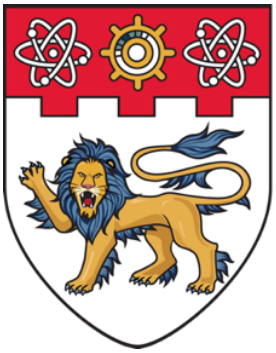


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

USING RICH MODELS OF LANGUAGE IN GRAMMATICAL ERROR DETECTION

LUIS MORGADO DA COSTA

Interdisciplinary Graduate School

Global Asia

USING RICH MODELS OF LANGUAGE IN GRAMMATICAL ERROR DETECTION

LUIS MORGADO DA COSTA

Interdisciplinary Graduate School

Global Asia

A thesis submitted to the Nanyang Technological University

in partial fulfilment of the requirement for the degree of

Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

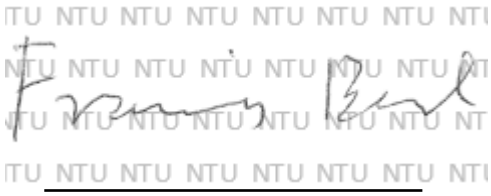
August 19, 2021

Date

Luis Morgado da Costa

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.



August 19, 2021

Date

Assoc. Prof. Francis Bond

Authorship Attribution Statement

This thesis contains material from three (peer-reviewed) papers accepted at conferences in which I am listed as an author.

Parts of Chapter 2 and Chapter 5 are published as Winder, Roger V. P. and MacKinnon, Joe and Li, Shu Yun and Lin, Benedict and Heah, Carmel and Morgado da Costa, Luis and Kuribayashi, Takayuki and Bond, Francis. 2017. NTUCLE: Developing a corpus of learner English to provide writing support for engineering students. *Proceedings of the 4th Workshop on NLP Techniques for Educational Applications (NLPTEA 2017).* Taipei, Taiwan. (IJCNLP 2017 Workshop)

The contributions of the co-authors are as follows:

- Assoc. Prof Francis Bond and Dr. Carmel Heah provided the initial project direction.
- I defined the corpus goals, designed the corpus tagging tool, preprocessed the data, guided the tagging team through the annotation process, and analyzed the tagged data. I'm the maintainer of this corpus.
- The Language and Communication Centre team (Roger V. P. Winder, Joe MacKinnon, Shu Yun Li, Benedict Lin and Carmel Heah) helped defining the error annotation schema and tagged the corpus.
- I prepared the manuscript draft (based on my Qualifying Exam Report) and addressed reviewers' comments. The manuscript was revised by all authors.

Parts of Chapter 8 and Chapter 9 are published as Morgado da Costa, Luis and Winder, Roger V P and Li, Shu Yun and Liang, Benedict Christopher Lin Tzer and Mackinnon, Joseph and

Bond, Francis. 2020. Automated Writing Support Using Deep Linguistic Parsers. *Proceedings of the 12th Conference on Language Resources and Evaluation. European Language Resources Association (ELRA)*. Marseille, France. The contributions of the co-authors are as follows:

- Assoc. Prof Francis Bond, Senior Lecturer Roger V P Winder, and I defined the project requirements and goals.
- I defined, designed and developed the error detection system and web-application. I designed the the evaluation experiment to test this system, and analyzed the results of this experiment. I'm the maintainer of this system.
- The Language and Communication Centre team (Roger V. P. Winder, Joe MacKinnon, Shu Yun Li and Benedict Lin) helped designing the corrective feedback messages produced by the system, hosted the system in their classrooms, and helped carrying the evaluation experiment.
- I prepared the manuscript drafts and addressed reviewers' comments. The manuscript was revised by all authors.

Parts of Chapter 8 are published as Morgado da Costa, Luis and Sio, Joanna Ut-Seong. 2020. CALLIG: Computer Assisted Language Learning using Improvisation Games. *Proceedings of the Games and Natural Language Processing Workshop at the 12th Edition of the Language Resources and Evaluation Conference. European Language Resources Association (ELRA)*. Marseille, France.

The contributions of the co-authors are as follows:

- Assist. Prof. Joanna Sio and I defined the initial project direction;
- Assist. Prof. Sio provided guidance to define the improvisation elements for the available games in the system.
- I designed, developed and am the maintainer of this web-application.

- I prepared the manuscript drafts and addressed reviewers' comments. The manuscript was also revised by Assist. Prof. Sio.

August 19, 2021

Date

Luis Morgado da Costa

Acknowledgments

I would like to thank my Thesis Advisory Committee – Francis Bond, Annabel Chen and Erik Cambria – for the years of continued support, without which the completion of this dissertation would not have been possible. Francis, however, must be mentioned twice. Once for this exemplary support as main supervisor, and a second time as a mentor and friend. Francis welcomed me into his home, introduced me to his family, and made sure to remind me often of the importance of taking breaks and enjoying the journey (usually with an invitation for a game night!). There are no words to fully express the extent of my gratitude for your friendship and support.

I also have some special words of gratitude for Dan Flickinger – first, for inspiring my PhD topic with his work, but also for his friendship, for being so generous with his time and knowledge, for his willingness to take so many of my silly questions and transform them into teachable moments, for inviting and hosting me at the Center for Advanced Study in Oslo, and for helping me train some of my student assistants.

I would also like to thank my colleagues at the Language and Communication Centre (LCC) – especially Roger Winder, who spent so much of his own time helping make this collaboration possible. It was a privilege to work with all of them. Their support made my dissertation stronger. It was a great responsibility but also immensely rewarding to have some of my work tested and incorporated in their classrooms.

I also must thank my family and friends, who supported my decision to move half-way across the globe to pursue a PhD – at the expense of so many missed important moments from which I was *exempted* to attend in person. It was very challenging not to be able to be with you as often as I wanted/needed to, but knowing I had your support made the distance a bit more tolerable. And while I can't name all those to whom I am indebted, some deserve a special mention for

always being there and for helping me when I needed most – even when I failed to ask... to Hannah, Yasu, Fran, Mafalda, Miguel, Elsa, and Celia. Each of you have become an essential part of my life. Near or far, your love and support made even the most difficult days possible.

I also want to thank my fellow members of NTU's Computational Linguistics Lab, for their friendship, for serving as inspiration, for sharing their knowledge, and for the many karaoke parties that made our lab feel like a great big family. To Michael, Sanghoun, Zhenzhen, Giulia, David, Mindy, Kuribayashi, Hanah, Siew Yeng. And a special mention to my dear friend Tuấn Anh, my favorite intellectual sparring partner, and his wife, Van, for her lovely friendship and for putting up with our long discussions over so many cups of coffee.

I would also like to give a warm thanks to Kodrah Kristang and the Eurasian community in Singapore. It has been a pleasure to dedicate much of my little free time to the revitalization of Kristang in Singapore. Through this beautiful language I've made many friends – Fran, Kevin, Fuad, Andre, Gerald, Cass, Brenda and many others (some of whom have since parted). They have opened their homes and hearts to me, and made Singapore feel like a true home.

And my thanks would not be complete without a special mention to my fellow game night companions – Francis, Quen, Hannah, Arthur, Siew Yeng, Mike, Ning, Rachel. Thank you for the friendship, the competition, and for the most fun way of escaping my dissertation when things weren't flowing. You've helped recharge my soul every time it needed it.

Finally, this PhD thesis would also not been possible without the help of multiple generous sources of funding. In particular the work presented here received support from: i) my NTU/Singapore MOE Research Scholarship (RSS); ii) the Singapore MOE Tertiary Research Fund entitled *Syntactic Well-Formedness Diagnosis and Error-Based Coaching in Computer Assisted Language Learning using Machine Translation Technology*; iii) Fuji Xerox Corporation through a joint research project on *Multilingual Semantic Analysis*; iv) and the 2 EdeX Teaching and Learning Grant administered through NTU's Teaching, Learning and Pedagogy Division.

Luis Morgado da Costa

Singapore, 2021

Contents

List of Abbreviations and Conventions

The following abbreviations are used for syntactic classes, functions, and other categories:

1	first person
2	second person
3	third person
ADJ	adjective
ADV	adverb
ASP	aspect
ATTRIB	attributive
CLF	classifier
COP	copula
DEF	definite
DET	determiner
EXCL	exclusive
FEM	feminine
LOC	locative
MASC	masculine
NEG	negation
NP	noun phrase
PART	particle
PL	plural

PP	prepositional phrase
PROG	progressive
QUES	question word
REL	relativizer
SG	singular
VP	verb phrase

HPSG/DELPH-IN Feature Glossary

This glossary defines and explains common terms the reader will see in attribute-value matrices (AMV), as they are commonly understood within HPSG and, more specifically, the the DELPH-IN community. This list is not exhaustive, as it excludes specific features that are explained in detail during the discussion of this thesis. This glossary does not aim to provide final definitions for these terms, as it is well known that different individuals and communities use the same or similar terms with different meanings. The definitions presented here are for the sole benefit of the reader, only features used within this thesis are introduced, and their definitions pertain only to how they are used in this thesis.

AGR: A feature used to define agreement values. It is often a complex feature, and can include multiple sub-features like NUM, PER or PERNUM.

CAT: A complex feature containing all the syntactic properties of a word or phrase. (See SYN)

CONT: A complex feature containing all the semantic properties of a word or phrase. (See SEM)

COMPS: A feature used to define the list of complements still available to fill in a word or phrase. For example, transitive verbs typically start off by having a COMPS list of length one, while a ditransitive verb typically start with a COMPS list of length two.

HEAD: A feature containing information about the word class that determines the syntactic behavior and properties of that word or phrase. It is related with but no the same as a part-of-speech. For example, in English, the value of HEAD for full sentences is typically *verb*, because sentences are generally headed by a *verb*.

INPUT: A feature used in lexical rules, defining the necessary conditions a word needs to be able to be transformed by this kind of rule. (see OUTPUT)

NUM: A feature used to define the grammatical number of a word or phrase. It is typically used inside the complex feature AGR.

PER: A feature used to define the grammatical person of a word or phrase. It is typically used inside the complex feature AGR.

SPR: A feature used to define the list of specifiers available to be filled in a word or phrase. Specifiers can serve multiple purposes, depending on the grammar in question. In traditional HPSG, for example, a specifiers can be the determiner in English noun phrases, or the subject of English verb phrases. (see SUBJ)

SUBJ: A feature used to define the list of subjects a word or phrase can take. This list is usually defined to have length one, and it can be defined as complementary to specifiers by some grammars (see SPR).

VAL: A complex feature that defined the valency status of a word or phrase. This feature usually comprises of features like SUBJ, SPR and COMPS.

OUTPUT: A feature used in lexical rules, defining the properties of a word after it has been transformed by this kind of rule. (see INPUT)

SEM: A complex feature containing all the semantic properties of a word or phrase. Within DELPH-IN, this feature is usually known as CONT (for content).

SYN: A complex feature containing all the syntactic properties of a word or phrase. Within DELPH-IN, this feature is usually known as CAT (for category).

SYNSEM: A complex feature defining both the semantic and syntactic properties of a word or phrase. It usually contains SYN and SEM.

List of Figures

List of Tables

Summary

In this thesis, I show the advantages of using symbolic parsers for Grammatical Error Detection and Correction. In particular, I work with computational grammars for English and Mandarin Chinese to demonstrate how linguistically motivated research using symbolic parsers is still an extremely viable approach to build educational applications.

During the various chapters of this thesis, I will guide the reader through the entire process of creating a successful educational application that has benefited thousands of NTU students.

To this end, I will start by describing the creation of two new learner corpora, one for English and one for Mandarin Chinese, through which I collected first-hand data about common errors NTU students make in these two languages. I will follow with a discussion of my contributions to ZHONG, an open source computational grammar of Mandarin Chinese using a theoretical framework known as Head-Driven Phrase Structure Grammar, with special emphasis on the design of special rules capable of transforming a computational grammar into an error detection system. I will then discuss the creation of a new treebank used to train parse-ranking models to help symbolic parsers decide the most likely correction for a given error. And I will conclude by describing the development of two web-based applications exploiting a mature symbolic parser to provide immediate corrective feedback for a large number of common errors.

This thesis presents multiple sets of positive results. I have not only substantially increased ZHONG's coverage, but I have also successfully implemented dozens of checks to detect common grammatical mistakes made by learners of Mandarin Chinese. Using the new parse-ranking models, I was also able to improve the precision of error detection in both English and Mandarin Chinese by between 15% and 20%. Finally, a blended learning experiment involving more than 1,800 NTU students has shown the success of an application developed specifically to help

improve students' writing.

All developed systems, as well as most of the data collected and tagged during this thesis, are released under open-source licenses.

Part I:

Background

Chapter 1

Introduction

In this thesis, I show the advantages of using symbolic parsers for Grammatical Error Detection and Correction.

In a context where statistical and deep neural methods have been in the spotlight of the large majority of research conducted in Natural Language Processing (Manning, 2015), it is important to step back and evaluate some of the repercussions of these trends. In the field of Computer Assisted Language Learning, in particular, misunderstanding and unrealistic expectations over the quality and performance of statistical-based methods have been largely responsible for fostering a lack of trust in technologies for language teaching and learning (Leacock et al., 2010; Tafazoli et al., 2019).

In this thesis, I work with computational grammars for English and Mandarin Chinese, and demonstrate how linguistically motivated research using symbolic parsers is still an extremely viable approach to build educational applications. At the core of my research agenda is the concept of *mal-rules* (Schneider and McCoy, 1998) – a method that relies on the understanding and manipulation of linguistically-rich models of language, such as computational grammars, to perform grammatical error detection and correction.

Mal-rules are not a quick solution, and require an in-depth understanding of linguistic theory to be implemented. However, because they are only truly viable within linguistically-rich models, mal-rules can also help systems delve deeper into important questions such as: '*Why is a sentence ungrammatical?*', '*How many ways are there to correct an ungrammatical sentence?*,

or '*What are the possible intended meanings behind an ungrammatical sentence?*'.

In this interdisciplinary thesis, I will essentially cover all the steps between identifying a sentence like (1) as a common error among NTU's student population, to the implementation of a web system designed to help improve student's writing. This system is capable of providing immediate corrective feedback for a wide variety of different common errors, like the one in (1) – see Figure 1.The system has now been successfully used by thousands of NTU students, who have directly benefited from this technology.

(1) * *This systems corrects errors.*

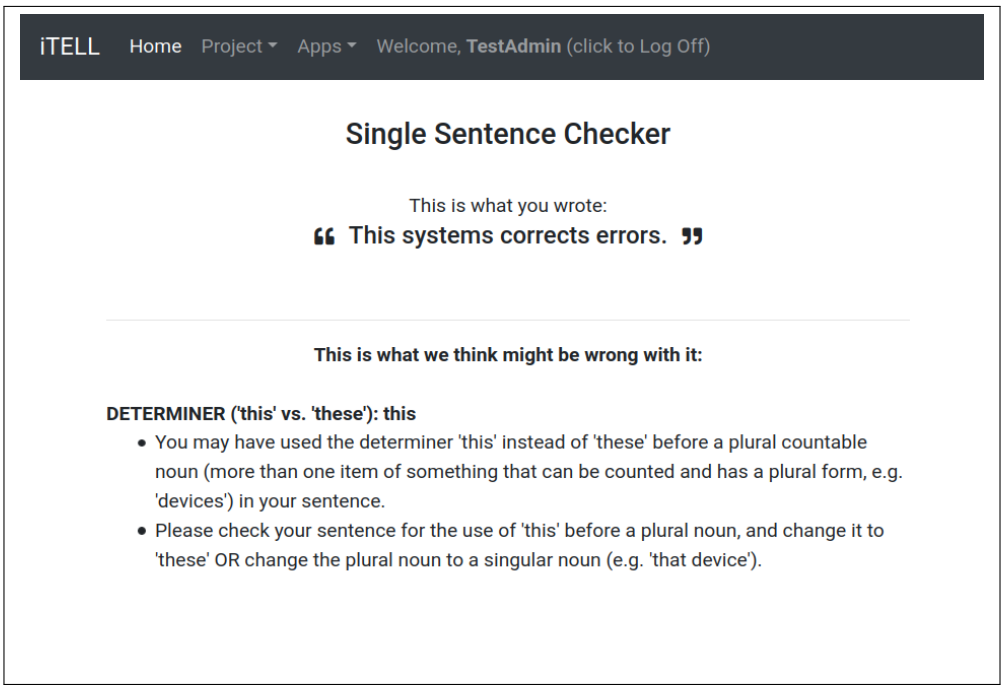


Figure 1.1: Feedback Example

These steps included: collecting first-hand data about common errors to understand which errors were worth detecting; preparing and expanding a symbolic parser to make sure it was ready to be used for grammatical error detection; designing and implementing mal-rules targeting relevant error classes; training parse-ranking models to help symbolic parsers decide the most likely correction for a given error; and building multiple end-to-end educational applications exploiting this technology to help students improve their language skills.

This thesis considers both English and Mandarin Chinese. For English, I have mostly adapted and improved existing work using mal-rules (Bender et al., 2004; Flickinger and Yu,

2013; Suppes et al., 2014). This included working with a team of lecturers to define the pedagogical goals of the system, collecting first-hand data about the common errors NTU students made while writing English essays, and building an end-to-end system that was tested and validated in a blended-learning experiment.

For Mandarin Chinese, where existing technology was less mature, I have both improved an existing Mandarin Chinese grammar (Fan et al., 2015a,b; Fan, 2019) and introduced dozens of mal-rules, effectively transforming it into a full-fledged error detection system.

The goal of this thesis is to establish a vision of what high-quality, hand-curated, and theoretically sound symbolic parsers can contribute to the field of Computer Assisted Language Learning and, with it, challenge the preconceptions that have been created against the use of technology in language learning.

This thesis is structured as follows:

Part I contains the motivation for this thesis, provides background to important theoretical concepts used throughout this thesis, and lays the technical foundation to understand some of these concepts in greater detail.

- **Chapter 2** provides an introduction to the field of Computer Assisted Language Learning, as well as to the sub-fields of Grammatical Error Detection and Correction, Corrective Feedback, Learner Corpora and Gamification.

- **Chapter 3** introduces the concept of Deep Linguistic Processing, along with DELPH-IN – an international consortium supporting the infrastructure used in this thesis. It provides an introduction to Head-driven Phrase Structure Grammar and Minimal Recursion Semantics, two essential frameworks in this thesis, and how these frameworks can be used for Grammatical Error Detection and Correction.

- **Chapter 4** provides an introduction to major tools and language resources used in this thesis, namely the English Resource Grammar (a large computational grammar for English) and ZHONG (a medium-sized computational grammar for Mandarin Chinese). It also provides brief introductions to a hand-full of tools made available by DELPH-IN, as well as the Princeton WordNet.

Part II provides detailed accounts on the actual execution of this thesis' contributions.

- **Chapter 5** provides details concerning the creation of three corpora – the NTU Corpus of Learner English, the NTU Corpus of Learner Mandarin Chinese and the Mandarin Education Corpus.
- **Chapter 6** provides a summary of all contributions made to ZHONG. These improvements include bug-fixes, and theoretical contribution to the analysis of Mandarin Chinese in the form of reanalyses of certain phenomena, and the implementation of missing phenomena. Crucially, mal-rules used to detect common errors made by early learners of Mandarin Chinese are also discussed in depth.
- **Chapter 7** introduces the creation of the Tembusu Treebank – a multilingual treebank of learner and educational data, used to create new mal-rule enhanced parse ranking models to improve error detection and diagnosis for the English Resource Grammar and ZHONG.
- **Chapter 8** provides a description of two web applications that exploit mal-rules in educational contexts: the LCC-APP (and academic writing support system used to help NTU engineering students improve their writing skills) and CALLIG (an experimental web application that explores the concepts of gamification and improvisational comedy to design engaging language games).

Part III contains the evaluation, discussion and concluding statements for this thesis.

- **Chapter 9** contains a series of experiments evaluating the LCC-APP in a blended learning experiment, ZHONG's improved parsing coverage, and the performance of error detection and diagnosis for the English Resource Grammar and ZHONG using the new parse ranking models trained during this thesis.
- **Chapter 10** provides a general discussion of the findings of this thesis, especially in relation to limitations and ideas for future work.
- **Chapter 11** contains a few concluding remarks along with a summary of major contributions.

