Computers and Automation*, vol. 2, no. XI, pp. 16–18, 1962.

[27] J. Richards, *Communicative Language Teaching Today*. Cambridge: Cambridge University Press, 2006.

[28] R. G. Kern, "Restructuring classroom interaction with networked computers: Effects on quantity and characteristics of language production," *The Modern Language Journal*, vol. 79, no. 4, pp. 457–476, 1995.

[29] G. Duman, G. Orhon, and N. Gedik, "Research trends in mobile assisted language learning from 2000 to 2012," *ReCALL*, vol. 27, no. 2, p. 197–216, 2015.

[30] A. Kukulska-Hulme and L. Shield, "An overview of mobile assisted language learning: From content delivery to supported collaboration and interaction," *ReCALL*, vol. 20, no. 3, p. 271–289, 2008.

[31] G. Stockwell, "Vocabulary on the move: Investigating an intelligent mobile phone-based vocabulary tutor," *Computer Assisted Language Learning*, vol. 20, no. 4, pp. 365–383, 2007.

[32] J. Fouz-González, "Podcast-based pronunciation training: Enhancing FL learners' perception and production of fossilised segmental features," *ReCALL*, vol. 31, no. 2, p. 150–169, 2019.

[33] C. R. Wigham and T. Chanier, "Interactions between text chat and audio modalities for L2 communication and feedback in the synthetic world Second Life," *Computer Assisted Language Learning*, vol. 28, no. 3, pp. 260–283, 2015.

[34] L. Lee and A. Markey, "A study of learners' perceptions of online intercultural exchange through Web 2.0 technologies," *ReCALL*, vol. 26, no. 3, p. 281–297, 2014.

[35] E. M. Golonka, A. R. Bowles, V. M. Frank, D. L. Richardson, and S. Freynik, "Technologies for foreign language learning: a review of technology types and their effectiveness," *Computer Assisted Language Learning*, vol. 27, no. 1, pp. 70–105, 2014.

[36] M. H. Long, *The role of the linguistic environment in second language acquisition*, p. 413–468. San Diego, CA: Academic Press, 1996.

[37] C. Price, A. Bunt, and G. McCalla, "L2tutor: A mixed-initiative dialogue system for improving fluency," *Computer Assisted Language Learning*, vol. 12, no. 2, p. 83–112, 1999.

[38] M. Rypa and P. Price, "VILTS: A tale of two technologies," *CALICO Journal*, vol. 16, no. 3, pp. 385–404, 1999.

[39] A. Raux and M. Eskenazi, "Using task-oriented spoken dialogue systems for language learning: Potential, practical applications and challenges," in *Proceedings of InSTIL/ICALL – NLP and Speech Technologies in Advanced Language Learning Systems*, (Venice, Italy), 2004.

[40] J. Jia and W. Chen, "The further development of CSIEC project driven by application and evaluation in english education," *British Journal of Educational Technology*, vol. 40, no. 5, pp. 901–918, 2009.

[41] E. Ayedoun, Y. Hayashi, and K. Seta, "Adding communicative and affective strategies to an embodied conversational agent to enhance second language learners' willingness to communicate," *International Journal of Artificial Intelligence in Education*, vol. 29, no. 1, pp. 29–57, 2018.

[42] T. Sydorenko, P. Daurio, and S. L. Thorne, "Refining pragmatically-appropriate oral communication via computer-simulated conversations," *International Journal of Artificial Intelligence in Education*, vol. 31, no. 1-2, pp. 157–180, 2018.

[43] H. Noh, K. Lee, S. Lee, and G. G. Lee, "POMY: A conversational virtual environment for language learning in POSTECH," in *Proceedings of the SIGDIAL 2011 Conference*, (Portland, Oregon), pp. 344–346, Association for Computational Linguistics, June 2011.

[44] K. Lee, S.-O. Kweon, S. Lee, H. Noh, and G. G. Lee, "POSTECH immersive English study (POMY): Dialog-based language learning game," *IEICE TRANSACTIONS on Information and Systems*, vol. 97, no. 7, pp. 1830–1841, 2014.

[45] K. Lee, S.-O. Kweon, S. Lee, H. Noh, J. Lee, J. Lee, H.-R. Kim, and G. G. Lee, "Effects of language learning game on Korean elementary school student," in *ISCA SIG on Speech and Language Technology in Education (SLaTE 2011), Venice*, August 2011.

[46] N. Taguchi, Q. Li, and X. Tang, "Learning Chinese formulaic expressions in a scenario-based interactive environment," *Foreign Language Annals*, vol. 50, no. 4, pp. 641–660, 2017.

[47] S. Bibauw, T. François, and P. Desmet, "Dialogue systems for language learning: Chatbots and beyond," in *The Routledge Handbook of Second Language Acquisition and Technology*, pp. 121–135, Routledge, 2022.

[48] S. Bibauw, W. Van Den Noortgate, T. François, and P. Desmet, "Dialogue systems for language learning: a meta-analysis," *Language Learning & Technology*, vol. 26, no. 1, 2022.

[49] J. Wong and H. Z. Waring, *Conversation analysis and second language pedagogy: A guide for ESL/EFL teachers*. Routledge, 2010.

[50] P. Seedhouse, "Conversation analysis and language learning," *Language Teaching*, vol. 38, no. 4, p. 165–187, 2005.

[51] R. Lyster and K. Saito, "Oral feedback in classroom SLA: A meta-analysis," *Studies in Second Language Acquisition*, vol. 32, p. 265–302, 2010.

[52] R. Ellis, *Oral corrective feedback in L2 Classrooms: What We Know So Far*, vol. 66 of *ESL & Applied Linguistics Series*, ch. 1, pp. 3–18. Oxford, UK: Taylor & Francis, 2017.

[53] R. Lyster, K. Saito, and M. Sato, "Oral corrective feedback in second language classrooms," *Language Teaching*, vol. 46, no. 1, p. 1–40, 2013.

[54] P. Seedhouse, *The interactional architecture of the language classroom: A conversation analysis perspective*. Malden, MA: Blackwell, 2004.