Part IV contains the bibliography.

Part V contains three short appendices that point readers to the source code, give a list of my publications, and present the error tag set used in the NTU Corpus of Learner English.

Chapter 2

Computer Assisted Language Learning

In this chapter I will provide an overview of the major sub-fields of Computer Assisted Language Learning (CALL) that are relevant to this thesis. I will first provide a brief introduction and background to the field of Computer Assisted Language Learning. This will be followed by a short discussion of the task of Grammatical Error Detection and Correction, along with the introduction of a core concept of this thesis: mal-rules. The chapter will conclude with brief introductions to Corrective Feedback, Learner Corpora, and Gamification – along with their relation to the field of CALL and the work presented in this thesis.

2.1 Computer Assisted Language Learning

Since the early 1960s, with the advent of the personal computer, computer mediated education has become a popular interdisciplinary field, intersecting the fields of education and computer science with a variety of domain specific fields, e.g., mathematics, natural sciences and, of course, language. The sub-field of computer mediated language education is often referred to as Computer Assisted Language Learning, or CALL.

In the past decade or so, it has become evident that Learning Sciences are rapidly entering a new era of online mediated education. Many of the main players in the worldwide education system have identified the need of belonging to this new virtual learning space – embracing the trend of digital transformation (Benavides et al., 2020), often developing their own digital platforms or adopting open platforms to support technologically enhanced education.

Even though concepts like Massive Open Online Courses (MOOCs) have only been around for a bit more than a decade, this new learning paradigm has already caused an unprecedented change in worldwide education, especially in Higher Education (Yuan et al., 2013). Unfortunately, the number of Language Massive Open Online Courses (LMOOC) available is proportionately very small (Perifanou and Economides, 2014; Sallam et al., 2020). Sallam et al. (2020) report that fewer than 200 language related MOOCs (LMOOCs) existed in a 2018 survey (less than 1.8% of the total number of courses surveyed in that year). There is, arguably, not a lack of demand for such language courses, but a lack of the technological infrastructure to support them.

LMOOCs share many of the challenges MOOCs face. Yuan et al. (2013) identify 'sustainability', 'pedagogy', 'quality and completion rates', and 'assessment and credit' as the four main challenges that all MOOCs must struggle with. From these, Perifanou and Economides (2014) foreground the problems of 'pedagogy' and 'assessment' as being especially challenging to LMOOCs. How to best instruct and assess language content in a way that is scalable to thousands of students is an unsolved problem. Sallam et al. (2020) argue that LMOOCs are best suited to improve receptive skills (i.e., listening and reading), and that the lack of spaces of interaction – both between peers and between teachers and students – is a key limiting factor to the creation and uptake of LMOOCs. And Vorobyeva (2018) discusses the lack of technological infrastructure to support key aspects of language learning as a key problem – framing the inability to provide high quality feedback for key aspects of language (e.g., grammar and pronunciation) as one of the main limitations faced by LMOOCs.

While there has been a considerable amount of research conducted in CALL in the last decades, designing an efficient language learning course or developing a language learning platform is still a very complex process (Perifanou and Economides, 2014).

CALL is also a central topic in Intelligent Language Tutoring Systems (ILTS). ILTSs are educational systems where the computer tries to mimic the role of a teacher or lecturer by steering and delivering content in an interactive manner (Tafazoli et al., 2019), often with a big emphasis on corrective feedback (Al Emran and Shaalan, 2014) – i.e., having the ability to perform an assessment of the quality of student input, and to provide timely corrective feedback in case the

system detects problems.

ILTSs make use of Artificial Intelligence and, in particular, Natural Language Processing (NLP) techniques to focus on problems like user modeling, grammatical error detection and classification, semantic analysis, etc. (Schulze, 2008; Gamper and Knapp, 2002; Al Emran and Shaalan, 2014; Tafazoli et al., 2019). An earlier survey by Gamper and Knapp (2002) recognizes a large variability between systems, and acknowledges that ILTSs vary immensely by the features they possess. Some systems have linguistic domain knowledge, allowing detailed feedback to be provided to the learner, while others just guide students through a virtually designed course. Some include adaptive user models, and a few incorporate automated speech synthesis and recognition. Some systems focus on one basic language skill (e.g., reading, writing, listening, or speaking), while others look for broader coverage. Some systems have a larger focus on grammar, others on vocabulary, and some even specialize in dialog interaction.

While it is important to understand that ILTSs can differ immensely (sometimes becoming even difficult to compare them), it is perhaps more important to understand that these systems are simply another medium for language teaching – and thus, to some sense, comparable and competing with all other existing mediums (e.g., a human lecturer, a textbook, a set of exercises, etc.). An important study conducted by Nagata (1996) showed that it is not the medium itself (e.g., a computer vs. a book) that determines success in learning, it is the quality of the feedback produced by that medium that affects the results. This is why a language teacher is likely to be a better medium than a book, and the same reason why a properly designed ILTS can be a better medium than a book and, in principle, also better than a human – assuming, of course, the unlikely case where a system is able to provide better feedback than a human.

Tafazoli et al. (2019) provide an updated history and survey of ILTSs, especially in relation to early and simpler CALL systems. In this survey, they also highlight the problems with trust and uptake of these systems by lecturers, ascribing a lot of misunderstanding and unrealistic expectations over the quality and performance of these systems to the fact that the teaching community does not trust them. Leacock et al. (2010) note a similar trend: blaming inflated expectations to the growing idea that NLP techniques are and will always be unable to deal with the full complexity of language (and hence should be avoided in educational applications).

While the maturation of the field of NLP has indeed helped produce useful systems, this maturation is still an ongoing process. Even though there are some language skills where the relentless persistence of ILTSs can indeed help or compete with human lecturers, there are also many aspects of language learning where the current technology does not begin to compare with a human lecturer. On this topic, Gamper and Knapp (2002) point out that most ILTSs concentrate mainly on syntax and give less attention to semantic components, and only very few try to address the problem of pragmatics – since this is an extremely hard problem. This can be explained, fairly straightforwardly, by the limits of current NLP techniques, which have historically focused more on morpho-syntax because it is easier (though far from simple) to model computationally. One of the uses of building morpho-syntactic models is the ability to perform grammatical error detection and correction, which will be discussed in greater detail below.

2.2 Grammatical Error Detection and Correction

Automated Grammar Error Detection and Correction are tasks that have attracted some attention within the NLP community soon after CALL emerged as research field. This is especially true for English, where a myriad of shared tasks periodically compare and attest the impact of the latest available technology. Some recent efforts in organizing shared tasks within these topics include: the 2011 Helping Our Own (Dale and Kilgarriff, 2011, HOO) shared task on Grammar Error Correction; the more focused 2012 HOO shared task on Preposition and Determiner Error Correction (Dale et al., 2012); the 2013 and 2014 CoNLL shared tasks on English Grammar Error Correction (Ng et al., 2013, 2014); the 2016 shared task on Automated Evaluation of Scientific Writing, focusing on error detection (Daudaravicius et al., 2016a, AESW); and, most recently, the BEA-2019 Shared Task on English Grammar Error Correction (Bryant et al., 2019).

Similar efforts to provide support for Mandarin Chinese Grammar Error Detection and Correction have also emerged. Pioneering this effort is the shared task organized by the Natural Language Processing Techniques for Educational Applications (NLP-TEA) held from 2014–2020 (Yu et al., 2014; Lee et al., 2015, 2016b; Gaoqi et al., 2017; Rao et al., 2018, 2020). Rao and

Lee (2018) provide an overview of all previous Mandarin Chinese Grammar Error Detection tasks, drawing attention to the intrinsic difficulty of this task, and the long road ahead.

However, despite becoming an increasingly popular topic, there are also problems with the way this topic has been embraced by the research community, especially concerning its adequacy to educational contexts. The majority of these tasks have a fairly shallow approach to the concept of error detection and correction. For example, in the 2014 CoNLL shared task on English Grammar Error Correction (Ng et al., 2014), the main example presented to summarize the task is shown as (2).

(2) * *Social network plays a role in providing and also filtering information.*

In this shared task, systems had to detect that there was something wrong with the sentence and then had to be able to transform this sentence into the gold correction shown in (3).

(3) *Social networks play a role in providing and also filtering information.*

One of the main problems with the design of this shared task is that it does not require systems to have or produce any kind of linguistic domain knowledge about the error in question. This basically means that these systems would never be able to accurately describe what is wrong with the sentence (i.e., provide a meta-linguistic description of why this is an error, or what could be done in order to correct it – see Section 2.3 on corrective feedback, below).

Another related problem is the fact that the correction shown in (3) is not the only possible correction for the ungrammatical sentence shown in (2). In fact, there are multiple ways of correcting this sentence, a few extra corrections are provided as (4), (5) and (6).

(4) *The social network plays a role in providing and also filtering information.*

(5) *A social network plays a role in providing and also filtering information.*

(6) *Social networks played a role in providing and also filtering information.*

Not acknowledging the fact that a sentence can be corrected in multiple ways hurts the spirit of the task, and further incentivizes systems to use shallow methods such as N-gram based

replacement models, machine translation models and transformer models based on neural networks – as was observed in the last few shared tasks (Ng et al., 2014; Bryant et al., 2019). In these tasks, some systems are able to come up with plausible corrections that are ignored in the evaluation metrics because they did not match the gold correction provided by the shared task. This effectively discourages teams from building holistic systems, capable of providing multiple corrections of a single sentence, due to the risk of not being able to adequately choose the corrections chosen by the task.

In the 2014 CoNLL shared task on English Grammar Error Correction (Ng et al., 2014) the task organizers allowed systems that were sure they came up with viable alternative corrections to self-report a second version of the scores, including alternative corrections found by their systems. However, in the aftermath of this task, Ng et al. (2014) actually "recommend[ed] that [in the future] evaluation be carried out in the setting that does not use alternative answers, to ensure a fairer evaluation".

This reveals a profound disconnection between the current trends in grammatical error detection and correction and the the field of CALL. The current design of shared tasks in grammatical error detection and correction are producing systems that are most suitable to be integrated into tools such as word processors, or the like. The inability to correctly diagnose the source of an error prevents these systems from adequately integrating into tutoring systems, since they can not provide useful feedback.

The heavy dependence on statistical methods and biased datasets also raises problems concerning flexibility and granularity in this task. Flexibility concerns itself with being able to detect different kinds of errors, including rare or infrequent errors (e.g., particular to a specific population of students), or with being able to detect new types of errors that were not previously included in their datasets. Granularity concerns itself with the level of 'explainability' contained in the annotated data. If, for example, multiple errors concerning the use of determiners are identified as a single error tag, data-driven methods will never be able to discern between different kinds of 'determiner errors'.

Systems using statistical methods are often called 'black boxes', as they do not possess any kind of first order linguistic knowledge about their decision making, which is essential to be

able to explain why a sentence is ungrammatical.

Finally, the current shared tasks are defining a trend that sees automated correction as the ultimate goal of grammatical error detection and correction. However, this contradicts an important idea in educational research that feedback (and more importantly its quality) is an essential vector through which learning happens and can be improved (Nagata, 1996). Although automated correction is a suitable goal for a variety of use cases, it is not necessarily so in educational contexts (see Section 2.3 on corrective feedback).

These trends are perpetuating the problems of a chronic lack of technological infrastructure for computer-assisted language education, and of trust issues raised and held by language educators.

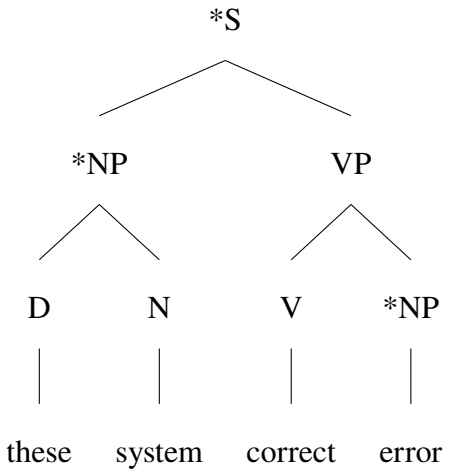
Mal-rules

In response to the limitations raised above, this thesis will make use of a concept known as mal-rules, sometimes also called 'error-production rules'. The concept of *mal-rules* was first proposed by Schneider and McCoy (1998). These rules are used to extend descriptive grammars in order to allow specific ungrammatical phenomena to be parsed, while reconstructing structures that were violated. Although the design of mal-rules is time consuming, they enable fine-tuned error distinctions that shallower parsers would have a hard time dealing with.

Mal-rules can enable CALL systems to answer more challenging questions about ungrammatical sentences, such as: '*Why is a sentence ungrammatical?*', '*Is there more than one way to correct a specific ungrammatical sentence?*', or '*What are the possible intended meanings behind an ungrammatical sentence?*'.

Consider example (7), below. A descriptive grammar of English should reject (7) as a proper sentence. However, the decision of how to correct this sentence is not a simple one. Without context, at least four corrections (8) to (11) should be considered – but other options are probably also available. From a pedagogical point of view, each of these corrections should elicit different kinds (or sets) of corrective feedback.

(7) * *These system correct the error.*



(8) *These systems correct the error.*

(9) *These systems correct errors.*

(10) *This system corrects the error.*

(11) *This system corrects errors.*

A few mal-rules would allow this sentence to be parsed while reconstructing all of the meanings shown above. This would require, for example: i) a mal-rule that allows disagreement between nouns and determiners (to license the NP *these system*); ii) a mal-rule to either add a quantifier or change *error* into plural (to license the VP *correct error*); and, iii) a rule that would allow subject-predicate disagreement, so that both *this system* and *these systems* can be possible interpretations of the subject. Chapter 3 will provide a formal account of how mal-rules can be designed using Head Driven Phrase Structure Grammar (Pollard and Sag, 1994; Sag et al., 1999), along with a more in-depth account of how multiple mal-rules interact with each other.

mal-rules can act on syntactic structures and on individual lexical items. A system using mal-rules can perform both error detection and error correction (since ungrammatical structures can be reconstructed). From a pedagogical point of view, each mal-rule activation can be converted into a constructive feedback message (e.g., *'the subject and the verb in your sentence do not agree in number'*). Mal-rules can be as specific as necessary, and hence are able to accommodate different kinds of corrective feedback – discussed in greater detail below.

When compared to shallower statistical methods, mal-rules have both advantages and disadvantages. The main disadvantage is, most definitely, coverage. The creation of mal-rules presupposes the existence of theoretically inspired computational grammars, which can take a long time to develop. When coverage is not an issue, however, mal-rules can perform error detection and correction with much higher quality, using richer models of language to provide great flexibility and granularity in error detection and correction, while also providing linguistic domain knowledge to support extremely detailed forms of corrective feedback.

In this thesis, I will explore mal-rules in two main ways. First, I will use a computational grammar of Mandarin Chinese to develop mal-rules that model many common errors of Mandarin Chinese learners – this will be presented as part of Chapter 6. Second, I will exploit previous work conducted in creating mal-rules for English to build an online writing support system designed to help students with English Scientific Writing – this system will be introduced in Chapter 8.

2.3 Corrective Feedback

The notion of Corrective Feedback (CF) – that which is said about an identified error to help a student/user correct that particular error – is intrinsically tied to the idea of Grammatical Error Detection and Correction, especially in the context of language education.

There is a great deal of evidence that shows the benefits of corrective feedback, however it is important to acknowledge that CF research is still somewhat controversial, and its benefits not universally accepted (for a summary of this discussion see, e.g., Van Beuningen, 2010). Most of the controversy surrounding CF and its potential benefits arise from theoretical perspectives concerning Second Language Acquisition (SLA). More specifically, these discussions are tied to theoretical assumptions concerning implicit and explicit learning, and the assumed role that CF can have in each of these.

Despite these disagreements, the evidence that CF provides effective means of improving learner's writing is both quantifiable and long-lasting (Van Beuningen, 2010). In addition, in the context of English for Academic Purposes, where this thesis will further explore the concept

of corrective feedback, the benefits of written corrective feedback need not be limited by SLA theories. As Ferris (2010) points out, although L2 Writing could be seen as a sub-discipline of SLA, it has crossed disciplinary boundaries and now has issues and goals of its own. It is not surprising, therefore, that most L2 writing teachers and researchers agree that CF has a role in L2 Writing instruction (Ferris, 2010).

L2 Writing instruction is closely related to the task of assisting academic writing, which also includes a range of copy editing issues (e.g., style, voice, word-choice, etc.) that go beyond strictly grammatical issues. The role of CF in L2 Writing should be seen as helping students improve the overall effectiveness of their writing and are, therefore, not strictly bound to SLA theories.

Finally, it is also important to point out that L2 Writing and SLA also present themselves on opposite sides concerning their preference of indirect over direct CF. While SLA research prefers direct CF – where errors are explicitly corrected for the students – L2 Writing argues in favor of the benefits of indirect approaches (Lalande, 1982; Ashwell, 2000; Ferris, 2010) – only highlighting the error (through more or less explicit means), engaging the learners in the correction process, and allowing them to take full ownership of the writing process.

However, both direct indirect and corrective feedback face challenges. The use of direct CF, while often preferred by students, forces the decision of how to correct an error onto the lecturers (or a computational system). And while human lecturers are fairly good at this task, error correction is intrinsically ambiguous. There are studies that show that even human lecturers are prone to errors (Lee, 2004), and this has been used to justify a preference for more indirect methods. On the other hand, using indirect CF most often assumes that students possess sufficient linguistic knowledge to understand and self-correct the errors on their own (Eslami, 2014), and this is, unfortunately, not always the case.

In the particular context of this thesis, some of the ideas discussed here concerning corrective feedback will be revisited to motivate the design choices of the LCC-APP, an online writing support system designed to help students with English Scientific Writing introduced in Chapter 8. And even though this thesis does not focus particularly on the design or on measuring the quality of corrective feedback, some of the results presented in Chapter 9 will, once again, touch

on this topic by posing some important questions and suggesting future directions of research.

2.4 Learner Corpora

First language transfer is widely accepted to play an important role in foreign language learning (Gass, 1988). Because of this, many CALL systems have been implemented for pairs of languages (i.e., a specific source language is considered in the development process) (Gamper and Knapp, 2002). In order to be able to check and correct grammar, CALL systems need to be aware of the most common mistakes its users are known or likely to make. For instance, missing the copula *be* is a common mistake made by native Chinese speakers learning English (Schneider and McCoy, 1998). Similarly, using an unnecessary copula (是 *shì*) in adjectival predication constructions is a common mistake made by native English speakers learning Mandarin.

The study of learner corpora focuses on the collection and analysis of language learner data. This data is especially of interest to CALL research if it has been error-tagged (Granger, 2003) – i.e., all the errors in the corpus have been identified and described. Before one can hope to design error detection and correction systems, it is necessary to survey errors contained in language-specific learner corpora (Granger, 2003). Another important use for learner corpora is the ability to check the appropriateness of the error detection and/or correction of CALL systems, which is usually a valuable metric for the intrinsic evaluation of these systems (Schulze, 2008).

Even though producing an error-tagged corpus can be very time-consuming, the huge return on invested resources is undisputed. For instance, documented and organized data can be used to customize exercises in accordance with the learners' proficiency level and/or mother tongue background (Granger, 2003). Semantically annotated Learner Corpora are a good resource to predict the intended meaning of students (Hellan et al., 2013). And finally, the ungrammatical inputs collected by learner corpora can also be useful examples of unparseable sentences for computational grammarians.

The work in this thesis focuses on English and Mandarin Chinese. There are fairly large learner corpora projects for English – e.g., the NUS Corpus of Learner English (Dahlmeier et al., 2013a) or the Cambridge Learner Corpus (CLC, Nicholls, 2003). The main problem with

these corpora are their restrictive licenses.

The situation for Mandarin Chinese is even more problematic. Not only are there fewer text-based learner corpora, but the ones that exist are either not freely accessible, or designed with very narrow tasks in mind. The Jinan Learner Corpus (Wang et al., 2015a) seems to no longer be accessible online, and the iCALL Corpus (Chen et al., 2015) is a speech corpus mostly concerned with errors in pronunciation. The TOCFL Learner Corpus (Lee et al., 2018), the HSK Dynamic Composition Corpus[1] and the Lang-8 corpus (Mizumoto et al., 2011) are three learner corpora containing data for written Mandarin Chinese. Unfortunately, they are released under restrictive, non-commercial, non-redistribution licenses. In addition, the TOCFL includes only four very broad error types: 'redundant words', 'missing words', 'word selection errors', and 'word ordering errors'. The Lang-8 corpus is an automatically collected corpus providing only pairs of sentences and their respective corrections.

The large majority of learner corpora have not embraced the value of open data, and most often only allow the use of the data for academic purposes while restricting other uses or redistribution of the data. These licenses effectively prevent further work to be done on these corpora (e.g., addition of new error tags, treebanking, etc.), as they are only meant to be used 'as is'. In addition, some corpora have been created with specific tasks in mind, using very few and/or narrow error codes, and end up being suitable only for the contexts they were created for (e.g., shared tasks).

Another issue that arises from reusing existing learner corpora is the fact that the student population used to collect the data is extremely important, and often includes differences across geographic locations, language background, proficiency levels, and even the sensibility of the language teachers judging the appropriateness of a language.

When the purpose of a learner corpus is to build a system that targets a specific population, as was the case in this thesis, it becomes very important to confirm that the general trends found elsewhere are also found in the target population. For these reasons, the research for this thesis includes the construction of two new Learner Corpora – for English and Mandarin Chinese – described in great detail in Chapter 5.

[1] http://yuyanziyuan.blcu.edu.cn/en/info/1043/1501.htm

2.5 Gamification

Despite being a relatively young field, gamification of learning has become a trending topic in recent years. And, as the number of papers published on gamification grows quickly (Hamari et al., 2014), so does general public awareness and peer scrutiny of its effectiveness.

Gamification is broadly understood as the *use of game design elements in non-game contexts* (Deterding et al., 2011). These can include game mechanics, game dynamics, and frameworks, such as badge or point reward systems, time constraints, limited resources, turn taking, interaction, competition, role-playing, etc. – integrated in a way that encourages users to achieve some desired learning goals (Tu et al., 2015; Deterding et al., 2011).

An extensive literature review presented by Hamari et al. (2014), looking at the effectiveness of gamification, suggests that gamification does work, despite also suggesting that more rigorous methodologies ought to be used to further research on gamification. Moreover, gamification can be used for multiple domains of learning, including declarative knowledge, conceptual knowledge, rule-based knowledge, and procedural knowledge (Kapp, 2012).

Gamification in CALL, even though not entirely new, is still widely unexplored. Nevertheless, a few CALL platforms must be acknowledged due to their popularity. Duolinguo[2] is one of such applications. Duolinguo is a free mobile and web-based platform, where users can learn dozens of different languages through vocabulary and translation-based exercises (Garcia, 2013). It presents gamification elements such as badges, point systems, leader-boards, and a skill tree for users to progress through, to name a few. Two other systems, very similar in nature, are Memrise[3] and Quizlet.[4] These two are free mobile and web-based platforms focusing on learning through digital flashcards. Learning through flashcards is widespread in language learning, though in and on its own, it is not specific to language learning. This kind of learning method is known to aid vocabulary retention (Kornell, 2009), which has undoubtedly contributes to its popularity. Both platforms also include gamification elements such as point systems, leader-boards, time constraints, along with a few different games to explore and learn

[2] www.duolingo.com/
[3] www.memrise.com/
[4] www.quizlet.com/

the content of the flashcards. Most language learning platforms available today share, in great part, a lot of the mechanics and goals of the applications mentioned above.

Even though the work presented in this thesis will not focus in great detail on the various aspects of gamification, I will present some exploratory work done in collaboration with Dr. Joanna Sio, a linguist and established stand-up and improv comedian. I will present how the technology developed as part of this thesis (initially targeting more formal settings of language educations) is being adapted to create improvisation-based language learning games. This collaboration was embodied in a suite of mini-games known as CALLIG (Computer Assisted Language Learning using Improvisation Games), which will be introduced in Chapter 8.

2.6 Summary

In this chapter I have introduced some of the main sub-fields of Computer Assisted Language Learning. In particular, I've provided an in-depth discussion of the current trends and limitations in the field of automated Grammar Error Detection and Correction, highlighting how it has strayed away from core principles of CALL due to the trend to use shallow statistical methods that limit the ability to provide meaningful corrective feedback essential in educational contexts.

I've introduced the concept of mal-rules as a method that offers a suitable solution for many of the problems raised for shallow statistical methods. And I've also provided brief introductions to the topics of Corrective Feedback, Learner Corpora and Gamification.

All these topics will be revisited throughout this thesis. The main goal of this chapter was to provide the reader with a brief introduction of these topics, as well as pointers to how they are related and their relevance to the work presented in this thesis.

Chapter 3

Grammar Engineering

This chapter contextualizes how the work conducted during this thesis is related to the field of Grammar Engineering. It introduces the DELPH-IN Consortium, within which this thesis is developed. It describes the standard methodology used in Grammar Engineering development, which was followed throughout this thesis. And it also provides brief introductions to Head-Driven Phrase Structure Grammar (HPSG) and Minimal Recursion Semantics (MRS) – the syntactic and semantic formalisms adopted by DELPH-IN. The chapter concludes with an introduction of how deep linguistic processing with HPSG can be used for robust parsing – in particular, for error detection and correction using mal-rules.

3.1 The DELPH-IN Consortium

The Deep Linguistic Processing with HPSG Initiative (DELPH-IN) is an international consortium that shares a commitment to develop open-source resources for deep linguistic processing.

Deep linguistic processing (Uszkoreit, 2004; Baldwin et al., 2007) is often defined in contrast to 'shallow' linguistic processing and, in particular, by the amount of linguistic information available to parsers and to NLP tools/applications built on top of them. There is a broad consensus that different NLP problems require different levels of information to be available. Deep linguistic methods draw on implementable frameworks in theoretical linguistics (e.g., Head-Driven Phrase Structure Grammar, Lexical Functional Grammar, Tree-adjoining Grammar, etc.) to provide richer syntactic and/or semantic structures than shallow methods. His-

torically, deep linguistic methods were forced to sacrifice efficiency and robustness in order to provide deeper and richer structures. As time went by, usages and applications that required efficient and robust parsers helped shallow parsing techniques take over much of the research conducted in NLP culminating, more recently, with what has been described as the *deep learning tsunami* (Manning, 2015) in natural language processing.

However, it has also become clear that this divide between statistical and rule-based methods has been slowly closing (Uszkoreit, 2004). Deep linguistic processing has embraced statistical components in its pipeline (e.g., in parse selection or in generating robustness through the creation of generic lexical entries based on statistical part-of-speech tagging, etc.). At the same time, statistical methodologies are also starting to come to terms with the the limitations and pitfalls of using purely statistical approaches to solve NLP problems. The closing of this divide has benefited from mediatic contributions by prominent figures in the field (see, e.g., Manning, 2015; Bender et al., 2021).

DELPH-IN partners (including NTU) dedicated many years towards open source multilingual parallel grammar development using Head Driven Phrase Structure Grammar (HPSG, Pollard and Sag, 1994; Sag et al., 1999) and Minimal Recursion Semantics (MRS, Copestake et al., 2005; Copestake, 2007) – which will be introduced below.

Some DELPH-IN partners have also worked on speeding up the creation of these grammars through the LinGO Grammar Matrix project (Bender et al., 2002) – an open-source starter-kit for the development of broad-coverage HPSG grammars. This project not only reduces development time for new grammars, but also ensures a level of compatibility between the syntactic and semantic structures produced by each grammar, which is highly desirable for multilingual projects.

Currently, DELPH-IN has multiple grammars at different stages of development. There are large 'resource' grammar projects for English (Flickinger, 2000; Flickinger et al., 2000; Copestake and Flickinger, 2000), Spanish (Marimon, 2010), German (Müller and Kasper, 2000; Crysmann, 2003, 2005), Japanese (Siegel and Bender, 2002; Siegel, 2006; Siegel et al., 2016), Norwegian (Hellan and Haugereid, 2003) and Korean (Kim et al., 2011), as well as medium-sized and smaller experimental projects for many other languages.

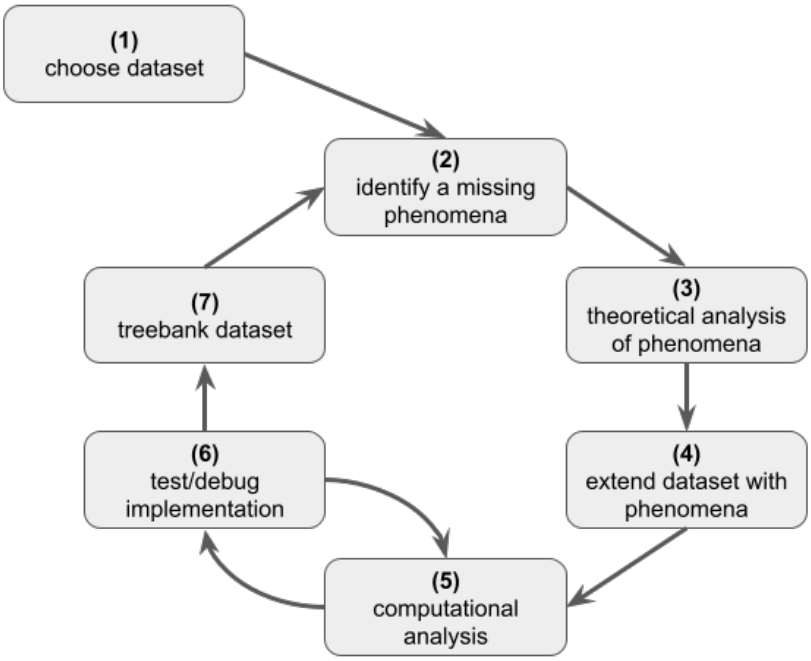


Figure 3.1: Grammar Engineering Workflow

In addition to developing grammars, many DELPH-IN members develop open-source tools that support the necessary infrastructure to build and exploit these grammars. These tools include grammar development and documentation tools, efficient parsers, treebanking platforms, and even Python libraries to analyze and manipulate the output produced by the grammars – many of which will be introduced in greater detail in Chapter 4.

3.2 Methodology: Grammar Engineering Workflow

Grammar engineering uses an interdisciplinary methodology between linguistics (i.e., theoretical analysis) and spiral models common in software engineering (see, e.g., Boehm, 1988).

In the context of grammar engineering, these spiral models follow an incremental and data-driven development cycle adapted specifically for linguistic hypothesis testing (Bender et al., 2008). Figure 3.1 shows a visual depiction of the spiral methodology applied to grammar engineering, adapted from Bender et al. (2011):

1. The process starts by identifying a set of data (i.e., an 'initial test suite'). This set of data

usually has a particular interest to the grammarian or the project in question. In the case of this thesis, this data consisted mostly of educational materials and data produced by students (i.e., learner corpora) – discussed in detail in Chapter 5;

2. The following step is to identify a single phenomena of interest in the data (this can also be a couple closely related phenomena). This is usually done through parsing exercises to show which sentences lack an adequate analysis. In the specific context of this thesis, a common grammatical mistake can also be a phenomena of interest. A discussion of how to provide theoretical analyses for grammatical errors will be introduced in detail later;

3. The third step in this cycle is the theoretical analysis. This analysis usually employs multiple methodologies across linguistics, including surveys of previous work, elicitation of data, and the use of a theoretical framework to try to balance generalization and explicability of the data;

4. Once a foundation of the new analysis is developed, it is often useful to extend the original dataset to include other relevant examples of the phenomena in question (most phenomena include important but infrequent interactions that are missing from average-sized datasets);

5. The fifth step in this cycle is the implementation of the formal/theoretical analysis in the computational grammar. In this step, the theoretical analysis may change slightly due to impositions or previous design decisions included in the grammar. This step may include the revision of certain rules to account for certain phenomena, or the addition of completely new rules (e.g., mal-rules);

6. The computational implementation is debugged by parsing a selected number of sentences. The goal of this functional testing is to ensure that the new phenomena are implemented successfully. Ideally, a larger collection of sentences is also used as a regression test to ensure that these changes did not adversely affect other parts of the grammar. Depending on the results of these tests, there may be a small loop between steps (5) and (6) if the implementation needs to be revised multiple times;

7. The final step of this cycle is to thoroughly test the implementation by treebanking the extended dataset. Treebanking consists of using the grammar to create an exhaustive linguistic analysis for each sentence in the set. The purpose of this process is two-fold:

   (a) treebanking can be seen as an exhaustive test, that tests the new or improved implementations of a phenomena in the context of other phenomena that might not have been present during the initial stages of debugging;

   (b) sentences that cannot be fully analyzed during treebanking will include phenomena that need to be further developed in the grammar. Identifying missing phenomena during the treebanking process effectively restarts the cycle from step 2;

Data Sources:

The first, third and fourth steps of the cycle described above require the use of data. This thesis used multiple types of data sources.

As already mentioned above, the data used in the first step consisted mostly of corpora developed specifically for this thesis. This included data from educational materials and data collected from students – both of which will discussed in detail in Chapter 5. The data for these corpora was carefully selected because this data effectively constrained the level of complexity of the language to be analyzed. Without such constraints, a grammar engineering project can become too broad, since there is no clear end for the task in question. Choosing an adequate dataset is, therefore, essential to manage grammar engineering projects. For the purpose of this thesis, language proficiency was the main dimension to select data.

The data used for the third and fourth steps are different in nature. In order to work on a theoretical analysis for a given phenomena, it is important to consult previous research on the same and similar topics (often including similar phenomena in other languages). This usually includes, at least, traditional sources of written data such as linguistic publications and published grammars. In some particular cases, as it happened with this thesis, language informants can also be used to elicit data to test hypotheses. For this thesis, this was be done through iTalki[1] – an online platform providing access to native speakers and certified language teachers. All users in

[1] `https://www.italki.com/`

this platform have a self-determined cost for an hour of their time. iTalki was used to collect and study native speaker intuitions about Mandarin syntax, and to discuss with Mandarin Chinese teachers about common mistakes students make. All data was collected with the consent of involved parties, who were made aware of the goals of this project.

3.3 Head-driven Phrase Structure Grammar

Head-driven Phrase Structure Grammar (HPSG, Pollard and Sag, 1994; Sag et al., 1999) is a monostratal, declarative, constraint-based linguistic framework capable of representing linguistic information, of both words and phrases, in a single typed-feature structure, or *sign*. In HPSG, signs contain phonological, syntactic and semantic information in a single typed-feature structure, allowing an interface between all these layers of information while building a grammar.

HPSG associates hierarchies of types with feature structures to define linguistic objects. As a constraint-based framework, it makes use of feature constraints (e.g., [NUM *singular*] or [NUM *plural*]) to define what is a valid linguist structure. This happens through *unification*, which is the operation that defines if the properties of two different types are consistent with the type hierarchy (i.e., can be licensed by the grammar) or if they should be rejected.[2]

One core aspect of HPSG is its type hierarchy. A grammar's hierarchy defines a massive chain of inheritances (often multiple inheritances) to define both lexical types and rules. Through such hierarchies, HPSG grammars can keep information organized, capturing similarities between types as much as possible, and making differences among types explicit only when needed. For example, it is plausible to assume that a toy grammar of English would say that *mass noun* and *count noun* are both sub-types of *common noun*. And that *count noun* could be further split into *single count noun* and *plural count noun*. This hierarchy would allow the grammar to add constraints only at the level they are strictly necessary to differentiate these classes. For instance, it is safe to assume that all types under *common noun* are 'nouns' (i.e., share the feature constraint [HEAD *noun*]). Similarly, all types inheriting from *common noun* can take a determiner as specifier (some of types would require it, but all types would allow it).

[2]Readers that are not familiar with HPSG or the meaning of its common features may find helpful a quick revision of the glossary provided in the beginning of this thesis.

However, some constraints must be used to distinguish between *mass noun* and *count noun* – the English Resource Grammar (ERG) uses the feature 'IND', from 'individuated' (to mark whether a noun is essentially countable or not). Following the ERG, *count noun* would have [IND +] as a constraint, while *mass noun* would have the opposite, [IND -]. A similar thing would need to happen to differentiate *single count noun* and *plural count noun*, in this case one could use a constraint such as [NUM *singular*] or [NUM *plural*].

The design of these hierarchies is an essential part of HPSG. Different hierarchies can achieve similar results, although they can differ both in performance (e.g., parsing speed), and in the 'elegance' of an analysis. Compare two example hierarchies provided in (12) and (13). Both these hierarchies can be used capture constraints about number and person in English, although they do it in very different ways.

The hierarchy in (12) treats *person* and *number* as parallel features. Using this hierarchy, a noun would need different constraints for person and for number, which is exemplified in (14). In this case, as a common plural noun, *students* has the constraints [PER *3*] and [NUM *plural*]. The hierarchy shown in (13),[3] on the other hand, is a bit more complex. Instead of two constraints, this hierarchy allows a grammar to define person and number using a single constraint, named 'PERNUM' in (15). In the hierarchy shown in (13) there are six leaf nodes (*1sg*, *2sg*, *3sg*, *1pl*, *2pl*, and *3pl*). Unsurprisingly, this is the same number as the product of three values for *person* and two values for *number* possible to derive from (12).

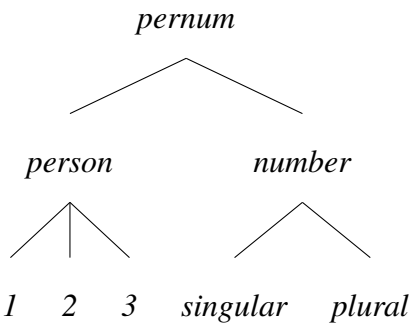
The main difference between both hierarchies is the fact that (13) is able to better capture a fact about the English, pertaining to subject-verb agreement. In Present Simple, English has only two inflected forms. For a verb like *to sleep*, these two forms would be *sleep* and *sleeps*. Out of the six possible values to derive from *person* and *number*, only the third-person singular takes the form *sleeps*, while the other five combinations take the form *sleep*.

Using the hierarchy shown in (13), one can see that this fact is encoded in the hierarchy – all five combinations of *person* and *number* that take the form form *sleeps* can be found under the type *non3sg*, and only the type *3sg* is not a sub-type of *non3sg*. Using this hierarchy, a verb with the form *sleep* could constraint its subject to be [PERNUM *non3sg*], which would license

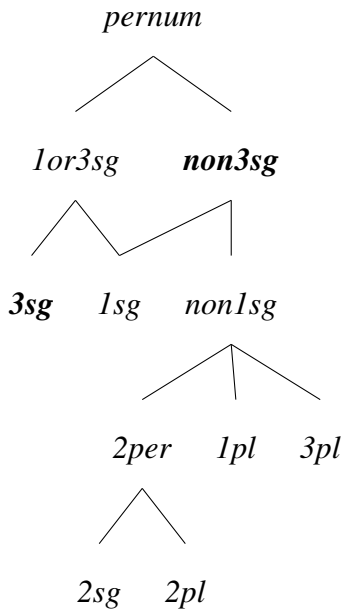
[3]This simplified hierarchy was extracted from Flickinger et al. (2000).

examples (16) to (21), but not (22) and (23) because the feature [PERNUM *3sg*] of the subject is incompatible with feature [PERNUM *non3sg*] of the verb.

(12)



(13)



(14)

$$\left\langle students, \begin{bmatrix} word & \\ \textsc{head} & noun \\ \textsc{arg.pernum} & \begin{bmatrix} \textsc{per} & 3 \\ \textsc{num} & plural \end{bmatrix} \\ \textsc{val.spr} & \left\langle \begin{bmatrix} \textsc{head} & determiner \\ \textsc{opt} & + \end{bmatrix} \right\rangle \end{bmatrix} \right\rangle$$

(15) $\left\langle students, \begin{bmatrix} word & \\ \textsc{head} & noun \\ \textsc{arg.pernum} & 3pl \\ \textsc{val.spr} & \left\langle \begin{bmatrix} \textsc{head} & determiner \\ \textsc{opt} & + \end{bmatrix} \right\rangle \end{bmatrix} \right\rangle$

(16) *I* *sleep* .
1SG sleep.*non3sg* .