[55] E. Khodadady and J. Alifathabadi, "Repairing conversation and foreign language proficiency," *Journal of Language Teaching and Research*, vol. 3, no. 4, pp. 736–743, 2012.

[56] S. Li, "Oral corrective feedback," *ELT Journal*, vol. 68, pp. 196–198, 12 2013.

[57] R. Ellis, "Corrective feedback and teacher development," *L2 Journal*, vol. 1, no. 1, 2009.

[58] J. D. Kaplan, M. A. Sabol, R. A. Wisher, and R. J. Seidel, "The military language tutor (MILT) program: An advanced authoring system," *Computer Assisted Language Learning*, vol. 11, no. 3, pp. 265–287, 1998.

[59] M. Eskenazi, "Using automatic speech processing for foreign language pronunciation tutoring: some issues and a prototype," *Language Learning & Technology*, vol. 2, no. 2, pp. 62–76, 1999.

[60] W. Menzel, D. Herron, R. Morton, D. Pezzotta, P. Bonaventura, and P. Howarth, "Interactive pronunciation training," *ReCALL*, vol. 13, no. 1, pp. 67–78, 2001.

[61] H. Ye and S. J. Young, "Improving the speech recognition performance of beginners in spoken conversational interaction for language learning.," in *Interspeech*, pp. 289–292, 2005.

[62] P. H. Su, Y. B. Wang, T. H. Yu, and L. S. Lee, "A dialogue game framework with personalized training using reinforcement learning for computer-assisted language learning," in *Proceedings of ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8213–8217, 2013.

[63] N. R. Walker, P. Trofimovich, H. Cedergren, and E. Gatbonton, "Using ASR technology in language training for specific purposes: A perspective from Quebec, Canada," *CALICO Journal*, vol. 28, no. 3, pp. 721–743, 2011.

[64] W. L. Johnson, S. Marsella, and H. Vilhjalmsson, "The DARWARS tactical language training system," in *Proceedings of Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, 2004.

[65] H. Strik, J. Colpaert, J. van Doremalen, and C. Cucchiarini, "The DISCO ASR-based call system: practicing L2 oral skills and beyond," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pp. 2702–2707, 2012.

[66] O. Engwall, "Analysis of and feedback on phonetic features in pronunciation training with a virtual teacher," *Computer Assisted Language Learning*, vol. 25, no. 1, pp. 37–64, 2012.

[67] R. Ai, M. Charfuelan, W. Kasper, T. Kluwer, H. Uszkoreit, F. Xu, S. Gasber, and P. Gienandt, "Sprinter: Language technologies for interactive and multimedia language learning," in *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pp. 2733–2738, 2014.

[68] K. A. Wachowicz and B. Scott, "Software that listens: It's not a question of whether, it's a question of how," *CALICO journal*, pp. 253–276, 1999.

[69] S. Lee, H. Noh, J. Lee, K. Lee, and G. G. Lee, "Cognitive effects of robot-assisted language learning on oral skills," in *Second Language Studies: Acquisition, Learning, Education and Technology*, 2010.

[70] Y. Nagai, T. Senzai, S. Yamamoto, and M. Nishida, "Sentence classification with grammatical errors and those out of scope of grammar assumption for dialogue-based call systems," in *Text, Speech and Dialogue* (P. Sojka, A. Horák, I. Kopeček, and K. Pala, eds.), (Berlin, Heidelberg), pp. 616–623, Springer Berlin Heidelberg, 2012.

[71] C. Cucchiarini, S. Bodnar, B. P. de Vries, R. van Hout, and H. Strik, "ASR-based CALL systems and learner speech data: new resources and opportunities for research and development in second language learning," in *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*, pp. 2708–2714, 2014.

[72] C. Wang and S. Seneff, "A spoken translation game for second language learning," *Frontiers in Artificial Intelligence and Applications*, vol. 158, p. 315, 2007.

[73] P. Bouillon, E. Rayner, N. Tsourakis, and Z. Qinglu, "A student-centered evaluation of a web-based spoken translation game," in *Proceedings of the ISCA International Workshop on Speech and Language Technology in Education (SLaTE)*, pp. 33–36, 2011.

[74] V. Zue, S. Seneff, J. Polifroni, H. Meng, and J. Glass, "Multilingual human-computer interactions: From information access to language learning," in *Proceeding of the Fourth International Conference on Spoken Language Processing (ICSLP'96)*, vol. 4, pp. 2207–2210, IEEE, 1996.

[75] O. P. Kweon, A. Ito, M. Suzuki, and S. Makino, "A Japanese dialogue-based CALL system with mispronunciation and grammar error detection," in *Eighth International Conference on Spoken Language Processing*, 2004.

[76] O.-W. Kwon, K. Lee, Y.-K. Kim, and Y. Lee, "Genietutor: a computer assisted second-language learning system based on semantic and grammar correctness evaluations," in *Critical CALL–proceedings of the 2015 EUROCALL conference, Padova, Italy*, pp. 330–335, 2015.

[77] S. Chevalier, "Speech interaction with Saybot, a CALL software to help Chinese learners of English.," in *SLaTE*, pp. 37–40, 2007.

[78] H. Morton and M. A. Jack, "Scenario-based spoken interaction with virtual agents," *Computer Assisted Language Learning*, vol. 18, no. 3, pp. 171–191, 2005.

[79] A. Raux and M. Eskenazi, "Using task-oriented spoken dialogue systems for language learning: potential, practical applications and challenges," in *InSTIL/ICALL Symposium 2004*, 2004.

[80] J. Bernstein, A. Najmi, and F. Ehsani, "Subarashii: Encounters in Japanese spoken language education," *CALICO Journal*, vol. 16, no. 3, 1999.

[81] C. M. Forsyth, C. Luce, D. Zapata-Rivera, G. T. Jackson, K. Evanini, and Y. So, "Evaluating English language learners' conversations: Man vs. machine," *Computer Assisted Language Learning*, vol. 32, no. 4, pp. 398–417, 2019.

[82] E. A. Schegloff, G. Jefferson, and H. Sacks, "The preference for self-correction in the organization of repair in conversation," *Language*, vol. 53, no. 2, pp. 361–382, 1977.

[83] P. Buckwalter, "Repair sequences in Spanish L2 dyadic discourse: A descriptive study," *The Modern Language Journal*, vol. 85, no. 3, pp. 380–397, 2001.

[84] C. James, *Errors in language learning and use: exploring error analysis*. Applied linguistics and language study, London: Longman, 1998.

[85] W. J. Levelt, "Monitoring and self-repair in speech," *Cognition*, vol. 14, no. 1, pp. 41–104, 1983.

[86] M. Purver, J. Hough, and C. Howes, "Computational models of miscommunication phenomena," *Topics in Cognitive Science*, vol. 10, pp. 425–451, 2018.

[87] M. Colman and P. Healey, "The distribution of repair in dialogue," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 33, 2011.

[88] E. Charniak and M. Johnson, "Edit detection and parsing for transcribed speech," in *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001.

[89] E. E. Shriberg, *Preliminaries to a theory of speech disfluencies*. PhD thesis, University of California, Berkeley, 1994.

[90] J. Bear, J. Dowding, and E. Shriberg, "Integrating multiple knowledge sources for detection and correction of repairs in human-computer dialog," in *30th Annual Meeting of the Association for Computational Linguistics*, pp. 56–63, 1992.

[91] W. J. Levelt and A. Cutler, "Prosodic marking in speech repair," *Journal of Semantics*, vol. 2, no. 2, pp. 205–218, 1983.

[92] M. H. Goodwin and C. Goodwin, "Gesture and coparticipation in the activity of searching for a word," *Semiotica*, vol. 62, no. 1-2, 1986.

[93] J. Radford, "Word searches: on the use of verbal and non-verbal resources during classroom talk," *Clinical linguistics & phonetics*, vol. 23, no. 8, pp. 598–610, 2009.

[94] C. Oomen and A. Postma, "Effects of divided attention on the production of filled pauses and repetitions.," *Journal of Speech, Language & Hearing Research*, vol. 44, no. 5, pp. 997–1004, 2001.

[95] P. H. Tannenbaum, F. Williams, and C. S. Hillier, "Word predictability in the environments of hesitations," *Journal of Verbal Learning and Verbal Behavior*, vol. 4, no. 2, pp. 134–140, 1965.

[96] P. Sen, "Speech disfluencies occur at higher perplexities," in *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, pp. 92–97, Association for Computational Linguistics, Dec. 2020.

[97] R. J. Lickley, "Dialogue moves and disfluency rates," in *ISCA tutorial and research workshop (ITRW) on disfluency in spontaneous speech*, 2001.

[98] H. Moniz, F. Batista, A. I. Mata, and I. Trancoso, "Speaking style effects in the production of disfluencies," *Speech communication*, vol. 65, pp. 20–35, 2014.

[99] H. Bortfeld, S. D. Leon, J. E. Bloom, M. F. Schober, and S. E. Brennan, "Disfluency rates in conversation: Effects of age, relationship, topic, role, and gender," *Language and speech*, vol. 44, no. 2, pp. 123–147, 2001.

[100] T. Tran, M. Tinkler, G. Yeung, A. Alwan, and M. Ostendorf, "Analysis of disfluency in children's speech," in *Proc. Interspeech 2020*, pp. 4278–4282, 2020.

[101] S. Oviatt, "Predicting spoken disfluencies during human-computer interaction," *Computer Speech and Language*, vol. 9, no. 1, pp. 19–36, 1995.

[102] G. Savova and J. Bachenko, "Designing for errors: similarities and differences of disfluency rates and prosodic characteristics across domains," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[103] B. A. Fox, Y. Maschler, and S. Uhmann, "A cross-linguistic study of self-repair: Evidence from English, German, and Hebrew," *Journal of Pragmatics*, vol. 42, no. 9, pp. 2487–2505, 2010. How people talk to Robots and Computers.

[104] R. Wiese, *Language production in foreign and native languages: Same or different*, pp. 11–25. Narr Verlag.

[105] L. Temple, "Disfluencies in learner speech," *Australian Review of Applied Linguistics*, vol. 15, no. 2, pp. 29–44, 1992.

[106] H. Hilton, "The link between vocabulary knowledge and spoken L2 fluency," *Language Learning Journal*, vol. 36, no. 2, pp. 153–166, 2008.

[107] N. H. De Jong, M. P. Steinel, A. Florijn, R. Schoonen, and J. H. Hulstijn, "Linguistic skills and speaking fluency in a second language," *Applied Psycholinguistics*, vol. 34, no. 5, pp. 893–916, 2013.

[108] D. Jurafsky and J. H. Martin, "Speech and language processing (3rd (draft) ed.)," 2022.

[109] D. Hindle, "Deterministic parsing of syntactic non-fluencies," in *21st Annual Meeting of the Association for Computational Linguistics*, (Cambridge, Massachusetts, USA), pp. 123–128, Association for Computational Linguistics, June 1983.

[110] M. G. Core and L. Schubert, "A syntactic framework for speech repairs and other disruptions," in *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 413–420, 1999.

[111] J. G. Kahn, M. Lease, E. Charniak, M. Johnson, and M. Ostendorf, "Effective use of prosody in parsing conversational speech," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, (Vancouver, British Columbia, Canada), pp. 233–240, Association for Computational Linguistics, Oct. 2005.

[112] M. Lease and M. Johnson, "Early deletion of fillers in processing conversational speech," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, (New York City, USA), pp. 73–76, Association for Computational Linguistics, June 2006.

[113] S. Young and M. Matessa, "Using pragmatic and semantic knowledge to correct parsing of spoken language utterances," in *Proc. 2nd European Conference on Speech Communication and Technology (Eurospeech 1991)*, pp. 223–227, 1991.

[114] S. Wu, D. Zhang, M. Zhou, and T. Zhao, "Efficient disfluency detection with transition-based parsing," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 495–503, 2015.