(17) *You* *sleep* .
2SG sleep.*non3sg* .

(18) *We* *sleep* .
1PL sleep.*non3sg* .

(19) *You* *sleep* .
2PL sleep.*non3sg* .

(20) *They* *sleep* .
3PL sleep.*non3sg* .

(21) *Students* *sleep* .
student.3PL sleep.*non3sg* .

(22) * *He/She/It* *sleep* .
3SG sleep.*non3sg* .

(23) * *The* *student* *sleep* .
det student.3SG sleep.*non3sg* .

The example used above is but a small snippet of what would be expected from a large resource grammar. To provide a sense of scale, the ERG has a hierarchy with over 9,000 types.[4] In addition, these hierarchies are mostly language specific. Despite some expected overlap in fundamental aspects of the HPSG theory, hierarchies are mostly language dependent. This is fairly easy to understand since the example presented above is an idiosyncrasy of English – many

[4] `https://github.com/delph-in/docs/wiki/GrammarCatalogue`

other languages have unique inflections for each combination of person and number, and may even need extra features such as *gender*; other languages might not inflect verbs at all, making concepts like subject-verb agreement irrelevant.

Providing a full overview of HPSG is well beyond of the scope of this thesis, for a good general introduction see Sag et al. (1999). However, a basic understanding of the framework is essential to follow the discussion regarding the development of ZHONG (including the creation of new mal-rules) – which will be the focus of Chapter 6. I will therefore introduce core concepts (e.g., rule-names, feature-names, their usages, etc.) wherever relevant, and only with the necessary depth to follow the work being presented.

3.4 Minimal Recursion Semantics

All DELPH-IN grammars produce similar semantics – Minimal Recursion Semantics (MRS, Copestake et al., 2005). MRS is a computationally tractable semantic framework, capable of retaining and representing scope ambiguity in a text-based, non-recursive structure. It is also able to support a fluid interface between syntax and semantics, espousing the principles of semantic composition. MRS can be used for both parsing and generation, using the same grammar, which allows many interesting research opportunities in Natural Language Generation, including semantic-based machine translation (Bond et al., 2005, 2011).

Copestake et al. (2005) list four criteria that went into the design of MRS. These criteria were: expressive adequacy (i.e., the ability to adequately express meaning), grammatical compatibility (i.e., the ability to interface with syntax in a clean easy manner), computational tractability (i.e., the ability to efficiently check for partial/full equivalence of semantic representations), and underspecifiability (i.e., the ability to express semantic ambiguity when necessary, namely scope ambiguity).

At its core, MRS is a representational framework integrating a range of techniques and assumptions from other semantic theories or representations (Copestake et al., 2005), in a way that makes MRS especially compatible with feature-based grammars (e.g., HPSG, Lexical Function Grammar, etc.) – i.e., allowing semantic composition through the unification of type feature

structures. MRS uses many of the assumptions of predicate calculus with generalized quantifiers. MRS's primary units of meaning are known as *elementary predicates* (EP), which represent semantic relations with any number of associated arguments. EPs can have semantic features such as tense, mood, person or number associated with them. The scope ambiguity, for example in quantification, is defined by *qeq* (equality modulo quantifiers) relations.

Example (24) shows an example MRS for the sentence: *These students sleep.* The same MRS can be visualized as a feature structure in Figure 3.2 or as a dependency graph (also known as DMRS, Copestake, 2009) in Figure 3.3. In order to simplify the visualization, features on variables (such as the fact that `x3` is `pl`) are not shown in the feature structure or the graph although they are present in the formalism.

In (24), the MRS includes three EP relations for the determiner, noun and verb, respectively: `_these_q_dem`, `_student_n_of`, and `_sleep_v_1`. The `ARG0` (the identity-defining argument) for `_these_q_dem` and `_student_n_of` are co-indexed (i.e., `x3`), and the restriction `h5 qeq h7` shows that the quantifier `_these_q_dem` can only scope over `_student_n_of` through the use of labels (`LBL`). The relation `_student_n_of` presents an unfilled `ARG1` (i.e., `i8`). This argument would be filled if the object of study is included in a sentence (e.g., *The students of linguistics sleep.*). Since `i8` is not co-indexed with any other predicate, this shows that such semantic argument was not used in the source sentence. Finally, the verbal predicate `_sleep_v_1` indexes its `AGR1` (i.e., the entity that sleeps) to `x3` – *these students* –, effectively linking all predicates in the sentence.

In the context of this thesis, the reader needs only to understand the basic framework of MRS. A full account of the algebra and notation of the various formats of MRS can be found in Copestake et al. (2005) and Copestake (2009).

(24) These students sleep.

```
[ LTOP: h0 INDEX: e2 [ e SF: prop TENSE: pres MOOD: indicative
                       PROG: - PERF: - ]
  RELS: < [ _these_q_dem<0:5>
            LBL: h4
            ARG0: x3 [ x PERS: 3 NUM: pl IND: + ]
            RSTR: h5
            BODY: h6 ]
          [ _student_n_of<6:14>
            LBL: h7
            ARG0: x3
            ARG1: i8 ]
          [ _sleep_v_1<15:21>
            LBL: h1
            ARG0: e2
            ARG1: x3 ] >
  HCONS: < h0 qeq h1
           h5 qeq h7 >
  ICONS: < > ]
```

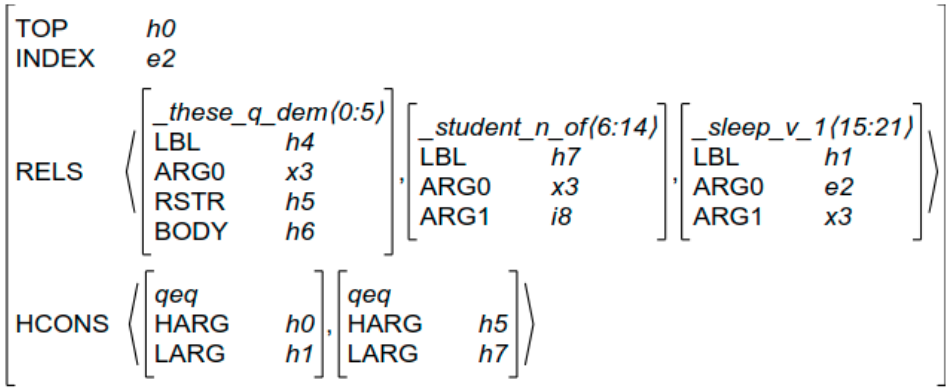


Figure 3.2: MRS for *These students sleep.*

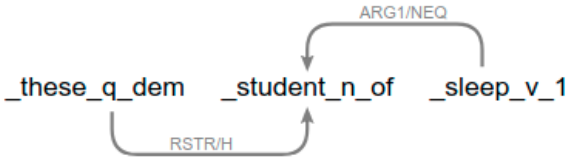


Figure 3.3: DMRS for *These students sleep.*

In Section 3.5, below, the relation between semantics and mal-rules will be discussed in greater detail. However, since this discussion would be valid for any sufficiently-expressive computational semantics framework, MRS will not be on the foreground of this discussion. The goal of this section was to provide the reader with a notion of how semantics are encoded in DELPH-IN grammars, and to assure them that DELPH-IN grammars use a very robust semantic framework, with sufficient expressive adequacy. But from this point on, semantics will be discussed more abstractly, allowing syntax to be the in the foreground of the discussions – especially in relation to mal-rules.

3.5 Robust Parsing and Mal-Rules in HPSG

Robustness is the inherent ability of a system to continue performing (well) even in contexts or situations that challenge its original design. In the specific case of NLP systems, the concept of robustness becomes linked to the idea of a system being able to parse *any* linguistic input – regardless of its well-formedness, the use of new or out-of-vocabulary words, etc.. Zhang (2008) provides a fuller definition and discussion of different dimensions of robustness in NLP, especially in the context of deep linguistic processing and HPSG.

In constraint-based linguistic language models, such as HPSG, robustness is an early and ever present concern. When compared with shallow parsing methods, the explicit nature of constraint-based linguistic language models tends to make these models much less robust. In other words, forms of input that were not explicitly accounted for in grammar are simply rejected. This is not necessarily a bad thing, since constraint-based models, such as HPSG, are theorized to make an implicit grammaticality judgment when they parse or reject an input – which is

usually not not true for statistical-based parsers.

And so, this rigidity (i.e., the lack of inherit robustness for ill-formed or unknown input) that may be considered a problem for some NLP applications, becomes an invaluable tool to deal with problems concerning grammatically.

Mal-rules, introduced above in Section 2.1, can be seen as a way to increase robustness in constraint-based grammars. In HPSG, mal-rules can be seen as drawing inspiration from *constraint relaxation* or *partial constraint satisfaction* – an early idea present throughout problem solving in AI (Guesgen and Hertzberg, 1992). But instead of relaxing existing constraints, mal-rules effectively perform targeted constraint relaxation by adding new rules that are less constrained than what would be expected in a prescriptive grammar – i.e., they can parse ungrammatical input which should, in principle, be rejected by the grammar.

In order to keep the grammars true to their nature – and able distinguish grammatical language from ungrammatical language – it is important to draw clear distinctions between normal rules and mal-rules. This is usually done through naming conventions within a grammar (e.g., using the prefix 'mal-'). By using traceable features such as naming conventions, checking the nodes of a parse tree can easily identify if a sentence is grammatical or not.

When used by error detection or correction systems, the full rule name or lexical entry can then be used to identify the specific kind of error and hence allow a system to say, for example, "there is something wrong with the subject-verb agreement in this sentence".

Within implemented grammars, mal-rules can be selectively available for parsing but not for generation (Bender et al., 2004), or to allow certain types of errors but not others. For grammars that produce a semantic representation, as is the case in this thesis, mal-rules can be designed to reconstruct the semantics of ungrammatical sentences in a way that allows the generation of corrected counterparts (Bender et al., 2004). And, in some cases, a single ungrammatical sentence can trigger multiple parses using mal-rules, each reconstructing different semantics that define different possible intended meanings of that specific ungrammatical input.

Previous works exploring mal-rules in HPSG include English (Bender et al., 2004; Flickinger, 2010; Flickinger and Yu, 2013), Norwegian (Hellan et al., 2013), German (Heift, 1998), Spanish (Costa et al., 2006) and French (Hagen, 1994). From these, only English and Norwegian are

known to still be in active development.

According to Bender et al. (2004), the implementation of mal-rules in HPSG grammars can be done through three major classes of linguistic objects: syntactic rules, lexical rules, and lexical items. But even though each method has some degree of specificity, making them useful in detecting different kinds of errors, there is also some overlap in their explanatory power (i.e., similar errors can be captured in more than one way).

Each grammar has a single hierarchy of types from which all grammar rules and lexical types are inherited. Mal-rules usually branch off a generic rule or lexical type from which both mal-types and normal types are formed. The place in the hierarchy from where these rules branch off is dictated by which features the mal-rule needs to differ from the prescriptive grammar. However, following common practice, only the most specific types are usually instantiated in the grammar – i.e. only the most specific types are available for the grammar to see/use. For mal-rules, it is especially important that only the most specific types be instantiated since this provides a better control over which errors they can detect.

In the remainder of this section I will explore these different levels of specificity, as well as how multiple mal-rules can be used together to reconstruct multiple plausible meanings for a single ungrammatical sentence.

Syntactic Mal-Rules in HPSG

The use of syntactic mal-rules in HPSG is both powerful and flexible. Consider the ungrammatical noun phrase (NP) *this students*. Under normal circumstances, this phrase is not grammatical. In HPSG, this is ensured by the Specifier Head Agreement Constraint (SHAC) present in the Head-Specifier Rule (25), as proposed in Sag et al. (1999). According to the SHAC, phrases taking a specifier are required to unify their agreement features with those of their specifier – this is shown by $\boxed{2}$ in (25). The specifier of a NP is its determiner, so this is what establishes the required agreement between the noun and the determiner.

(25) *Head-Specifier Rule*

$$\begin{bmatrix} \textit{phrase} \\ \text{SYN} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{SPR} \ \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \rightarrow \boxed{1} \quad \mathbf{H} \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{AGR} \ \boxed{2} \end{bmatrix} \\ \text{VAL} \begin{bmatrix} \text{SPR} \ \left\langle \boxed{1} \begin{bmatrix} \text{AGR} \ \boxed{2} \end{bmatrix} \right\rangle \\ \text{COMPS} \ \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(26) *Mal-Head-Specifier Rule*

$$\begin{bmatrix} \textit{phrase} \\ \text{SYN} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{SPR} \ \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \longrightarrow \boxed{1} \quad \mathbf{H} \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{AGR} \ \boldsymbol{X} \end{bmatrix} \\ \text{VAL} \begin{bmatrix} \text{SPR} \ \left\langle \boxed{1} \begin{bmatrix} \text{AGR} \ \boldsymbol{Y} \end{bmatrix} \right\rangle \\ \text{COMPS} \ \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

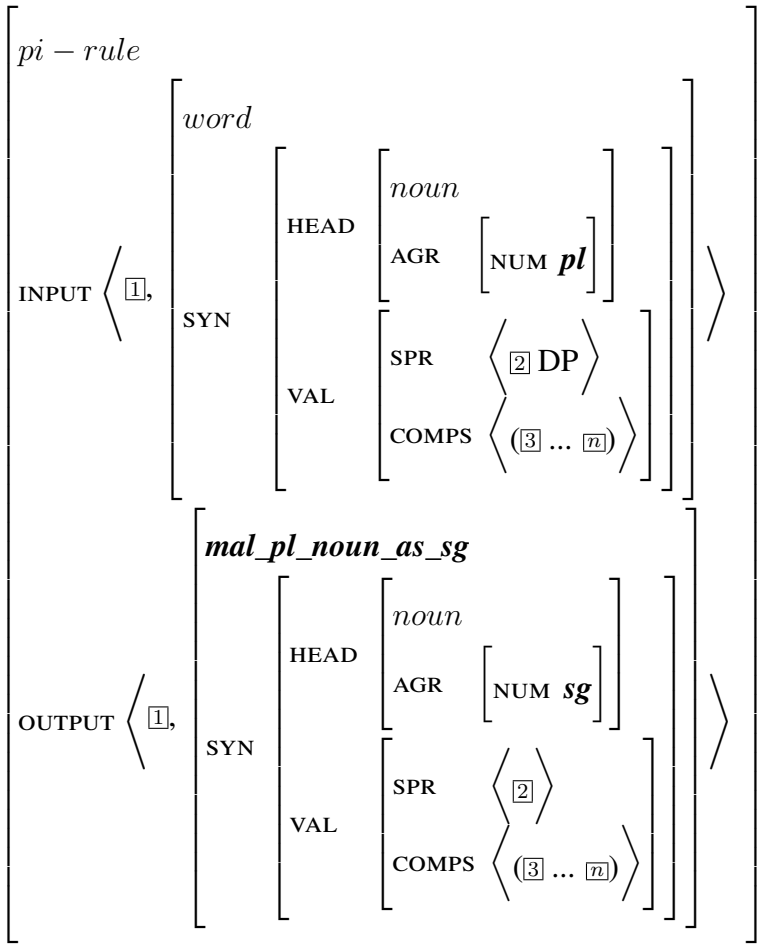
One possible way to build the NP *this students* would be to relax the constraint imposed by the SHAC. Creating a new rule where this constraint is not enforced would qualify it as a mal-rule – since such rule would allow ungrammatical phrases to be licensed by the grammar. This mal-rule can be found in (26). Note that where $\boxed{2}$ in (25) made sure both the head-daughter (i.e., noun) and its specifier (i.e., determiner) agreed, in (26) this is not true. (26) would allow the grammar to build *this students* as a valid NP.

The Head-Specifier Rule, as described in Sag et al. (1999), is used to build many kinds of phrases, including full sentences (e.g., linking NP subjects and their VP predicates). This means that the mal-rule shown in (26) would also license sentences such as '*Students sleeps.*' or '*I sleeps.*' – where the subject does not agree with the main verb. This accounts for the flexible power of syntactic mal-rules, but also shows that even though (26) could be used to detect ungrammatical sentences, it has a fairly low precision with regard to what kind of error it licenses – i.e., an unspecified problem in agreement.

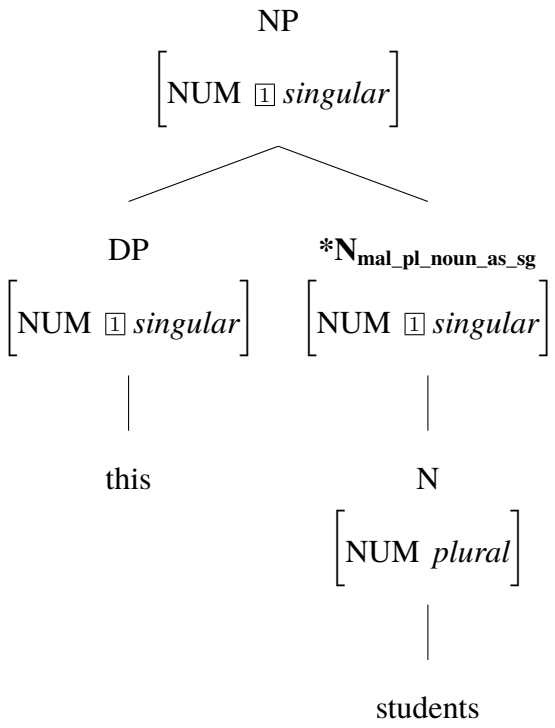
Lexical Mal-Rules in HPSG

HPSG grammars often have a rich hierarchy of lexical rules. An alternative way to build the NP *this students* would be through lexical mal-rules. This could be done with a lexical rule that allows, for example, a plural noun to be used as a singular noun. An example of this rule is shown in (27). This lexical mal-rule can only be applied to plural nouns, and produces a copy of the input noun, changing only the number feature (i.e., from *plural* to *singular*). The output of this rule takes a special type that is traceable in the syntactic tree (i.e., in this case, $mal\_pl\_noun\_as\_sg$). Using the lexical mal-rule shown in (27), an English grammar would be able to build the NP *this students* by first changing the number feature of the word *students* to singular, and then using the normal rule that joins nouns and determiners – as is shown in (28). Note that since the number feature for the noun was changed, the original Head-Specifier Rule shown in (25) can be used to create this NP. The SHAC is still satisfied, as can be seen by the co-indexation of the number feature through $\boxed{1}$.

(27)


pi − rule
INPUT
1
word
SYN
HEAD
noun
AGR
NUM pl
VAL
SPR
2 DP
COMPS
(3 ... n)
OUTPUT
1
mal_pl_noun_as_sg
SYN
HEAD
noun
AGR
NUM sg
VAL
SPR
2
COMPS
(3 ... n)

(28)



Mal Lexical Entries in HPSG

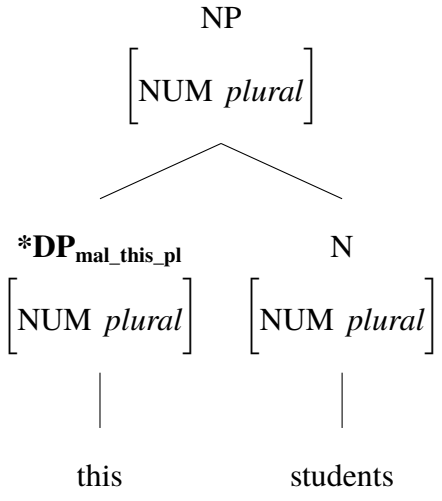
Finally, a third way to build the NP *this students* is to use a mal lexical entry. This method is similar, in spirit, to lexical mal-rules, but instead of generalizing across word classes, it provides an alternative mal lexical entry for specific words that are known to be source of errors. One such example would be the correct and mal lexical entries for *this*, shown as (29) and (30), respectively.

Entries (29) and (30) differ only slightly. The first of these differences is the value for the number feature. For the mal lexical entry, shown in (30), it is set to *plural*. Additionally, the semantic relation it introduces is similar to what would be expected of an entry for the determiner *these*. In short, (30) behaves like the word *these* but carries the form *this*. This mal lexical entry would allow a grammar to licence the NP *this students* following the tree shown in (31).