[115] M. Yoshikawa, H. Shindo, and Y. Matsumoto, "Joint transition-based dependency parsing and disfluency detection for automatic speech recognition texts," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1036–1041, 2016.

[116] S. Wang, W. Che, Y. Zhang, M. Zhang, and T. Liu, "Transition-based disfluency detection using LSTMs," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2785–2794, 2017.

[117] T. Tran, S. Toshniwal, M. Bansal, K. Gimpel, K. Livescu, and M. Ostendorf, "Parsing speech: a neural approach to integrating lexical and acoustic-prosodic information," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 69–81, Association for Computational Linguistics, June 2018.

[118] P. Jamshid Lou, Y. Wang, and M. Johnson, "Neural constituency parsing of speech transcripts," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 2756–2765, Association for Computational Linguistics, June 2019.

[119] P. Jamshid Lou and M. Johnson, "Improving disfluency detection by self-training a self-attentive model," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3754–3763, Association for Computational Linguistics, July 2020.

[120] 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-04)*, pp. 33–39, 2004.

[121] M. Honal and T. Schultz, "Correction of disfluencies in spontaneous speech using a noisy-channel approach," in *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, pp. 2781–2784, 2003.

[122] M. Johnson, E. Charniak, and M. Lease, "An improved model for recognizing disfluencies in conversational speech," in *Proceedings of Rich Transcription Workshop*, 2004.

[123] S. Zwarts and M. Johnson, "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, (Portland, Oregon, USA), pp. 703–711, Association for Computational Linguistics, June 2011.

[124] G. Neubig, Y. Akita, S. Mori, and T. Kawahara, "A monotonic statistical machine translation approach to speaking style transformation," *Computer Speech & Language*, vol. 26, no. 5, pp. 349–370, 2012.

[125] P. Jamshid Lou and M. Johnson, "Disfluency detection using a noisy channel model and a deep neural language model," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Vancouver, Canada), pp. 547–553, Association for Computational Linguistics, July 2017.

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

[127] P. Jamshid Lou, P. Anderson, and M. Johnson, "Disfluency detection using auto-correlational neural networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 4610–4619, Association for Computational Linguistics, Oct.-Nov. 2018.

[128] K. Georgila, "Using integer linear programming for detecting speech disfluencies," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pp. 109–112, 2009.

[129] P. Heeman and J. Allen, "Detecting and correcting speech repairs," in *32nd Annual Meeting of the Association for Computational Linguistics*, (Las Cruces, New Mexico, USA), pp. 295–302, Association for Computational Linguistics, June 1994.

[130] P. A. Heeman, K.-h. Loken-Kim, and J. F. Allen, "Combining the detection and correction of speech repairs," in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, vol. 1, pp. 362–365, IEEE, 1996.

[131] Y. Liu, E. Shriberg, and A. Stolcke, "Automatic disfluency identification in conversational speech using multiple knowledge sources," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[132] A. Stolcke and E. Shriberg, "Statistical language modeling for speech disfluencies," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1, pp. 405–408, IEEE, 1996.

[133] E. Shriberg, R. Bates, and A. Stolcke, "A prosody only decision-tree model for disfluency detection," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[134] J. Kim, S. E. Schwarm, and M. Ostendorf, "Detecting structural metadata with decision trees and transformation-based learning," in *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, (Boston, Massachusetts, USA), pp. 137–144, Association for Computational Linguistics, May 2 - May 7 2004.

[135] P. A. Heeman and J. F. Allen, "Intonational boundaries, speech repairs, and discourse markers: Modeling spoken dialog," in *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, (Madrid, Spain), pp. 254–261, Association for Computational Linguistics, July 1997.

[136] A. Stolcke, E. Shriberg, R. A. Bates, M. Ostendorf, D. Z. Hakkani, M. Plauche, G. Tür, and Y. Lu, "Automatic detection of sentence boundaries and disfluencies based on recognized words," in *ICSLP*, vol. 2, pp. 2247–2250, 1998.

[137] D. Baron, E. Shriberg, and A. Stolcke, "Automatic punctuation and disfluency detection in multi-party meetings using prosodic and lexical cues," in *Proc. 7th International Conference on Spoken Language Processing (ICSLP 2002)*, pp. 949–952, 2002.

[138] Y. Liu, E. Shriberg, A. Stolcke, and M. Harper, "Comparing HMM, maximum entropy, and conditional random fields for disfluency detection," in *Proceedings of Interspeech 2005*, pp. 3313–3316, 2005.

[139] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *IEEE Transactions on audio, speech, and language processing*, vol. 14, no. 5, pp. 1526–1540, 2006.

[140] Q. Zhang, F. Weng, and Z. Feng, "A progressive feature selection algorithm for ultra large feature spaces," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, (USA), p. 561–568, Association for Computational Linguistics, 2006.

[141] K. Georgila, N. Wang, and J. Gratch, "Cross-domain speech disfluency detection," in *Proceedings of the SIGDIAL 2010 Conference*, pp. 237–240, 2010.

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

[143] V. Zayats, M. Ostendorf, and H. Hajishirzi, "Multi-domain disfluency and repair detection," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[144] V. Zayats, M. Ostendorf, and H. Hajishirzi, "Unediting: Detecting disfluencies without careful transcripts," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1410–1415, 2015.

[145] J. Ferguson, G. Durrett, and D. Klein, "Disfluency detection with a semi-markov model and prosodic features," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 257–262, 2015.

[146] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[147] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[148] E. Cho, J. Niehues, T.-L. Ha, and A. Waibel, "Multilingual disfluency removal using NMT," in *Proceedings of the 13th International Conference on Spoken Language Translation*, 2016.

[149] S. Wang, W. Che, and T. Liu, "A neural attention model for disfluency detection," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 278–287, 2016.

[150] Q. Dong, F. Wang, Z. Yang, W. Chen, S. Xu, and B. Xu, "Adapting translation models for transcript disfluency detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6351–6358, 2019.