(29) $\left\langle this, \left[\begin{array}{l} \textit{determiner} \\ \textsc{syn} \left[\begin{array}{l} \textsc{head} \left[\begin{array}{l} \textsc{det} \\ \textsc{agr}|\textsc{num} \ \textbf{\textit{singular}} \end{array}\right] \end{array}\right] \\ \textsc{sem} \left[\begin{array}{l} \textsc{index} \ i \\ \textsc{restr} \left\langle \left[\begin{array}{ll} \textsc{reln} & \textbf{\textit{this}} \\ \textsc{bv} & i \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$

(30) $\left\langle this, \left[\begin{array}{l} \textit{mal\_this\_pl} \\ \textsc{syn} \left[\begin{array}{l} \textsc{head} \left[\begin{array}{l} \textsc{det} \\ \textsc{agr}|\textsc{num} \ \textbf{\textit{plural}} \end{array}\right] \end{array}\right] \\ \textsc{sem} \left[\begin{array}{l} \textsc{index} \ i \\ \textsc{restr} \left\langle \left[\begin{array}{ll} \textsc{reln} & \textbf{\textit{these}} \\ \textsc{bv} & i \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$

(31)



Combining Approaches

Although they might seem to provide very similar results, it should be noted that the trees shown in (28) and (31) differ in one key aspect – the value for the syntactic number feature of the produced NP. In HPSG, similar to other grammar theories, the syntactic number of a phrase is determined by the head of that phrase – in a NP, this would be the noun. This is a good example of how mal-rules can be used to reconstruct different possible meanings from a single ungrammatical input.

To be able to evaluate the full reach of meaning reconstruction, consider (32) – a variation of the mal-rule introduced in (26). This rule is a variation of the Head-Specifier, but it is more restrictive because it requires its head-daughter to be headed by a VERB (i.e., a verb phrase). This effectively changes the general Head-Specifier rule into a Head-Subject rule (since the specifier of a verb-phrase is its subject). Note, however, that similar to what was discussed above for (26), in (32) the agreement is also not enforced. In short, (32) allows sentences where the subject and the main verb of a sentence do not agree.

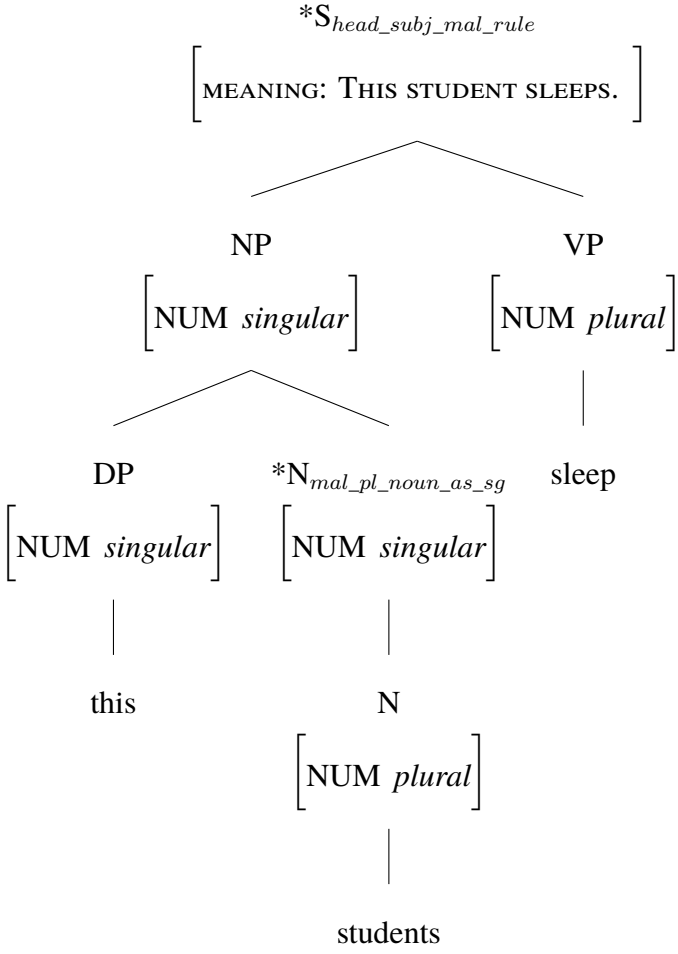
Using only the three mal-rules shown in (27), (30) and (32), the grammar can provide two reconstructions for the ungrammatical sentence: * *This students sleep*. These reconstructions are shown in (33) and (34). The main difference between these two trees is the reconstructed meaning. In (33), the grammar reconstructed a sentence where only a single student sleeps. And in (34), the reconstructed meaning assumes more than one student sleep.

For systems where the goal is simply grammatical error detection (i.e., without correction), traversing the parsing tree and looking for nodes where mal-rules were used is enough to diagnose the ways in which a sentence is ungrammatical. However, if a grammar has generation capabilities, reconstructing different meanings also allows the generation of the corrected counterparts. Using mal-rules, most implemented HPSG grammars within DELPH-IN can be used to produce fully capable error detection and correction systems.

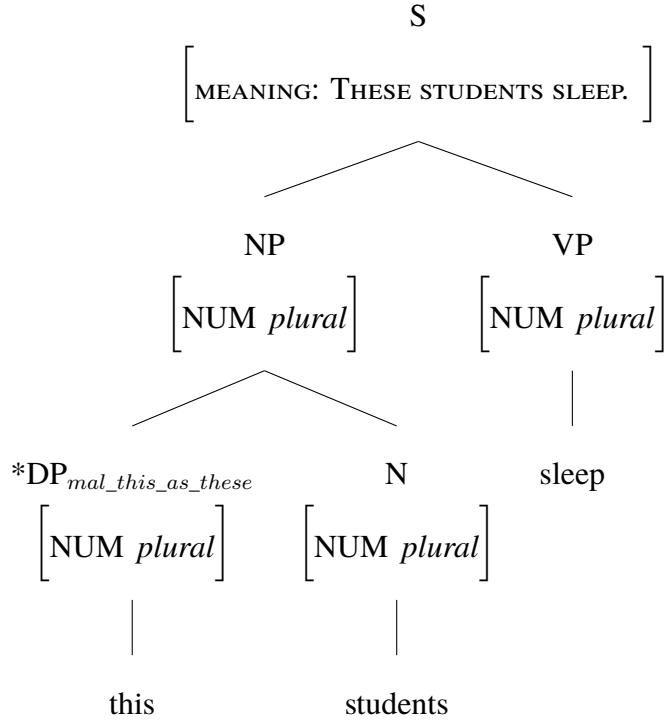
(32) *Head-Subject Mal-Rule*

$$\begin{bmatrix} \textit{phrase} \\ \text{SYN} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{SPR} \ \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \longrightarrow \boxed{1} \quad \mathbf{H} \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textbf{VERB} \\ \text{AGR} \ \boldsymbol{X} \end{bmatrix} \\ \text{VAL} \begin{bmatrix} \text{SPR} \ \left\langle \boxed{1} \begin{bmatrix} \text{AGR} \ \boldsymbol{Y} \end{bmatrix} \right\rangle \\ \text{COMPS} \ \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(33)



(34)



3.6 Summary

In this chapter I have introduced the DELPH-IN consortium, while also motivating some of the efforts that are being pursued to push deep linguistic processing into the foreground of NLP. Deep linguistic processing offers a theoretically motivated way to process natural language, providing deep syntactic and semantic structures that can be exploited in a variety of NLP tasks.

I have introduced the methodology used in grammar engineering – an intrinsically data-driven, interdisciplinary methodology drawing from software engineering and theoretical linguistics. And I have provided brief introductions to HPSG and MRS. HPSG is an constraint-based linguistic theory that relies on unification of feature structures to parse natural language. MRS is a state-of-the-art computational semantics framework, that can be flawlessly integrated in the unification process of HPSG grammars. Together, HPSG and MRS create an excellent platform to explore deep linguistic processing.

Finally, I concluded this chapter with a discussion of how mal-rules can be implemented within HPSG. Mal-rules can be easily implemented in HPSG because they rely on the important fact that HPSG tightly controls what can be parsed by a grammar. In normal grammars, there

is an implicit (positive) grammaticality judgment for every sentence that receives a parse, while ungrammatical sentences ought to be rejected. Mal-rules flip the issue of *lack of robustness*, often associated with constraint-based parses, into an unique opportunity to transform these parsers into error detection and correction systems – capable of performing error detection and correction with enormous amounts of precision, and linked to linguistically motivated features that can be used to provide corrective feedback in educational applications.

Chapter 4

Linguistic Tools and Resources

This chapter introduces some of the most important tools and language resources used throughout this thesis. It starts by providing some background on the English Resource Grammar and on ZHONG, a Mandarin Chinese grammar based on DELPH-IN's LinGO Grammar Matrix system. It then introduces multiple tools, made available through DELPH-IN, that were essential to the execution of my work. It concludes with a short introduction of the Princeton WordNet, a lexico-semantic resource used in CALLIG, presented in Section 8.2.

4.1 The English Resource Grammar

The English Resource Grammar (ERG: Flickinger, 2000; Flickinger et al., 2000; Copestake and Flickinger, 2000) is an open-source, broad-coverage computational grammar for English. Its roots trace back to 1991 (Flickinger, 2011), and it has remained in development ever since. Dr. Daniel Paul Flickinger is the maintainer and main developer of this grammar, at CSLI, Stanford.

The ERG is a symbolic parser with a very large lexicon and wide coverage of syntactic phenomena capable of producing high-precision syntactic and semantic representations for English. It follows the theoretical framework of HPSG and produces MRS semantics.

The ERG is one of the backbones of DELPH-IN. As its largest implemented grammar, it drives much of DELPH-IN's research agenda while, at the same time, it also functions as a gold-standard for grammar engineering – setting the example of what a grammar can achieve when provided with an abundant level of dedication and linguistic rigor. In this role, the ERG

is also a 'grammar engineering resource', providing inspiration to many linguistic analyses in a variety of other languages.

Despite a high standard of linguistic accuracy, the ERG has an impressive coverage over unseen English text. Flickinger (2011) reports observed coverage between 81.2% and 96.8% across a variety of genres, including essays, Wikipedia, technical manuals and online user forums. These numbers report the state of the ERG roughly 10 years ago, and it has been in constant development since then.

Of special interest for this thesis is the fact that this grammar has had substantial work on the design of mal-rules (Bender et al., 2004; Flickinger and Yu, 2013; Suppes et al., 2014; Flickinger et al., 2016). This work is released as a parallel version of the grammar, and it allows the ERG to identify and diagnose a variety of ungrammatical and stylistically deprecated sentences. Much of the early work in mal-rules for the ERG was dedicated to the *language arts* program of the Stanford's Education Program for Gifted Youth (EPGY), targeting elementary-school and middle-school English language education in the USA. Through a very large, multi-year study involving thousands of students in this educational program, Suppes et al. (2014) report that the use of corrective feedback produced on top of the ERG's ability to determine grammatical correctness of written English showed very positive results in students' ability to improve their language use.

Beyond the EPGY program, the mal-rule enhanced version of the ERG has also shown promising results in two different shared tasks.

In 2013, a system based on the ERG participated in the CoNLL Shared Task on Grammatical Error Correction. To the best of my knowledge, this was the first open task where the ERG participated as an error detection system. The task consisted of identifying and classifying errors in five broad categories. Despite fairly low official scores (Precision=0.2093, Recall=0.0586 and F1=0.0981), the ERG system was able to identify multiple problems concerning the 'gold' annotations provided as development and evaluation data in this shared task. This included missing error annotations in the gold data, wrong/unnecessary error annotations in the gold data, and the inability to provide other plausible corrections to annotated errors.

In 2016, a similar system (Flickinger et al., 2016) entered the Shared Task on Automated

Evaluation of Scientific Writing (Daudaravicius et al., 2016a). This shared task consisted of classifying sentences as 'needing revision' or 'not needing revision', and it had both a Boolean track and a probabilistic track. Out of nine systems that entered this shared task, the ERG-based system ranked second in the probabilistic estimation track (F1=0.7849), and fourth in the Boolean decision track (F1=0.5507). The ERG based system not only performed among the top for both tasks but was also, once again, capable of identifying multiple problems pertaining to inconsistencies in the gold data provided to develop and evaluate the systems in this task.

In this thesis, I use the ERG to build and evaluate an end-to-end online writing support system designed to help students with English scientific writing (presented in Chapter 8 and evaluated in Chapter 9). In the process, I improve the ERG's ability to classify and diagnose errors by building a new English treebank and a new parse-ranking model exploiting mal-rules (presented in Chapter 7 and evaluated in Chapter 9).

4.2 ZHONG: a Chinese HPSG Shared-Grammar

ZHONG (Fan et al., 2015a,b; Fan, 2019) is an open-source HPSG computational grammar based on the LinGO Grammar Matrix system (Bender et al., 2002, 2010), producing MRS semantics. ZHONG was born through the adaptation of ManGO: Mandarin Grammar Online (Flickinger and Yang, 2011), first started in 2011. This adaptation was done by Fan Zhenzhen (Fan, 2019), as the focus of her PhD studies at NTU. During this cycle of development, ZHONG was expanded from a 'toy grammar' (with an extremely limited vocabulary) to a 'resource grammar' – a computational resource aiming to cover naturally occurring language, so it could be used in a variety of Natural Language Processing tasks.

ZHONG was the first wide-coverage precision parser for Mandarin Chinese, leveraging and learning from previous theoretical work. ZHONG was also designed to allow the parallel development of a variety of Chinese languages/dialects and orthographies – sharing core elements of the syntax whenever possible and resorting to divergence in phenomena and orthography only when necessary – an idea that is inherently present in HPSG theoretical framework, and made explicit by the LinGO Grammar Matrix system. However, despite early efforts to accom-

modate this parallel development of multiple Chinese languages/dialects, this remains largely under-explored.

ZHONG's lexicon was built semi-automatically, using a free sample of the Sinica Treebank Corpus (Chen et al., 1996, 2003) made available through the Python Natural Language Toolkit (NLTK, Bird, 2006), and the Bilingual Chinese-English Wordnet (BOW, Xu et al., 2008).

The tagset used by the Sinica Treebank was mapped (by hand) to equivalent ZHONG lexical types – importing relevant lexical entries whenever equivalent types were available. The same principle was applied to BOW, mapping wordnet parts-of-speech to ZHONG's lexical types.

Basic verb-frames provided by the English Princeton WordNet (Fellbaum, 1998) were used to provide fine-grained mapping of verbal types. However, due to the coarse nature of verb-frames provided by the Princeton WordNet, the process was noisy – introducing many erroneous lexical entries that were left to be checked in the future (Fan, 2019).

Concerning the lexical acquisition done through the Sinica Treebank, even though this corpus had a detailed tagset with 178 syntactic categories (Chang and Chen, 1995), this semi-automatic process of lexical acquisition was still prone to introduce errors.

This automatic lexical acquisition was also done early in ZHONG's development process. In her work, Fan (2019) reports that since ZHONG did not have an equivalent lexical type for verbs taking sentential complements, a variety of tags in the Sinica Treebank that described this kind of verbs were mapped to regular transitive verbs instead. However, multiple lexical types describing verbs with sentential complements were introduced later in the development process, but the lexicon was not re-acquired.

Given the understanding that most of these automatically collected lexical entries could introduce errors or spurious ambiguity, all automatically collected lexical entries were kept separately from ZHONG's main lexicon – noting that they had never been hand-checked. A separate lexicon containing only hand-checked entries also exists, and it contains mostly functional words and basic vocabulary used for testing linguistic analyses.

The development of linguistic constructions within ZHONG followed a mix of data-driven and phenomena based methods, seeking wide coverage in available corpora while also taking a more linguistic perspective by developing analyses for a variety of phenomena involving the

function word 的 (*de*) – which is an ubiquitous word in Mandarin syntax. According to Fan (2019), more than 63% of sentences in the Penn Chinese Treebank (Xue et al., 2010) contained at least one instance of 的 (*de*) (≈1.98 occurrences/sentence). This particle is used, for example, in nominal and verbal modification, in nominalization, and in a variety of focus and partitive constructions. In addition to constructions using 的 (*de*), Fan (2019)'s work also focused on other phenomena, including reduplication and interrogatives.

ZHONG's version produced by Fan's PhD (v1.0) was evaluated for coverage using the Chinese portion of TUFS Open Language Resources (Kawaguchi, 2007) from Tokyo University of Foreign Studies). The Chinese portion of this corpus contains 1,523 sentences. Fan (2019) reports that ZHONG parsed 42.7% (n=651) of this (unseen) corpus. However, these numbers differ from the numbers I was able to produce using the officially released version of Fan's PhD on Github[1] – where it was verified that it was actually capable of parsing 60.6% of this corpus. This could, however, be explained by the fact that the officially released version was not exactly the version used to run Fan's evaluations, and that the released version contained other improvements that came from further revisions of the grammar.

In addition to this coverage evaluation, Fan (2019) also reports that ZHONG was able to generate at least one parse for 93.1% (n=743) of the 798 sentences contained in an internal development corpus known as CMNEDU – which will be revisited in Chapter 5. Of the 743 sentences with at least one parse, 92.7% (n=689) had at least one parse that was deemed suitable during a treebanking exercise. This also differs slightly from what could be replicated using the official release hosted on Github. Using this version I was able to parse only 91.9% of sentences belonging to the CMNEDU development corpus. This difference is fairly small, but shows a different trend than the coverage evaluation discussed in the paragraph above. This seems to corroborate the idea that the version released on Github was not exactly the same version used in the reported evaluations. In Chapter 9, where a series of evaluation experiments will be described, I will use the numbers I was able to produce instead of those reported by Fan (2019).

In this thesis, I further ZHONG's development – both adding new analyses for missing phenomena, and improving on existing analyses. These extensions are described in detail in

[1] `https://github.com/delph-in/zhong/releases/tag/v1.0-ZZ_PhD`

Chapter 6. The main drive of this development was the transformation of ZHONG into an error detection system through the implementation of mal-rules (this will also be presented in Chapter 6 and evaluated in Chapter 9). In the process, I also built a new Mandarin Chinese treebank and a new parse-ranking model for ZHONG using mal-rules (this will be presented in Chapter 7 and evaluated in Chapter 9).

4.3 DELPH-IN Tools

The work developed during this thesis also made extensive use of a variety of tools provided by DELPH-IN (introduced above, in Section 3.1). In particular, the work presented in this thesis relied heavily on parsing, documentation and treebanking tools made available by multiple members of this this consortium.

ACE Tools

ACE Tools[2] are a suite of open-source applications based on the Answer Constraint Engine[3] (ACE). ACE is a highly efficient HPSG unification engine for DELPH-IN grammars that supports both parsing and generation for grammars written in Type Description Language (TDL, Krieger and Schäfer, 1994). ACE was developed and is maintained by Woodley Packard.

In addition to the main parsing engine, this thesis also uses two other systems available in ACE Tools: the Full Forest Treebanker and ready-to-use binaries to train parse-ranking models from full-forest treebanks.

The Full Forest Treebanker (FFTB, Packard, 2015) is a open-source treebanking workbench designed specifically to work with HPSG constraint-based grammars producing MRS semantics.

Treebanking is an integral part of grammar engineering methodology (see Section 3.2 for more details), and it allows human annotators (often grammarians or trained annotators) to hand-pick the best analysis for a given sentence, often from a very large pool of available analyses (where the size of the pool reflects ambiguity in the different analyses produced by a com-

[2] `http://sweaglesw.org/linguistics/acetools/`
[3] `https://github.com/delph-in/docs/wiki/AceTop`

putational grammar). More concretely, treebanking workbenches are used by grammarians to evaluate their computational grammars by providing a way to evaluate computational implementations at scale, and to produce high quality datasets using a computational grammar – useful for many NLP tasks, such as training parse ranking models.