[151] P. J. Lou and M. Johnson, "End-to-end speech recognition and disfluency removal," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 2051–2061, 2020.

[152] Q. Chen, M. Chen, B. Li, and W. Wang, "Controllable time-delay transformer for real-time punctuation prediction and disfluency detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8069–8073, 2020.

[153] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[154] N. Bach and F. Huang, "Noisy BiLSTM-based models for disfluency detection," in *Proceedings of Interspeech 2019*, pp. 4230–4234, 2019.

[155] B. Lin and L. Wang, "Joint prediction of punctuation and disfluency in speech transcripts," in *Proc. Interspeech 2020*, pp. 716–720, 2020.

[156] J. C. Rocholl, V. Zayats, D. D. Walker, N. B. Murad, A. Schneider, and D. J. Liebling, "Disfluency Detection with Unlabeled Data and Small BERT Models," in *Proceedings of Interspeech 2021*, pp. 766–770, 2021.

[157] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[158] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[159] C. H. Nakatani and J. Hirschberg, "A speech-first model for repair detection and correction," in *31st Annual Meeting of the Association for Computational Linguistics*, pp. 46–53, 1993.

[160] D. O'Shaughnessy, "Correcting complex false starts in spontaneous speech," in *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. I–349, IEEE, 1994.

[161] V. Zayats and M. Ostendorf, "Giving attention to the unexpected: Using prosody innovations in disfluency detection," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 86–95, Association for Computational Linguistics, June 2019.

[162] S. Wang, W. Che, Q. Liu, P. Qin, T. Liu, and W. Y. Wang, "Multi-task self-supervised learning for disfluency detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 9193–9200, 2020.

[163] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 517–520, IEEE Computer Society, 1992.

[164] M.-H. Siu and M. Ostendorf, "Modeling disfluencies in conversational speech," in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, vol. 1, pp. 386–389, IEEE, 1996.

[165] S. E. Brennan and M. F. Schober, "How listeners compensate for disfluencies in spontaneous speech," *Journal of Memory and Language*, vol. 44, no. 2, pp. 274–296, 2001.

[166] D. Schlangen and G. Skantze, "A general, abstract model of incremental dialogue processing," *Dialogue & Discourse*, vol. 2, no. 1, pp. 83–111, 2011.

[167] A. Chen, V. Zayats, D. Walker, and D. Padfield, "Teaching BERT to wait: Balancing accuracy and latency for streaming disfluency detection," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Seattle, United States), pp. 827–838, Association for Computational Linguistics, July 2022.

[168] M. S. Rasooli and J. Tetreault, "Non-monotonic parsing of fluent umm i mean disfluent sentences," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pp. 48–53, 2014.

[169] M. Honnibal and M. Johnson, "Joint incremental disfluency detection and dependency parsing," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 131–142, 2014.

[170] P. A. Heeman and J. Allen, "Speech repairs, intonational phrases, and discourse markers: modeling speakers' utterances in spoken dialogue," *Computational Linguistics*, vol. 25, no. 4, pp. 527–572, 1999.

[171] T. Miller and W. Schuler, "A syntactic time-series model for parsing fluent and disfluent speech," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 569–576, 2008.

[172] J. Hough and M. Purver, "Strongly incremental repair detection," in *Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference (EMNLP)*, (Doha, Qatar), pp. 78–89, ACL, 2014.

[173] S. Zwarts, M. Johnson, and R. Dale, "Detecting speech repairs incrementally using a noisy channel approach," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 1371–1378, 2010.

[174] J. Hough and D. Schlangen, "Recurrent neural networks for incremental disfluency detection," in *Proceedings of Interspeech 2015*, pp. 849–853, 2015.

[175] J. Hough and D. Schlangen, "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: Volume 1, Long Papers*, pp. 326–336, 2017.

[176] I. Shalyminov, A. Eshghi, and O. Lemon, "Multi-task learning for domain-general spoken disfluency detection in dialogue systems," in *Proceedings of the 22nd Workshop*

on the Semantics and Pragmatics of Dialogue - Full Papers, (Aix-en-Provence, France), SEMDIAL, Nov. 2018.

[177] M. Rohanian and J. Hough, "Re-framing incremental deep language models for dialogue processing with multi-task learning," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 497–507, International Committee on Computational Linguistics, Dec. 2020.

[178] B. Madureira and D. Schlangen, "Incremental processing in the age of non-incremental encoders: An empirical assessment of bidirectional models for incremental NLU," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 357–374, Association for Computational Linguistics, Nov. 2020.

[179] M. Rohanian and J. Hough, "Best of both worlds: Making high accuracy non-incremental transformer-based disfluency detection incremental," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3693–3703, 2021.

[180] T. Baumann, O. Buß, and D. Schlangen, "Evaluation and optimisation of incremental processors," *Dialogue & Discourse*, vol. 2, no. 1, pp. 113–141, 2011.

[181] R. Moore, A. Caines, C. Graham, and P. Buttery, "Incremental dependency parsing and disfluency detection in spoken learner English," in *International Conference on Text, Speech, and Dialogue*, pp. 470–479, Springer, 2015.

[182] L. Chambers and K. Ingham, "The BULATS online speaking test," 2011.

[183] Y. Lu, M. J. F. Gales, K. M. Knill, P. Manakul, and Y. Wang, "Disfluency detection for spoken learner English," in *Proceedings of the 8th ISCA Workshop on Speech and Language Technology in Education (SLaTE 2019)*, pp. 74–78, 2019.

[184] Y. Lu, M. J. Gales, and Y. Wang, "Spoken language 'grammatical error correction'," in *Proceedings of Interspeech 2020*, pp. 3840–3844, 2020.

[185] G. W. C. M. Van Hest, *Self-repair in L1 and L2 production*. Tilburg: Tilburg University Press, 1996.

[186] J. Kormos, "A new psycholinguistic taxonomy of self-repairs in L2: A qualitative analysis with retrospection," *Even Yearbook, ELTE SEAS Working Papers in Linguistics*, vol. 3, pp. 43–68, 1998.