While the FFTB is not the only treebanking workbench within DELPH-IN (see, for example [incr tsdb()], Oepen, 2001) it is currently the most widely accepted system for newer grammars, mainly due to its exemplary use of data structures and memory. The FFTB allows grammarians to select any analysis among all possible analyses a grammar offers for a given sentence (which explains the origin of *full* in *full-forest*). The other available tool within DELPH-IN, the [incr tsdb()], while still considered the canonical system, is much older and its architecture is very taxing on computational resources (e.g., memory). As such, treebanking with the [incr tsdb()] often means that the grammarian needs to limit the maximum number of parses made available for each sentence before being able to select the best analysis (i.e., only a *partial forest* of all analyses is made available to the annotators). As such, while the [incr tsdb()] still has many features that are not available in the FFTB, such as different methods of annotation and better test suite management tools, the work presented in this thesis made the choice to use FFTB. Some of the valuable features made available in [incr tsdb()] and missing from FFTB, namely many test suite management and inspection tools, were implemented in a new web-system that falls outside the scope of this thesis.

This thesis made extensive use of ACE as main parser in the development and evaluation process of two main components: the development of new and improved analyses for ZHONG (see Chapter 6), and the development of a new online writing support system designed to help students with English Scientific Writing (see Chapter 8). The FFTB tools were also used to build and evaluate two different mal-rule enhanced treebanks and parse-ranking models, one for English and one for Mandarin Chinese (see Chapter 7).

Linguistic Knowledge Builder

The Linguistic Knowledge Builder (LKB, Copestake, 2002) is one of the main parsers and grammar development environments provided by DELPH-IN.

It is another unification engine for HPSG based grammars written in HPSG using Type Description Language (TDL, Krieger and Schäfer, 1994). It is a less efficient parser than ACE, but it includes features ACE does not – such as inspecting the entire type hierarchy of a grammar. In addition, the LKB is necessary to work with other systems within DELPH-IN.

Even though grammar development and parsing was done using ACE, I have made sure ZHONG always remained compatible with the LKB and, therefore, with other DELPH-IN systems that depend on the LKB to interface with grammars. The most notable example of this was the Linguistic Type Database, which will be introduced below.

In this thesis, I used the LKB-FOS[4], a more recent implementation of the legacy LKB system using fully open-source compilers – making this version truly open-source.

The Linguistic Type Database

The Linguistic Type Database (LTDB, originally named *Lextype DB*, Hashimoto et al., 2007, 2008) is an open-source web application capable of extracting information from DELPH-IN treebanks and grammars and present it in a structured/searchable way. In particular, the LTDB stores and displays useful information concerning the syntactic behavior of both lexical entries and larger structures – making it a sort of 'grammar-wide' documentation tool.

The LTDB is a valuable tool for both grammarians and treebankers. It can be used to remind grammarians or teach treebankers the inner working of particular phenomena in a grammar while, at the same time, it can also be used to help harmonize the treebanking process by providing an easy access to previously tagged data. This data can be used to see how similar linguistic structures have been treebanked before, so new treebanks can follow similar analyses.

The LTDB was originally developed with a focus on lexical information for Japanese, using the Hinoki Treebank (Bond et al., 2004, 2008) and its source grammar, JACY (Siegel and Bender, 2002; Siegel, 2006; Siegel et al., 2016). With time, the LTDB developed into a more holistic and grammar-agnostic system. The LTDB is now able to provide information about all types and rules contained in a grammar while also providing examples of how they have been used in the past (if a treebank is available) – making it an essential tool to build treebanks.

[4] `http://moin.delph-in.net/wiki/LkbFos`

In this thesis, I have not only helped improve the LTDB's user interface (integrating the Delphin-Viz[5] library into the system and adding visual syntactic and semantic representations for treebanked sentences), but I have also made extensive use of this system as a documentation tool to build both the English and Mandarin Chinese Treebanks (see Chapter 7).

PyDelphin

PyDelphin (Goodman, 2019) is an open-source library for Python, providing multiple APIs to use, read and manipulate DELPH-IN software, data and grammars. Because it is written in Python, PyDelphin is a great tool to bridge DELPH-IN tools (such as ACE) with web-applications. This was its main usage in this thesis.

PyDelphin is an essential building block of two applications built as part of this thesis – the LCC-APP and CALLIG, both of which will be fully discussed in Chapter 8. These applications use PyDelphin both as a wrapper for the ACE parser, but also to examine the parsing results and to identify if mal-rules exist in the parsing output.

Finally, PyDelphin also integrates well with another important DELPH-IN library: Delphin-Viz. This library is capable of producing HTML renderings of DELPH-IN syntactic and semantic results, and it is the same library used to improve the LTDB, introduced above. These two libraries have also been used extensively to build a set of grammar engineering development, debug and evaluation tools used throughout this thesis. These set of tools will not, however, be presented in this thesis.

4.4 Princeton WordNet

The Princeton WordNet (PWN, Miller, 1995; Fellbaum, 1998) is an open lexico-semantic resource for the English language. It organizes information through semantic relations (such as synonymy, hyponymy, meronymy, etc.), encoding encyclopedic knowledge about how concepts in English relate with each other.

Despite being useful for a variety of tasks in NLP, in this thesis I use the PWN simply as

[5] `https://github.com/delph-in/delphin-viz`

a repository of English concepts. This is used in a variety of language games belonging to CALLIG (introduced in Chapter 8) to select random words.

Towards the end of this thesis, however, I will try to elaborate how wordnets in general can be useful resources to use in tandem with computational grammars to better diagnose certain classes of errors where word senses become relevant.

4.5 Summary

In this chapter I have presented a variety of tools and resources essential to the execution of this thesis. I have introduced both the English Resource Grammar and ZHONG, two HPSG implemented grammars capable of producing MRS semantics. The ERG is a state-of-the-art grammar, the largest grammar within DELPH-IN and a mature error detection system through years of development of mal-rules. ZHONG, on the other-hand, is a fairly young grammar, but trying to follow in the footsteps of the ERG.

I have also presented multiple tools available within the DELPH-IN ecosystem, which were also essential to complete this thesis. These include two unification engines (ACE and LKB), a treebanking tool capable of training parse-ranking models (FFTB), a documentation tool (LTDB) and a Python library capable of interacting with many of DELPH-IN systems and data (PyDelphin).

I closed the chapter by introducing the Princeton WordNet, a lexical resource for English currently used in some of the applications produced by this thesis, but which will also be discussed as a focus for future work.

Part II:

Implementation

Chapter 5

Educational and Learner Corpora

This chapter describes the preparation and compilation of three corpora – including two learner corpora (one for English and one for Mandarin Chinese), as well as the Mandarin Education Corpus (MEC), which was used to collect information about the lexicon and the syntactic structures used in early Mandarin Chinese Education. The information contained in these corpora guided much of the remainder of the work presented in this thesis (i.e., the development and exploitation of the error detection technology).

5.1 Expanding IMI: a Learner Corpora Tagging Tool

In order to build learner corpora for both English and Mandarin Chinese, I have developed a new web-based learner corpus tagger. This tool was developed as an expansion of IMI – a multilingual semantic annotation environment (Bond et al., 2015), of which I was one of the main developers. IMI is a semantic tool for multilingual corpora that uses the Open Multilingual Wordnet (OMW, Bond and Foster, 2013) to enrich a corpus with multiple layers of morphosyntactic and semantic information, including sense tagging. This system was initially built to support the annotation of the NTU Multilingual Corpus (NTUMC, Tan and Bond, 2014) and, because of this, it has been widely tested with Chinese, English, Japanese and Indonesian languages.

In its original form, IMI provides multiple layers of annotation that include lemmatization, POS tagging, sense tagging, sentiment annotation and interlingual-mapping, just to mention a

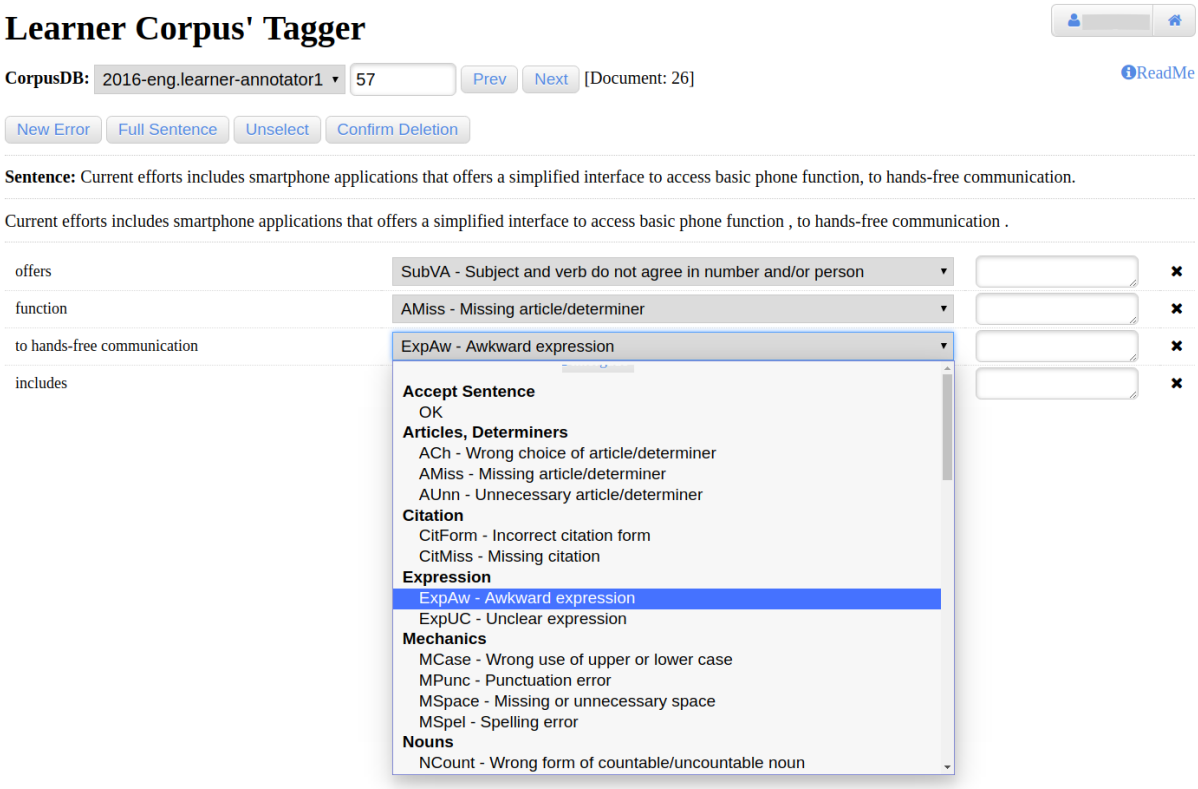


Figure 5.1: Annotation tool developed for the corpus annotation, as an extension of IMI

few. It is developed in Python and SQLite, and supports both concurrent annotation (i.e multiple taggers tagging the same data at the same time), as well as parallel tagging (i.e., multiple taggers tagging the same set of data in parallel, using multiple databases). The decision to develop an extra layer of annotation to identify and tag grammatical errors on top of this system was based on the other rare but useful layers of annotation included in IMI that will be useful for future directions of this project – i.e., sense annotation (see discussion in Chapter 10).

The learner corpora tagging tool (see Figure 5.1) allows the use of any custom tagset, which can be organized in different classes/types of errors. And although some tags may be usable across languages (e.g., 'spelling error'), each language is ultimately expected and able to have specialized set of error tags.

Currently, the annotation is done within the context of a single sentence (i.e., annotators can only see one sentence at a time). However, the tool is also able to provide annotators with access to the full text of each document to capture the context of individual errors. This is especially relevant, for example, when addressing referential pronouns.

To tag each error, annotators can select a single word, a collection of words – contiguous (e.g., a phrase) or non-contiguous (e.g., a pronoun and its referent) –, or a full sentence. Multiple errors can be tagged for each sentence. Total and partial overlap of errors within the same sentence are also allowed. The ability to overlap errors allows annotators to tag the same error in more than one way (i.e., two error tags can be assigned to the same span of words, marking that the same error can be corrected in more than one way). This ability to overlap errors is also necessary when an error occurs within a larger error (e.g., an agreement error inside an overly long sentence).

One of the current limitations of this tool is that errors must be tagged directly on word tokens, which means that sub-word units can not be selected. Allowing the creation of errors that target sub-word strings, which would be useful for some classes of errors (e.g., morphological), will very likely be a future improvement of the system.

Missing words can be indicated by selecting words surrounding the location of the missing word. A text-box is provided for each error, and can be used to leave comments about each particular error, including a possible correction.

This system has proved itself robust and immensely useful to create two learner corpora that will be further discussed below.

5.2 The NTU Corpus of Learner English

The NTU Corpus of Learner English (NTUCLE, Winder et al., 2017) was developed in collaboration with NTU's Language and Communication Centre (LCC).

The primary motivation for assembling the NTU Corpus of Learner English (NTUCLE) was to inform the development of an automated system for corrective feedback on students' writing, which will be discussed in detail in Chapter 8. For this, it was important to gather quantitative data to support a better understanding of students' needs, especially in view of the future development of a system designed to help them.

A secondary motivation to assemble this corpus while involving lecturers from NTU's Language and Communication Centre was to understand commonalities and differences among

lecturers. It was hypothesized from the start that it would be difficult to build a system that would satisfy all lecturers equally – something that was confirmed by fairly marked differences in what each lecturer prioritized, and that will be discussed further below. In this regard, building the NTUCLE was also an exercise to iron out acute differences between lecturers, while attempting to harmonize the lecturers' standards.

Despite NTUCLE's likely usefulness for other fields and purposes, this corpus was not assembled with a broad research agenda in mind, in which it differs from, for example, the Cambridge Learner Corpus (Nicholls, 2003). At the same time, it was also not designed to be a ready-to-use data resource in the development of automatic grammatical error correction systems, which is the goal of many corpora such as the NUS Corpus of Learner English (Ng et al., 2014). This is essential to understand some of its design decisions, such as its focus on a large range of problems which go well beyond issues of grammaticality, or the fact that corrections for each sentence were not provided.

Some of these design choices could be easily changed, or admittedly improved, in the future. However, these choices are important to understand the scope with which this corpus was created, and why certain time-consuming tasks such as proving corrected versions for each problematic sentence were not taken up.

Finally, the NTUCLE also differs from other similar corpora in its narrower focus on a specific genre (i.e., technical proposals) and its target population (i.e., Singaporean, engineering undergraduate students).

The contributions of this corpus include a tagged dataset comprised of assignments submitted by first year engineering students, and the creation of a new error tagset which was then used to annotate the corpus. The remainder of this section presents a brief description of these contributions, including some general statistics concerning the annotation exercise. A fuller description and discussion can be found in Winder et al. (2017).

Corpus Compilation

Approval was obtained from the university's Institutional Review Board (IRB) for the research protocol and the use of students' written assignments from previous cohorts, subject to the

students' consent. In total, 349 students gave written consent for this, and their assignments were retrieved from NTU's learning management system to create the corpus.

Of the assignments retrieved, only files in doc/docx format were kept, because it would be difficult to automate text extraction, while preserving headings, paragraphs, style and sentence boundaries for the other formats (e.g., pdf). In the end, 273 documents satisfied this format requirement.

The documents were assignments from a communication skills course taught at NTU for first-year engineering students. Specific demographic details of the population could not be collected. However, based on course lecturers' observations, the course cohort are usually predominantly Singaporean (≈80%), with many likely to have native speaker proficiency in English, mostly male (≈70%), and between 18 and 22 years of age. The assignments in question consisted of a 500-word technical proposal and offered an engineering solution to a real life problem. The solution could be a new product, service or process, or an improvement of an existing one. The instructions for the assignment specified a structure for the proposal consisting of seven sections: background, problem, solution, benefits, implementation, costs/budget and conclusion.

Annotation Schema

A new annotation schema was created, largely based on pre-existing tagsets such as the one used by the NUS Corpus of Learner English (NUCLE, Dahlmeier et al., 2013b) – which shares many similarities in the demographic profile of learners – but also with reference to other established learner corpora such as the Cambridge Learner Corpus (CLC, Nicholls, 2003).

The tagset was developed as a collective effort, based on the experience of six professional English lecturers, and refined over multiple rounds of parallel and individual tagging and revision. Even though it is a known fact that excessive granularity can lead to greater difficulty in applying an annotation schema (Nagata et al., 2011), the final annotation schema was considerably larger than others, with 53 error tags sorted over 15 categories. This was motivated by the primary purpose of the corpus – to inform the development of a system capable of identifying and providing feedback on common problems in students' writing.

In addition, the NTUCLE includes many classes of errors that are not strictly grammatical, including tags that pertain to matters of style, some of which can be automatically detected. These tags include some simple style issues such as the use of contractions and colloquial words, but also more subtle ones such as overly long or convoluted sentences and missing parallel clause structures.

A full account of the finalized tagging schema, along with examples for each tag and frequency information on each type of error is available in Table 3 of Winder et al. (2017), also included at the end of this thesis as Appendix C.

Annotation Process

From the 273 documents collected in the NTUCLE, only a random sample of 180 documents were tagged due to time and manpower constraints. The 180 documents were split into groups of 40, and randomly assigned to one of 6 annotators, ensuring that 20 of these 40 documents were overlapped evenly with two other annotators (i.e., 10 documents overlapped with another annotator, and another 10 documents overlapped with a second annotator).

The annotators were six full-time professional English lecturers from LCC, NTU. Since this was done in the context of a collaboration, the lecturers did not receive any payment for this task. Each lecturer tagged the assigned documents independently, and the annotators knew neither which documents were being double tagged, nor the identities of the other annotators that might be tagging the same documents (this was important because not all lecturers overlapped with each-other). This was done to prevent annotators from adapting their annotation to known idiosyncrasies or *pet peeves* of other annotators.

A total of 60 documents were double annotated. Annotators were instructed to tag every error identified as specifically as possible, and to tag the same error with different error tags if the error could be corrected in different ways. It was decided that lecturers did not have to spend time correcting each error because it would greatly increase the complexity of the task, especially in identifying all possible options for correcting each error while preserving the student's intended meaning (Sakaguchi et al., 2017). This would have required more time and resources than were available – although this can still be added anytime in the future.

The annotation was done online, using the new Learner Corpora Tagging Tool presented in Section 5.1.

Annotation Results

The annotation exercise revealed that the lecturers varied widely in their sensibility to errors. This is unfortunate although not at all unusual, as similar differences are often observed in similar annotation exercises (see, for example, Bryant and Ng, 2015). In the NTUCLE, this is evident not only from the number of errors tagged by each annotator, ranging from 380 (Annotator 2) to 1,183 (Annotator 3), but also to varying sensitivity to particular errors and different tagging practices. Evidence for this is shown in Table 5.1. The lecturers also differed in the frequency with which they tagged the same word or word string with different tags to acknowledge different ways of identifying errors.

A#	Errors	1st MCE	2nd MCE	3rd MCE
A1	1,101	awk. expr. (21%)	word choice (11%)	unclear expr. (10%)
A2	380	sg./pl. forms (22%)	word choice (7%)	miss. art./det. (6%)
A3	1,183	sg./pl. forms (12%)	miss. art./det. (10%)	word choice (8%)
A4	556	miss. art./det. (21%)	sg./pl. forms (11%)	verb form (9%)
A5	908	unclear expr. (12%)	awk. expr. (11%)	word choice (7%)
A6	972	sg./pl. forms (11%)	word choice (9%)	miss. art./det. (9%)
Total	**5,100**	**sg./pl. forms (10%)**	**miss. art./det. (8%)**	**word choice (8%)**

Table 5.1: Most Common Errors (MCE) by annotator

A closer inspection of Table 5.1 reveals that three annotators (A2, A3 and A6) were, in fact, quite similar in their tagging patterns in relation to the three main error categories tagged, namely 'singular/plural forms', 'missing article/determiner' and 'word choice' – which were also the top three error categories overall. Two other annotators (A1 and A5) also had similar top three error categories ('word choice', 'awkward expression' and 'unclear expression') but these were the third, fourth and fifth most common error categories tagged overall. The formation of two clusters is noteworthy, as it can show that while the first group of annotators shows some sensibility to grammatical issues, the second group of annotators tagged mostly problems concerned with style or content – which fell outside the scope of what the system being developed had the ability to deal with.

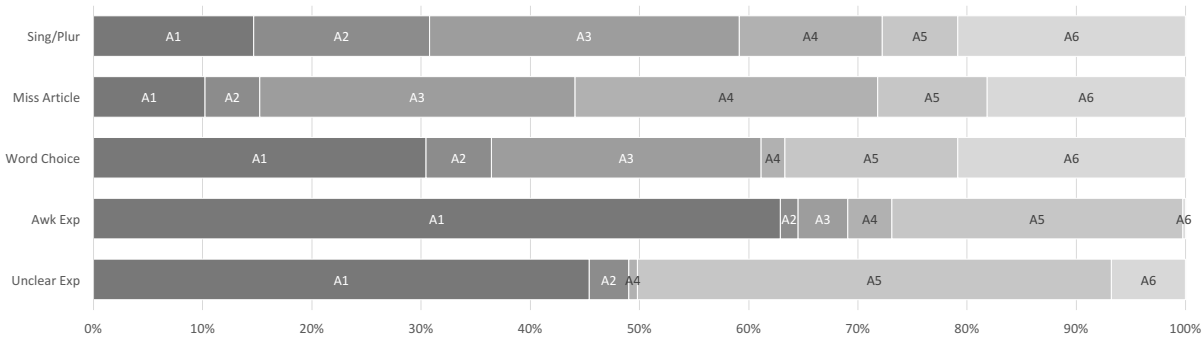


Figure 5.2: Contributions of annotators to top five errors tagged

The five most common errors, distributed by annotators, are shown in Figure 5.2: errors in using singular or plural forms, omitting articles or determiners, choosing inappropriate words, using awkward expressions and using unclear expressions. However, as mentioned above, the annotators showed very different emphases in their annotation. While overall, strictly grammatical errors (i.e., use of singular/plural forms and omission of articles and determiners) were the most commonly identified, annotators 1 and 5 identified far more errors in 'expression' (unclear/awkward), which may relate more to issues in semantics or idiomaticity.

Adjudication of Overlapped Documents

As has been mentioned above, 60 out of 180 documents were tagged by two different annotators. In most cases, both annotators tagged the same errors in the same sentences, either for exactly the same sentence span or for sentence spans with overlapping words. However, there were also significant differences, such as different sentence spans tagged for the same error type, or the same sentence span tagged for different error types.

All the annotators met to review every instance of different error tags assigned to the same exact sentence spans (words, expressions, etc.). In most cases, it was agreed that one or both of the annotators had made a mistake, either unintentionally, or through misunderstanding or misapplying a tag. The relevant word strings were then re-tagged with the correct tags. In a few instances, annotators tagged only one of multiple possible corrections for the same sentence: e.g., *However, trolley has its own limitations*, which can be construed as either 'NNum' [wrong

choice of singular/plural form of the noun] (*However, trolleys have their own limitations*) or 'AMiss' [missing article/determiner] (*However, a trolley has its own limitations*). In yet a few others, both tags apply, i.e., there are two errors conflated in the same word set (e.g., *For example, individual seats for individual cubical will be installed with motion sensor.*, where there is both an 'MSpel' [spelling error] (*cubical* to *cubicle*) and an 'NNum' [wrong choice of singular/plural form of the noun] (*cubical* to *cubicles*) error). In both these kinds of cases, it was agreed that both tags should remain, in the first instance because there are two ways of correcting the error, and in the second, because there are two overlapping errors.

Some differences arose because the lecturers involved found it difficult to use existing tags. In many of these instances, one of the annotators tagged the error under 'Others' and provided their own labels or comments. From these, a few new categories of errors were created and used instead, namely 'StyMood' for the inappropriate use of imperatives or interrogatives, 'SLong' for overly long sentences, and 'SConv' for convoluted sentences.

The lecturers also discussed their own 'pet peeves' in the texts they annotated. Among those most commonly shared was the problem of overly long sentences that made comprehension difficult. This reinforced the need for the category 'SLong'. Another commonly shared 'pet peeve' was the inappropriate over-use of certain colloquial words and informal clichés, *tackle* (to mean *study*, *address* or *solve* a problem) and *hassle* (to mean *inconvenience* or the like) being two of the most common. Another new category 'StyWch' for the use of casual or colloquial words and expressions was created to tag such words.

A more extensive discussion on the process of double tagging and revision of the annotation schema can be found in Winder et al. (2017). This adjudication process ultimately resulted in the production of a finalized error tagset, with 15 categories and 53 tags. This tagset is included at the end of this thesis as Appendix C.

Discussion

It is evident that some of the discrepancies introduced during the annotation of the NTUCLE could have been avoided, making the tagging process more streamlined and efficient. However, this would have required coming up with stricter guidelines for the annotators, or perhaps a few

rounds of standardization exercises prior to the annotation. However, this would go against the main goal of this corpus. As it has been stated above, the NTUCLE was primarily designed to collect data from students and lecturers alike. With this in mind, this corpus was able to showcase the natural discrepancies in human grading and possible foci of corrective feedback. Since the primary goal of the NTUCLE was to inform the development of a system that tried to emulate this human feedback, providing constructive feedback on issues lecturers usually highlight in student assignments, it was an important part of the experiment to pursue this kind of naturalistic tagging, which captures differences in grading expectations, editing experience and perceptions of acceptable or exemplary language use (Daudaravicius et al., 2016b; Rozovskaya and Roth, 2010).

Corpus Release

The final release of the NTUCLE includes six databases (one per annotator), all of them following the database schema used in IMI (Bond et al., 2015). Only anonymized data will be released, and this will be done under an Attribution 4.0 International license (CC BY 4.0),[1] in conformity with our IRB and the students' consent. The use of such an unrestricted license is a contrast to the common practice of similar corpora which often have restrictive non-commercial and/or fair-use only licenses (e.g., the CLC, or the NUCLE).

DB	Docs.	Overlapped Docs.	Sents.	Words	Sents. w/Errors	Errors
A1	40	10 (A6) + 10 (A2)	2,051	26,176	812	1108
A2	40	10 (A1) + 10 (A3)	2,144	26,764	372	390
A3	40	10 (A2) + 10 (A4)	2,269	27,603	625	1193
A4	40	10 (A3) + 10 (A5)	2,223	27,246	361	575
A5	40	10 (A4) + 10 (A6)	2,093	26,654	579	908
A6	40	10 (A5) + 10 (A1)	2,024	26,103	564	972
All	180	n.a.	9,571	119,727	2,751	4,860

Table 5.2: The NTUCLE Corpus Release in Numbers

Table 5.2 provides a quick overview of the tagged portion of the corpus: number of documents, overlaps, number of sentences, number of word tokens, number of sentences that contain

[1] https://creativecommons.org/licenses/by/4.0/

at least one error label, and the total number of errors included in each database. In the future, this release will also include a compiled database of the 180 tagged documents, merging the annotation of documents that were double tagged. And while the compiled database will allow for more traditional usages, the individual databases can be used to further analyze and discuss individual differences and idiosyncrasies among annotators.

5.2.1 The Expanded NTUCLE (NTUCLE-X)

From the moment the NTUCLE was created, new documents have been collected and slowly added to this corpus every semester. These documents were collected automatically, through the use of the system that will be discussed in detail in Chapter 8. The continuous collection of assignments using this system continues to follow strict IRB standards, including seeking explicit permission from students to use their assignments, and ensuring that all data is anonymized before being released to the public.

Currently, the NTUCLE-X has 1,968 students assignments (including those originally belonging to the NTUCLE). Many of these, however, are actually different versions of the same assignment, as students were able to submit their assignments multiple times to the system.

All assignments contained in the NTUCLE-X are from the same course used to create the NTUCLE, and the documents are structurally identical to the ones described for the original version of the NTUCLE. However, even though both corpora are similar in content, there are a few differences between the NTUCLE and the NTUCLE-X worth noting:

- the data contained in the NTUCLE-X was preprocessed very differently than the data for the NTUCLE. For the NTUCLE-X, a preprocessing step was used to ignore certain sections of the documents that were considered not relevant to the work in question (e.g., tables, figure legends, mid-sentence citations, bibliography, etc.). And while this should be considered an improvement to the data, it does cause some incompatibilities when comparing the same document between the NTUCLE and the NTUCLE-X;
- the assignments collected by the NTUCLE-X have not been hand-tagged;
- due small changes in the course curriculum, most assignments included in the NTUCLE-X were written in pairs, instead of by a single student, and have about 800 words instead

of the original 500;

- the NTUCLE-X includes multiple versions for the same assignment, preserving student history of submissions to a system providing corrective feedback;

Despite some backwards incompatibility of the tagging provided for the original version of the corpus, the NTUCLE-X is essentially a new and improved version of the same corpus.

The release of this new version of the corpus is tied to the same IRB constraints as the original version. Only anonymized data will be released, and this will be done under an Attribution 4.0 International license (CC BY 4.0).[2]

The first release of the NTUCLE-X contains 802 assignments and more than 24,500 sentences. These assignments were anonymized by two undergraduate students and hand-checked by me to ensure no information could be traced back to the original authors.

A small portion (≈20%) of the NTUCLE-X was used to build a new treebank focused on learner English. This will be discussed in greater detail in Chapter 7.

5.3 The NTU Corpus of Learner Mandarin Chinese

The NTU Corpus of Learner Mandarin Chinese is a new learner corpus collected from early learners of Mandarin Chinese, similar in spirit to the NTUCLE.

The main motivation to develop this corpus was to inform the design of the mal-rule enhanced ZHONG, enabling this grammar to perform grammatical error detection and diagnosis – which will discussed in detail in the next chapter. Similarly to what was discussed above, for English, while it would be possible to rely on lecturers intuitions or other published materials to come up with a list of common errors, it was decided that relying on first-hand data would provide unequivocal evidence about the problems that deserved attention. In addition, producing a corpus would also be useful to provide a suitable dataset to evaluate the development of mal-rules for ZHONG.

[2] https://creativecommons.org/licenses/by/4.0/

Corpus Compilation

This corpus was collected with the the support of the Chinese teaching team at NTU's Centre for Modern Languages after it obtained approval from the NTU's Institutional Review Board.

The data was sourced from student exams for the first level of Mandarin Chinese lectured at NTU. According to the approved research protocol, explicit student permission was not needed as long as the data remained private. While it would have been preferable to work with open data, the Centre for Modern Languages preferred that the data remain private.

Two exercises from a recent exam that was still in archive were selected. The choice of exercises was based on relevance to the corpus in question. The first exercise comprised a list of sentences for students to decide whether they were grammatical or ungrammatical, and requested students to provide a corrected version of the sentence if they thought it was ungrammatical. Most sentences were ungrammatical and required correction. This was a very relevant exercise, since it not only required students to have awareness of common errors they should avoid, but also frequently showcased other types of errors through failed attempts to correct the sentence. If students judged an ungrammatical sentence as grammatical, that meant that they also did not understand the error, which was also considered an error for the corpus.

The second exercise was a composition. Students were asked to write a dialogue between two exchange students who meet at a dormitory in China. The composition needed to have at least 100 words. This exercise was relevant due to the freedom students had to compose their own sentences, and was ideal to collect naturally occurring errors (i.e., without leading students towards a particular type of error).

In total, 85 exams were randomly selected, and the relevant exercises digitized (by hand) from scanned copies of the exams. This digitization was done by two undergraduate student assistants, fluent in Mandarin Chinese. Altogether, 5,513 sentences were extracted from these exams.

The student answers often included instances of Hanyu Pinyin (the official romanization for Mandarin Chinese) instead of Chinese characters. This was contemplated under the exam rubric, and students knew that they would lose points for this. However, allowing Hanyu Pinyin

was preferred to preventing students from providing full answers. During the digitization process, the student assistants were requested to keep two parallel records for each extracted sentence: the exact copy of the student's answer; and a version converted to characters based on the meaning/context of the answer.

Annotation Process

The annotation process of the NTUCLM happened in 2 phases. The first phase uses the exact answers of the students (i.e., including romanized forms if the students had used them). The annotation was done by two undergraduate student assistants, with linguistic background and native level fluency in Mandarin Chinese. The annotation was done online, and used the new tool presented in Section 5.1.

In collaboration with Mandarin Chinese lecturers at NTU, a custom list of 21 expected error tags was compiled. Out of these 21 tags, one was reserved for problems concerning tones (which will not be discussed in detail). The 20 remaining tags are listed in Table 5.3. The last one of these tags, labeled 'other', was meant to capture all errors not predicted by the existing tags.

Similar to what happens with the English corpus, some classes of errors suggested by Mandarin Chinese lecturers should not be strictly considered grammatical errors. A good example for this is the Error ID 18, '姓 used as a noun'. While using the word 姓 (xìng, *surname/to be surnamed*) as a noun is not technically a mistake, this word is most often used as a transitive verb with the meaning *to be surnamed*, taking a surname as complement. Using it as a noun may sound less fluent, and it should be brought to students' attention. Many other tags in this tagset have similar concerns which will be discussed, individually and in detail, in Section 6.6 – in the context of the creation of individual mal-rules.

The corpus was split in half, and each half was tagged by a different tagger, without overlap. Due to the exploratory nature of this tagset, both taggers were in constant communication with each-other and with me. Errors that had not been foreseen by the tagset were captured using the tag 'other' and further specified by semi-structured comments agreed by both taggers.

This first phase of annotation resulted in about 1,660 errors tagged over more than 1,360 sentences (roughly 24.7% of the corpus). About half of these errors signaled incorrect or miss-

ing tone information in romanized sections of student answers. Unfortunately, errors pertaining to tones are largely irrelevant to the the development of mal-rules, and were not analyzed in detail. Nevertheless, this data was kept and can be used in future projects.

The second phase of annotation for this corpus happened while it was being treebanked – a process which will be discussed in further detail in Section 7.1.2.

However, before the corpus could be treebanked, it needed to go through some changes. The main concern was that the treebanking process expected only sentences written in Chinese characters. As mentioned above, each sentence in this corpus had both its original surface form and a converted form, digitized using only Chinese characters (except in cases where it was deemed impossible to understand from the available context).

Given the nature of the exam questions from which the corpus was sourced, and the limited proficiency of its student authors, it was expected that many sentences would be repeated across exams – including both problematic and non-problematic sentences. This was also true for many sentences without the same surface form, which could be considered repeated if compared using their converted form.

Treebanking the same sentence multiple times would be a waste of limited resources. So, it was decided that repeated sentences would be merged. Using the converted form as pivot, all sentences with the same converted form were merged into a single sentence. Similarly, all errors tagged for merged sentences (except errors concerning tones) were also merged. This process created a much smaller corpus, with 2,300 unique sentences. And because this version of the corpus contained only sentences written in Chinese characters, problems concerning tones were not kept with this version.

During the treebanking process, which will be described in greater detail in Section 7.1.2, each of these sentences were revised in relation to their grammaticality, along with other non-grammatical problems they had (e.g., awkward semantics). During this phase, most sentences were treebanked by either 2 or 3 student assistants.

After the treebank was complete, the results from phase 1 and phase 2 were merged. I personally reviewed each instance where a problem was raised in either phase 1 or phase 2, and decided on a final tag using the same tagset defined for phase 1.

Results and Discussion

The final version of this corpus, along with the frequency of each error tag can be see in Table 5.3. This 2,300 sentence corpus contains 544 error tags divided among 490 problematic sentences. In other words, around 21.3% of the sentences in the corpus have at least one error tag assigned to them, and each problematic sentence has an average of 1.1 errors.

ID	Description	Total
1	吗 (*ma*, question particle) redundancy	26
2	Usage of 和 (*hé*, and) vs. 也 (*yě*, also)	25
3	Position of adverbial clauses	25
4	Usage of 是 (*shì*, to be) with adjectival predicates	23
5	Usage of 中国 (*zhōngguó*, China) vs. 中文 (*zhōngwén*, Chinese language)	18
6	Position of 也 (*yě*, also)	14
7	Usage of 有点儿 (*yǒudiǎnr*, somewhat) vs. 一点儿 (*yīdiǎnr*, a bit)	14
8	Bare adjectival predicates	9
9	Usage of 是... 的 (*shì...de*, focus cleft) constructions	8
10	Usage of 不 (*bù*, no) with specified adjectival predicates	6
11	Incorrect measure word	6
12	Missing measure word	5
13	Attributive 多 (*duō*, many) and 少 (*shǎo*, few) without degree specifiers	5
14	Usage of 二 (*èr*, two) vs. 两 (*liǎng*, two)	4
15	Usage of 不 (*bù*, no) vs. 没有 (*méiyǒu*, no)	3
16	Syntactic order of 也 (*yě*, also), 都 (*dōu*, all), 不 (*bù*, no)	3
17	Syntactic order of nominal 的 (*de*, possessive marker) modification	2
18	姓 (*xìng*, to be surnamed) used as a noun	0
19	Issues with numerical phrase predicates	0
20	Other Errors	348
	Total	544
	Sentences w/errors	490

Table 5.3: Distribution of Error Tags by Frequency

From the list of error frequencies, it is clear that the overwhelming majority of problematic sentences ended up receiving the tag labeled 'other'. This happened for multiple reasons:

- Many sentences had in fact problems that could be well clustered but that lacked a tag. Examples of these would include orthographic mistakes, missing verbs, semantically awkward phrases, or use of English words;
- Another large group of sentences had problems that could not easily be diagnosed with a single or even multiple tags. These included sentences where the order of words in a

sentence seemed practically random, and it was difficult to understand the thought process behind the error; and finally,

- A third group of sentences were included in this category due to narrow or sometimes ill-defined error tags. The best example for this problem concerned Error ID 2: 'Usage of 和 (*hé*, and) vs. 也 (*yě*, also)'. This tag was originally defined as a misuse of the conjunction 和 (*hé*, and) to join verb clauses (which should be accomplished through the use of a comman and the adverb 也 (*yě*, also) instead). However, many sentences contained problems related to 也 (*yě*, also) that were not clear if they should be tagged with this label – e.g., the use of 也 (*yě*, also) as a conjunction between two nouns;

Looking at the remaining frequencies, it is also relevant to point out that some expected classes of errors were not present in the corpus (e.g., Error ID 18 and 19). This most likely reveals some fallacies in the intuition of language lecturers, as other types of errors that did not receive a label (and were clustered as 'other') were in fact much more frequent the many of the lower frequency errors (e.g., missing verb).

Some of the problems raised above, concerning the error label 'other', show that the quality of the tagset used for the NTUCLM still has plenty of room for improvement. However, it is important to note that mal-rule development for ZHONG was informed not by the tagset specifically, but by the many instances of errors that were inside each error label. Due to the syntactic nature of different errors, some error labels were split into multiple different mal-rules, effectively showing that an error label could be split into multiple fine-grained classes.

However, since the data contained in this corpus cannot be made publicly available, and also since the evaluation of ZHONG's mal-rules was measured at the sentence level and not at the error label level, it was not deemed a worthy time investment to rework and re-tag this corpus with a new error tagset. The creation of a gold-standard error tagset for Mandarin Chinese is, however, a very interesting direction for future work.

Development and Evaluation sets

Since the NTUCLM was the main data source informing the development of mal-rules for ZHONG, it was important to plan the evaluation of this task as early as possible. As such,

the NTUCLM was split into two sub-corpora: the NTUCLM train/development set and the NTUCLM test/evaluation set.

The NTUCLM development set was used to collect vocabulary, guide the improvement and development of syntactic analyses in ZHONG, and guide the design and development of mal-rules. The evaluation set was reserved and used strictly for evaluation purposes.