[187] K. Kanzawa, Y. Kobayashi, J. Lee, H. Mitsunaga, M. Mori, and Y. Tanaka, "The KIT Speaking Test Corpus (KISTEC)." https://kitstcorpus.jp/, 2022.

[188] M. Meteer, A. Taylor, R. MacIntyre, and R. Iyer, "Disfluency annotation stylebook for the Switchboard Corpus," *Technical Report, Department of Computer and Information Science, University of Pennsylvania*, 1995.

[189] M. C. TEMPLIN, *Certain Language Skills in Children: Their Development and Interrelationships*, vol. 26. University of Minnesota Press, ned - new edition ed., 1957.

[190] R. S. Wachal and O. Spreen, "A computer-aided investigation of linguistic performance: Normal and pathological language," 1970.

[191] R. Koizumi and Y. In'nami, "Vocabulary knowledge and speaking proficiency among second language learners from novice to intermediate levels," *Journal of Language Teaching & Research*, vol. 4, no. 5, 2013.

[192] T. Uchihara and J. Clenton, "Investigating the role of vocabulary size in second language speaking ability," *Language Teaching Research*, vol. 24, no. 4, pp. 540–556, 2020.

[193] P. Skehan, "Modelling second language performance: Integrating complexity, accuracy, fluency, and lexis," *Applied linguistics*, vol. 30, no. 4, pp. 510–532, 2009.

[194] R. Nagata, T. Mizumoto, Y. Kikuchi, Y. Kawasaki, and K. Funakoshi, "A POS tagging model adapted to learner English," in *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pp. 39–48, 2018.

[195] C. Napoles, A. Cahill, and N. Madnani, "The effect of multiple grammatical errors on processing non-native writing," in *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 1–11, 2016.

[196] M. Zuniga and D. Simard, "Factors influencing L2 self-repair behavior: The role of L2 proficiency, attentional control and L1 self-repair behavior," *Journal of psycholinguistic research*, vol. 48, no. 1, pp. 43–59, 2019.

[197] T. West, "A conversation analysis of repair strategies in multilingual English discussion classes," *New Directions in Teaching and Learning English Discussion*, vol. 6, pp. 219–228, 2018.

[198] S. Zeng, "Second language learners' strong preference for self-initiated self-repair: Implications for theory and pedagogy," *Journal of Language Teaching and Research*, vol. 10, no. 3, pp. 541–548, 2019.

[199] A. Clark, G. Giorgolo, and S. Lappin, "Statistical representation of grammaticality judgements: the limits of n-gram models," in *Proceedings of the fourth annual workshop on cognitive modeling and computational linguistics (CMCL)*, pp. 28–36, 2013.

[200] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, (Valletta, Malta), pp. 45–50, ELRA, May 2010.

[201] D. Guo, G. Tur, W.-t. Yih, and G. Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, pp. 554–559, IEEE, 2014.

[202] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 252–259, 2003.

[203] L. Chen, M. Harper, and F. Quek, "Gesture patterns during speech repairs," in *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, ICMI '02, (USA), p. 155, IEEE Computer Society, 2002.

[204] P. Tavakoli, "Pausing patterns: Differences between L2 learners and native speakers," *ELT journal*, vol. 65, no. 1, pp. 71–79, 2011.

[205] C. Mazzocconi, Y. Tian, and J. Ginzburg, "What's your laughter doing there? A taxonomy of the pragmatic functions of laughter," *IEEE Transactions on Affective Computing*, 2020.

[206] E. Holt, "The last laugh: Shared laughter and topic termination," *Journal of Pragmatics*, vol. 42, no. 6, pp. 1513–1525, 2010.

[207] P. Glenn, "Interviewees volunteered laughter in employment interviews: A case of 'nervous' laughter," *Studies of laughter in interaction*, pp. 255–275, 2013.

[208] M. Haakana, "Laughter as a patient's resource: Dealing with delicate aspects of medical interaction," *Text & Talk*, vol. 21, no. 1-2, pp. 187–219, 2001.

[209] C. Petitjean and E. González-Martínez, "Laughing and smiling to manage trouble in French-language classroom interaction," *Classroom Discourse*, vol. 6, no. 2, pp. 89–106, 2015.

[210] S. D. Looney and Y. He, "Laughter and smiling: sequential resources for managing delayed and disaligning responses," *Classroom Discourse*, vol. 12, no. 4, pp. 319–343, 2021.

[211] Y. Gao and Y. Wu, "Laughter as responses to different actions in L2 oral proficiency interview," *Open Journal of Modern Linguistics*, vol. 8, no. 6, pp. 199–220, 2018.

[212] Y. Gao, "Laughter as same-turn self-repair initiation in L2 oral proficiency interview," *Open Journal of Social Sciences*, vol. 8, no. 4, pp. 479–494, 2020.

[213] V. Maraev, B. Noble, C. Mazzocconi, and C. Howes, "Dialogue act classification is a laughing matter," in *Proceedings of the 25th Workshop on the Semantics and Pragmatics of Dialogue*, pp. 120–131, 2021.

[214] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

[215] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[216] P. Jamshid Lou and M. Johnson, "Improving disfluency detection by self-training a self-attentive model," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 3754–3763, Association for Computational Linguistics, July 2020.

[217] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 1715–1725, Association for Computational Linguistics, Aug. 2016.

[218] J. Kormos, "Monitoring and self-repair in L2," *Language Learning*, vol. 49, no. 2, pp. 303–342, 1999.

[219] J. Yang, D. Yang, and Z. Ma, "Planning and generating natural and diverse disfluent texts as augmentation for disfluency detection," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 1450–1460, Association for Computational Linguistics, Nov. 2020.

[220] T. Passali, T. Mavropoulos, G. Tsoumakas, G. Meditskos, and S. Vrochidis, "LARD: Large-scale artificial disfluency generation," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, (Marseille, France), pp. 2327–2336, European Language Resources Association, June 2022.

[221] A. Caines, D. Nicholls, and P. Buttery, "Annotating errors and disfluencies in transcriptions of speech," Tech. Rep. UCAM-CL-TR-915, University of Cambridge, Computer Laboratory, Dec. 2017.

[222] S. Ishikawa, *Design of the ICNALE Spoken: A new database for multi-modal contrastive interlanguage analysis*, pp. 63–76. Kobe, Japan: Kobe University, 2014.

[223] S. Ishikawa, "The ICNALE Spoken Dialogue: A new dataset for the study of Asian learners' performance in L2 English interviews," *English Teaching (The Korea Association of Teachers of English)*, vol. 74, pp. 153–177, 2019.

[224] S. Reder, K. Harris, and K. Setzler, "The Multimedia Adult ESL Learner Corpus," *TESOL Quarterly*, vol. 37, no. 3, pp. 546–557, 2003.

[225] S. Yamamoto, K. Taguchi, K. Ijuin, I. Umata, and M. Nishida, "Multimodal Corpus of Multiparty Conversations in L1 and L2 Languages and Findings Obtained from It," *Lang. Resour. Eval.*, vol. 49, p. 857–882, Dec. 2015.

[226] D. Gablasova, V. Brezina, and T. McEnery, "The Trinity Lancaster Corpus: Development, description and application," *International Journal of Learner Corpus Research*, vol. 5, pp. 126–158, 2019.

[227] S. Diemer, B. M-L., S. Schmidt, and C. Collect, "ViMELF: Corpus of Video-Mediated English as a Lingua Franca Conversations. Version 1.0. The CASE project." Birkenfeld: Trier University of Applied Sciences, 2018.

Appendix

Table 1: *Feedback types and methods of dialogue-based CALL systems based on a systematic analysis of corrective feedback.*

System	Year	Feedback Type	Feedback Method						
			repetition	elicitation	clarification request	recast	explicit correction	explicit correction with metalinguistic feedback	word search
Language Tutor	1996	reformulation					✓		
VILTS/ECHOS	1996	delayed							
Subarashii	1997	prompting		✓					
Fluency	1998	delayed							
TraciTalk	1999	prompting, reformulation			✓	✓			
ISLE	2000	delayed							
ARTUR	2004	delayed							
[Kweon]#	2004	reformulation					✓		
LEt's Go	2004	reformulation				✓			
SCILL	2004	delayed							
TLCTS	2004	delayed							
SPELL System	2005	reformulation				✓			
Let's Translate	2007	prompting, bridging the gap			✓				✓
Saybot	2007	prompting, reformulation	✓		✓			✓	
DISCO	2009	prompting, reformulation, delayed			✓	✓			
Engkey	2010	prompting, reformulation			✓	✓			
Mero	2010	reformulation						✓	
POMY	2010	prompting, reformulation	✓			✓			
CALL-SLT	2011	prompting, bridging the gap			✓				✓
Virtual Language Patient	2011	delayed							
[Nagai]#	2012	prompting, reformulation			✓	✓			
[Su]#	2013	delayed							
GREET	2014	prompting, reformulation, delayed		✓	✓	✓			
Sprinter	2014	delayed							
GenieTutor	2015	reformulation, delayed			✓			✓	
CBA	2019	prompting		✓					

Table 2: *Conversational corpora explored for experimentation, where the Switchboard and NICT-JLE are the only corpora that meet the requirements for experimentation.*

Corpus	L1	L2	Disfluency-tagged	Publicly available
BULATS [182, 221]		✓	✓	
ICNALE [222, 223]	✓	✓		✓
MAELC [224]		✓		✓
MCMC [225]		✓		✓
NICT-JLE [7]		✓	✓	✓
Switchboard [163]	✓		✓	✓
Trinity Lancaster Corpus [226]	✓	✓		✓
ViMELF [227]	✓	✓		✓

Table 3: *The means and standard deviations of reparandum phrase length, repairs per utterance and edit terms per utterance for the train, heldout and test sets for the NICT-JLE corpus, showing equivalency across sets. Each set has an equal distribution of learner proficiency levels and all files in the test set contain labels for learners' errors.*

	Train	Heldout	Test
reparandum phrase length	2.04 (±0.19)	2.02 (±0.18)	2.08 (±0.23)
repairs per utterance	0.38 (±0.14)	0.37 (±0.13)	0.36 (±0.15)
edit terms per utterance	0.75 (±0.20)	0.74 (±0.21)	0.73 (±0.17)

Table 4: *Count and percentage distribution of SST proficiency levels for the train, heldout and test sets for the adapted NICT-JLE corpus, showing equivalency across sets.*

SST Level	Train	Heldout	Test
1	2 (0.2%)	0 (0.0%)	1 (0.8%)
2	27 (2.6%)	4 (3.1%)	4 (3.1%)
3	176 (17.2%)	23 (18.0%)	23 (18.0%)
4	395 (38.5%)	44 (34.3%)	43 (33.6%)
5	190 (18.5%)	23 (18.0%)	23 (18.0%)
6	102 (10.0%)	14 (10.9%)	14 (10.9%)
7	59 (5.8%)	9 (7.0%)	9 (7.0%)
8	44 (4.3%)	6 (4.7%)	6 (4.7%)
9	30 (2.9%)	5 (3.9%)	5 (3.9%)

Table 5: *Precision, recall and F-score results for repair start (rps) and reparandum phrase (rm) detection of the difficult-to-detect cases for the STIR model trained on the Switchboard corpus.*

Model iteration	Precision		Recall		F-score	
	rps	*rm*	*rps*	*rm*	*rps*	*rm*
length = 1	0.72	0.71	0.83	0.81	0.77	0.76
length = 2	0.71	0.69	0.77	0.73	0.74	0.71
length = 3	0.54	0.48	0.66	0.58	0.59	0.53
length = 4	0.45	0.39	0.63	0.56	0.52	0.46
length = 5+	0.52	0.31	0.60	0.40	0.56	0.35
length = 2+	0.62	0.56	0.71	0.65	0.66	0.60
not nested	0.68	0.66	0.79	0.77	0.73	0.71
nested	0.72	0.64	0.77	0.66	0.74	0.65
with-edit	0.78	0.72	0.72	0.67	0.75	0.70
without-edit	0.67	0.65	0.80	0.77	0.73	0.70

Table 6: *Precision, recall and F-score results for repair start (rps) and reparandum phrase (rm) detection of the difficult-to-detect cases for the STIR model trained on the NICT-JLE corpus.*

Model iteration	Precision		Recall		F-score	
	rps	*rm*	*rps*	*rm*	*rps*	*rm*
length = 1	0.60	0.58	0.84	0.80	0.70	0.67
length = 2	0.71	0.64	0.80	0.71	0.75	0.67
length = 3	0.66	0.55	0.76	0.63	0.71	0.59
length = 4	0.67	0.49	0.69	0.54	0.68	0.51
length = 5+	0.54	0.31	0.52	0.30	0.53	0.31
length = 2+	0.68	0.57	0.76	0.65	0.72	0.61
not nested	0.61	0.59	0.82	0.75	0.70	0.66
nested	0.65	0.56	0.71	0.57	0.68	0.56
with-edit	0.75	0.67	0.79	0.71	0.77	0.69
without-edit	0.59	0.55	0.82	0.76	0.69	0.63
rps-0	0.62	0.57	0.76	0.70	0.68	0.63
rps-1	0.27	0.24	0.80	0.72	0.40	0.36
rps-0 and rps-1	0.36	0.33	0.79	0.71	0.50	0.45
all rps -1	0.32	0.29	0.80	0.71	0.46	0.41

Table 7: *Precision, recall and F-score results for repair start (rps) and reparandum phrase (rm) detection of the difficult-to-detect cases for the DEEP model trained on the Switchboard corpus.*

Model iteration	Precision		Recall		F-score	
	rps	*rm*	*rps*	*rm*	*rps*	*rm*
length = 1	0.85	0.80	0.73	0.71	0.79	0.75
length = 2	0.77	0.66	0.57	0.51	0.66	0.58
length = 3	0.77	0.60	0.45	0.24	0.57	0.34
length = 4	0.74	0.32	0.26	0.06	0.38	0.11
length = 5+	0.00	0.50	0.32	0.02	0.00	0.03
length = 2+	0.77	0.63	0.49	0.37	0.60	0.46
not nested	0.83	0.76	0.65	0.61	0.73	0.68
nested	0.85	0.72	0.62	0.50	0.72	0.59
with-edit	0.74	0.62	0.58	0.48	0.65	0.54
without-edit	0.85	0.78	0.66	0.61	0.75	0.69

Table 8: *Precision, recall and F-score results for repair start (rps) and reparandum phrase (rm) detection of the difficult-to-detect cases for the DEEP model trained on the NICT-JLE corpus.*

Model iteration	Precision		Recall		F-score	
	rps	*rm*	*rps*	*rm*	*rps*	*rm*
length = 1	0.75	0.70	0.77	0.74	0.76	0.72
length = 2	0.65	0.59	0.80	0.72	0.72	0.65
length = 3	0.65	0.57	0.72	0.57	0.68	0.57
length = 4	0.70	0.58	0.66	0.42	0.68	0.48
length = 5+	0.69	0.50	0.45	0.08	0.54	0.14
length = 2+	0.66	0.58	0.74	0.61	0.70	0.59
not nested	0.70	0.65	0.76	0.70	0.73	0.67
nested	0.73	0.67	0.58	0.42	0.65	0.52
with-edit	0.69	0.60	0.76	0.64	0.72	0.62
without-edit	0.72	0.67	0.76	0.71	0.74	0.69
rps-0	0.71	0.62	0.72	0.63	0.72	0.63
rps-1	0.39	0.35	0.67	0.61	0.49	0.45
rps-0 and rps-1	0.49	0.43	0.74	0.66	0.59	0.52
all rps -1	0.45	0.40	0.72	0.64	0.55	0.49

Table 9: *Precision, recall and F-score classification results for repair start (rps) and reparandum phrase detection (rm) for all model adaptations described in Chapter Five. Adaptations are added incrementally, building on the best model (in bold) from each prior section.*

Model iteration (added incrementally)	Repair phrase start (*rps*)			Reparandum phrase (*rm*)		
	P	R	F	P	R	F
baseline	0.716	0.759	0.736	0.654	0.693	0.673
lemmatized	0.715	0.804	**0.757**	0.654	0.735	**0.692**
silence	0.701	0.813	0.753	0.644	0.747	0.692
laughter (non-verbal)	0.716	0.80	0.756	0.654	0.733	0.691
laughed word	0.713	0.801	0.755	0.652	0.733	0.690
laughter (non-verbal) & laughed word	0.716	0.804	0.757	0.652	0.733	0.690
silence & laughter (non-verbal)	0.708	0.815	0.758	0.647	0.745	0.693
silence & laughed word	0.715	0.807	0.758	0.655	0.738	0.694
silence, laughter (non-verbal) & laughed word	0.723	0.802	**0.761**	0.664	0.737	**0.699**
word2vec trained on NICT-JLE (size = 50)	0.723	0.802	0.761	0.664	0.737	0.699
fastText trained on NICT-JLE (size = 50)	0.739	0.800	**0.768**	0.679	0.735	**0.706**
fastText trained on NICT-JLE (size = 100)	0.710	0.809	0.756	0.653	0.744	0.695
fastText trained on NICT-JLE (size = 200)	0.737	0.795	0.765	0.676	0.729	0.702
GloVe pretrained (size = 300)	0.703	0.802	0.749	0.642	0.732	0.684
word2vec pretrained (size = 300)	0.712	0.804	0.755	0.649	0.732	0.688
fastText pretrained (size = 300)	0.726	0.793	0.758	0.665	0.726	0.694
without edit term context	0.767	0.794	0.780	0.717	0.742	0.729
with edit term context	0.752	0.820	**0.784**	0.700	0.764	**0.730**
with proficiency level	0.763	0.813	**0.787**	0.712	0.761	**0.736**