ID	Description	Train	Test	Total
1	吗 (*ma*, question particle) redundancy	21	5	26
2	Usage of 和 (*hé*, and) vs. 也 (*yě*, also)	23	2	25
3	Position of adverbial clauses	22	3	25
4	Usage of 是 (*shì*, to be) with adjectival predicates	18	5	23
5	Usage of 中国 (*zhōngguó*, China) vs. 中文 (*zhōngwén*, Chinese language)	12	6	18
6	Position of 也 (*yě*, also)	12	2	14
7	Usage of 有点儿 (*yǒudiǎnr*, somewhat) vs. 一点儿 (*yīdiǎnr*, a bit)	8	6	14
8	Bare adjectival predicates	7	2	9
9	Usage of 是... 的 (*shì...de*, focus cleft) constructions	8	0	8
10	Usage of 不 (*bù*, no) with specified adjectival predicates	4	2	6
11	Incorrect measure word	4	2	6
12	Missing measure word	5	0	5
13	Attributive 多 (*duō*, many) and 少 (*shǎo*, few) without degree specifiers	4	1	5
14	Usage of 二 (*èr*, two) vs. 两 (*liǎng*, two)	3	1	4
15	Usage of 不 (*bù*, no) vs. 没有 (*méiyǒu*, no)	1	2	3
16	Syntactic order of 也 (*yě*, also), 都 (*dōu*, all), 不 (*bù*, no)	2	1	3
17	Syntactic order of nominal 的 (*de*, possessive marker) modification	2	0	2
18	姓 (*xìng*, to be surnamed) used as a noun	0	0	0
19	Issues with numerical phrase predicates	0	0	0
20	Other Errors	286	62	348
	Total	442	102	544
	Sentences w/errors	399	91	490

Table 5.4: Distribution of Error Tags in the Development and Evaluation Sets of the NTUCLM

With this in mind, a random sample of 10% of correct sentences and 20% of problematic sentences were set aside as the NTUCLM test/evaluation set. However, this split was done before the phase 2 of the annotation process, which means that the results of the annotation done during phase 2 were retroactively merged into the development and evaluation sets. As such, the actual numbers in relation to the final set of data differ slightly from the original 10%

and 20% mentioned above.

After merging the results of phases 1 and 2, the evaluation set contains 287 sentences: 196 correct sentences and 91 sentences tagged with at least one error label. The development set contains 2,013 sentences, 399 of which are tagged with at least one error label.

The final proportions of correct and problematic sentences in relation to the final version of the corpus (with 2,300 sentences) is: the NTUCLM evaluation set contains 196 (or 10.8%) of the correct sentences and 91 (or 18.6%) of the problematic sentences; and the NTUCLM development set contains 1,614 (or 89.2%) of the correct sentences and 399 (or 81.4%) of the problematic sentences.

Due to the scarcity of certain classes of errors, and because this split was done randomly, not all classes of errors are proportionally represented in both sets. The frequency of each error label in each set is shown in Table 5.4.

5.4 The Mandarin Education Corpus

The third and final corpus presented in this chapter is the Mandarin Education Corpus (MEC) – a corpus of educational material collected to guide the development and evaluate ZHONG's broader syntactic coverage.

This corpus compiles data from four different sources, each of which will be introduced below. Similarly to what happened with the NTUCLM, the MEC was also split into development and evaluation sets – following the same definitions given above. A summary of the size and distribution of this corpus by set is provided in Table 5.5, below.

The CMNEDU Corpus

The MEC includes a previously collected corpus, internally known as the CMNEDU Corpus. This corpus comprises data from two beginner Mandarin Chinese textbooks: the first volume of the New Practical Chinese Reader (Xun, 2010), and the first volume of Chinese Link: Beginning Chinese, Simplified Character Version (Wu et al., 2010).

The New Practical Chinese Reader is a six-volume series, published by the Beijing Lan-

	Set Name	Set Size	Avg. Sent. Length
Development Data	cmnedu	798	7.74
	tufs	1,531	6.46
	hsksc_01	175	5.71
	hsksc_02	200	7.92
	hsksc_03	81	9.42
	hsksc_04	200	10.51
	hsksc_05	200	11.89
	hsksc_06	157	13.48
Evaluation Data	hsksc_07	200	16.77
	hsksc_08	200	19.84
	hsksc_09	30	22.23
	hsksc_10	200	21.71
	hsksc_11	200	23.12
	hsksc_12	67	22.97
	tatoeba_01	10,000	8.47
	tatoeba_02	10,000	7.95
	tatoeba_03	10,000	7.94
	tatoeba_04	10,000	7.44
	tatoeba_05	7,216	7.23

Table 5.5: MEC Summary: split by development and evaluation sets

guage and Culture University Press and widely used by the Confucius Institute worldwide. The Chinese Link is an out of print series, published by Pearson, and was being used by the NTU Mandarin Chinese curriculum when this corpus was first compiled.

The MEC contains 798 sentences extracted from the main texts of these two textbooks. Since this corpus was the main dataset used in the development of ZHONG's first version, it was kept in MEC as a development set.

For copyright reasons, this data cannot be released publicly.

TUFS Open Language Resources

The MEC also includes 1,531 sentences made available from the TUFS Open Language Resources (Kawaguchi et al., 2007), published by the Tokyo University of Foreign Studies (TUFS). This data resource includes basic vocabulary and example sentences in 24 languages, and was also used to build the TUFS Asian Language Parallel Corpus, or TALPCo (Nomoto et al., 2018).

The data is publicly available under an open license (CC BY 4.0).[3]

The data in this resource is based on 799 basic vocabulary words and example sentences selected in accordance with the lowest level of the Japanese Language Proficiency Test (N5). Some of the vocabulary in this resource is slightly old-fashioned (i.e., words that we would not be considered basic vocabulary today), and sometimes even Japan-centric (e.g., including words like 'Japanese style room'). The data is used to power an online language learning platform known as Tokyo University of Foreign Studies Language Module.[4]

During the creation of this sub-corpus, I have also helped develop a multilingual resource of basic vocabulary linked to the Open Multilingual Wordnet (Bond et al., 2020). However, this work will not be presented as part of this thesis.

The Mandarin Chinese portion of this resource was added to the MEC. Due to its permissive license and controlled proficiency level, this sub-corpus was also chosen to feature in the development section of the MEC. This sub-corpus was treebanked using ZHONG (presented in Chapter 7), and will be made publicly available.

The HSK Standard Corpus Textbook Collection

The MEC also contains data from the first five volumes of the HSK Standard Course textbook collection (Liping, 2015), published by the Hanban. The Hanban is a public institution affiliated with Ministry of Education of the People's Republic of China (PRC), and is responsible for administering the HSK[5] – the official Mandarin Chinese proficiency exam of the PRC.

This textbook series differs from other mainstream textbooks by introducing vocabulary and grammar structures in the order the HSK will test students. The large majority of other textbooks focus on a classroom experience, and introduce vocabulary and grammar topics without necessarily maximizing the student's ability to clear the HSK exams as fast as possible.

In total, 1,910 sentences were extracted from the first five volumes of this textbook series. These sentences were split in 12 graded sets (see Table 5.5): set hsksc_01 covers the first level of the HSK exam; sets hsksc_02 and hsksc_03 cover the second level of the HSK exam; sets

[3] `https://malindo.aa-ken.jp/TUFSOpenLgResources.html`
[4] `http://www.coelang.tufs.ac.jp/mt/en/`
[5] `http://www.chinesetest.cn/index.do`

hsksc_04 through hsksc_06 cover the third level of the HSK exam; and sets hsksc_07 through hsksc_12 cover the two volumes pertaining to the fourth level of the HSK exam.

The increased language complexity of these sets can be confirmed by the increase in average sentence length of each set as the sets progress, shown in Table 5.5. Given the expected language complexity of each set, the data concerning the first three levels of the HSK exam (sets hsksc_01 through hsksc_06) was used as development data. And the remaining sets, containing data from the fourth level of the HSK exam, were kept as evaluation data.

Due to copyright reasons, this data cannot be released to the public.

The Tatoeba Corpus

The MEC's fourth data source is the Tatoeba Corpus.[6] The Tatoeba Corpus is an online and ongoing multilingual project that resorts to crowdsourcing to maintain a very large bank of linked sentences across hundred of languages. This corpus has its roots and is the current home of the the famous Tanaka Corpus (Tanaka, 2001).

This very large corpus includes data shared under multiple licenses, which are set for each sentence by their contributors. From a dump collected in December 2020, the MEC contains 47,216 sentences in Mandarin Chinese from the Tatoeba Corpus, all of which were released under an open (CC BY) license. These sentences were split into five sets for logistic reasons, and not because the content between sets differs in any meaningful way. The random nature with which the corpus is built, using permanent IDs when a new sentence is added, should ensure a homogeneous distribution of language complexity across all sets.

It is also important to reinforce the idea that the Tatoeba corpus is specifically catered for language learners. This is reflected by the relatively short sentence length, and the fact that most sentences contain only basic (i.e., non-specialist) vocabulary. By inspecting Table 5.5, one can see that the average sentence length in the Tatoeba Corpus is similar to the values presented for the CMNEDU corpus, and to the data pertaining to the second level of the HSK exam. For comparison, the NTUCLM has an average sentence length of 7.04 words per sentence.

Due to the fairly large size of this corpus, and a complexity level expected to slightly surpass

[6] `https://tatoeba.org/en`

the complexity of the NTUCLM, this data was kept as evaluation data.

5.5 Summary

This chapter reported on the creation of a new open-source online system to build learner corpora, as well as the compilation of three corpora. These corpora include two versions of the NTU Corpus of Learner English (NTUCLE) – a hand-tagged version containing 9,571 sentences and 4,860 error tags, and an extended version of this corpus containing more than 24,500 sentences. This data was used to inform the development of an automated system for corrective feedback on students' writing, which will be discussed in detail in Chapter 8, and will be shared under an open license.

In addition to the NTUCLE, a second learner corpus was developed: the NTU Corpus of Learner Mandarin (NTUCLM). The NTUCLM contains 2,300 unique sentences collected from answers to Mandarin Chinese exams administered at NTU. This corpus has 544 hand tagged errors, and was used to inform the development of mal-rules in ZHONG, discussed in Chapter 6. Despite licensing restrictions preventing this data from being publicly released, the knowledge extracted from this data will be indirectly shared through the design and implementation of mal-rules that will be released with ZHONG – which is an open source grammar.

Finally, this chapter also introduced the Mandarin Education Corpus (MEC), a dataset with 51,455 sentences collated from multiple educational sources, including language textbooks and public datasets. This corpus was collected both to inform the development of ZHONG, and to evaluate the improvements that will be presented in Chapter 6. Due to a careful selection of data sources, a large portion of this dataset will be able to be shared under an open license.

Chapter 6

Extending ZHONG

This chapter presents the main contributions made to the ZHONG (Fan, 2019) – an open-source HPSG grammar for Mandarin Chinese. The nature of these contributions vary widely, and include bug-fixes, reanalyses of certain phenomena, implementation of missing phenomena and, of course, the implementation of mal-rules.

It is impossible to provide an exhaustive description of all changes made to ZHONG – these changes can be inspected through the history of changes recorded in ZHONG's repository. I will, however, present an overview of some of the most important linguistic topics that were added or improved upon. I will finish this chapter with a non-exhaustive list of mal-rules added to ZHONG, which enable this grammar to detect a variety of common errors identified in the NTU Corpus of Learner Mandarin.

6.1 Theoretical Description vs. Implementation

This chapter provides multiple HPSG descriptions concerning the implementation of certain phenomena and mal-rules. As such, it is important to note that there are often some differences between so called "hand-written" descriptions and their respective implementations. The degree to which an implemented grammar resembles the "hand-written" one is coined as *faithfulness* by Melnik (2007), who also discusses some of the reasons why being fully *faithful* is not always possible or even desirable. In this thesis, in particular, since an exhaustive description of ZHONG's implementation choices would be necessary to understand and describe the

actual implementation of any rule or construction, this chapter will overtly focus on "hand-written" descriptions using general HPSG principles and features. The first motivation for this is readability, allowing readers to access core information without burdening them with all the necessary information to describe the actual implementation. The price for this is, admittedly, paid in *faithfulness* of the descriptions that will be provided. For the most part, it can be assumed that some particular aspects of the implementation differ from the descriptions provided here. To the best of my ability, the names of types, rules and features will remain faithful to those present in ZHONG. However, most descriptions will be simplified to provide an adequate level of discussion. Since ZHONG is an open-source project, the full extent of the implementation can inspected by those who choose to do so.

6.2 Extending the Lexicon

A substantial amount of work went into improving ZHONG's lexicon, including some changes to its type hierarchy to accommodate new lexical types previously missing. Contrary to what had been done in the original version of ZHONG (see Section 4.2), all lexicon acquisition done during this thesis was performed manually. This was accomplished using the development data of the Mandarin Education Corpus (see Table 5.5) and the development portion of the NTU Corpus of Learner Mandarin.

A semi-automated process was used to find missing words in ZHONG's lexicon by parsing each sentence and then using the parsing logs of failed sentences to come up with a list of words that did not find a leaf node in the grammar. This indicated that a sentence either had a word that was not included in ZHONG's lexicon, or that the sentence had problems concerning word segmentation. A small discussion on word segmentation will be included later in this chapter.

As introduced in Section 4.2, ZHONG's lexicon acquisition had mainly been automated, which generated a lot of noisy data. In an attempt to start sorting this data into hand-checked data, a new lexicon file was created to host only fully hand-checked vocabulary. Most new entries were added with extra documentation, including pronunciation, gloss and example sentences. An example of the lexical entry for the transitive verb 复印 (*fùyìn*, 'to photocopy') is

shown in (35).

(35)

```
复印_v_1 := v_np_le &
"""
[fùyìn] to photocopy, to duplicate a document
<ex> 他 每天 复印 资料 。
(He photocopies materials every day.)
"""
  [ STEM < "复印" >,
    SYNSEM [ LOCAL.CAT.HEAD.CHAR [ FCHAR "复", LENGTH two ],
             LKEYS.KEYREL.PRED "_复印_v_1_rel" ],
    TRAITS native_token_list ].
```

Within DELPH-IN, HPSG is formalized using the Type Description Language (TDL, Krieger and Schäfer, 1994). Looking at (35), one can see the basic definition of a type in TDL. This type happens to be a lexical entry (i.e., a leaf lexical type), but the same basic syntax would be used for all other types.

The name of this type is `复印_v_1`. The sign `:=` assigns the definition of the type. In this case, this definition includes the parent type, `v_np_le` (the type for basic transitive verbs taking an NP as complement – which is defined elsewhere in the grammar), followed by a conjunction sign (i.e., `&`) and a feature structure that further constrains any features introduced by the parent type. In this case, this feature structure defines a few important features relevant to a lexical entry. The feature `STEM` takes as value a list of tokens to be matched in the sentence surface form (i.e., a single list with the word `< "复印" >`). This lexical entry also includes other features such as the first character of the `STEM` (i.e., `FCHAR "复"`) and the length of the `STEM` in characters (i.e., `LENGTH two`). Finally, the last two features are `PRED "_复印_v_1_rel"` and `TRAITS native_token_list`. The first stores the string corresponding to the semantic predicate of this lexical entry, and the second is an internal feature marking this lexical entry as native to the grammar (i.e., not generated during preprocessing).

Between the parent and the additional feature structure, lies the documentation string (marked

by triple quotes on each side). This documentation string is useful to store information about the *why* and *how* a type was created, as well as expected interactions with other parts of the grammar. This kind of documentation is essential to maintain a large grammar, and is a basic requirement to maintain a project open to new collaborations. Tools like the Linguistic Type Database (Hashimoto et al., 2007, 2008), introduced in Section 4.3, make use of documentation strings, along with other information available in the grammar, to provide an interface to inspect and understand the design of a grammar.

Even though documentation strings are not usually used for lexical entries – because they are fairly simple types and require little explanation –, I have used documentation strings to store information that is useful to a grammarian, especially one that is not native to the language being analyzed. This includes the romanization and a gloss for the lexical entry, as well as one or more glossed example sentences showcasing the word behaving as defined by their type.

Regardless of the level of fluency a grammarian may have, the process of grammar engineering requires the grammarian to make many decisions concerning the grammaticality of certain phenomena. For lexical entries, these decisions can include whether a word can be used as a certain part-of-speech, or which are the available types of subcategorization available to a certain verb. The main motivation to add example sentences to lexical entries was to document their usages. In certain cases, a particular lexical behavior may not be extremely productive. If not documented properly, these lexical entries exist in a sort of limbo, where the grammarian is not entirely sure if that entry should exist or not.

A related problem existed within ZHONG due to the fact that the majority of the vocabulary had been collected automatically. As already discussed above, this was a fairly noisy process – generating many incorrect and/or unnecessary entries. However, as a non-native speaker, I often felt reluctant to delete lexical entries just because they seemed incorrect at a first glance. Even though I did not recognize a particular use case of a word, it did not mean that a corner case where that word could indeed have the predicted behavior did not exist. However, having incorrect or unnecessary lexical entries can also be very detrimental for a grammar – e.g., generating a lot of spurious ambiguity and slowing down the parsing process, making it more difficult to treebank (see Chapter 7) and, most importantly for the context of this thesis, it may

predict ungrammatical sentences as valid.

As discussed in Chapter 3, one of the benefits of using HPSG for grammatical error detection is the fact that HPSG grammars make an implicit grammaticality judgment when they parse or reject an input. Maintaining a rigorously defined lexicon is part of this effort, and incorrect lexical entries can easily generate parses for ungrammatical input. Before extending ZHONG with mal-rules, it was important to revise the lexicon in an effort to eliminate incorrect parses and to reduce spurious ambiguity.

This task of lexical expansion and revision was done using a mix of dictionary resources (MDBG,[1] TrainChinese,[2] ZDIC,[3] Purple Culture,[4] iCIBA,[5] and iCHACHA[6]) and native speaker informants. Many incorrect entries were deleted or corrected, and many others were added.

Table 6.1 shows a summary of the number of lexical entries in ZHONG per file, comparing the first version (V1.0) and the version produced by this thesis (V2.0). ZHONG's lexicon is distributed across a few different files: `lex-base` contains mostly closed class words and a few other hand-checked entries used to produce ZHONG V1.0; `lex-symbols` includes mostly punctuation and other symbols like parenthesis, etc.; `lex-classifiers` includes only classifiers; `lex-numbers` includes a very long list of numbers (i.e., numbers are currently not being treated compositionally); `lexicon` contains a long list of automatically collected lexical entries (discussed in Section 4.2); `lex-core` is a new file created for V2.0 to host only hand-checked lexical entries; and `mal-lex` is a file containing lexical entries associated with error detection, discussed in greater detail in Section 6.6, below.

Even though Table 6.1 shows a positive net difference of only 694 lexical entries between both versions of the lexicon, this does not fully reflect the work that was dedicated to revise it. For example, despite the file `lexicon` showing that the V2.0 constains only 27 more lexical entries than V1.0, a fair number of lexical entries were also deleted or revised within this file. The net outcome of 27 new lexical entries in this file can be explained by the fact that many entries that were not considered fully revised were also added to this file. This included, for

[1] `https://www.mdbg.net`
[2] `https://www.trainchinese.com`
[3] `https://www.zdic.net/`
[4] `https://www.purpleculture.net/chinese-english-dictionary/`
[5] `http://www.iciba.com/`
[6] `https://eng.ichacha.net/`

File	ZHONG V1.0	ZHONG V2.0
lex-base	171	177
lex-symbols	59	61
lex-classifiers	465	456
lex-numbers	20,255	20,262
lexicon	17,301	17,328
lex-core	-	613
mal-lex	2	50
Total	38,253	38,947

Table 6.1: ZHONG Lexical Entries

example, many English proper nouns found in the development data, and that should not necessarily be included in the grammar (i.e., they might be handled by a preprocessing step in the future). When in doubt whether a lexical entry was in its final form, words were not added to the hand-curated lexicon (i.e., `lex-core`).

I estimate that around 1,500 lexical entries were either deleted, revised or added between both versions.

6.3 Separable Verbs

Separable verbs are a class of verbs in Mandarin Chinese with two morphemes that can appear discontinuously in certain syntactic contexts. They are also known as 'separable words', 'separable idioms' or 'compound verbs' (Li and Thompson, 1981; Wang and Müller, 2013; Petrovčič, 2016). This class of verbs includes entries like 洗澡 (*xǐzǎo*, to bathe), 吃饭 (*chīfàn*, to eat), 念书 (*niànshū*, to read/to study/to be in school), or 跳舞 (*tiàowǔ*, to dance). On the surface, these lexical entries look like regular verbs, but they differ from other (non-separable) verbs in the fact that they can often appear noncontiguously – see examples (36) and (37).

(36) 他 会 跳舞 。
tā huì tiàowǔ .
3SG.MASC can dance .
'He can dance.'

(37) 他 跳 了 两 个 小时 的 舞 。
Tā tiào le liǎng ge xiǎoshí de wǔ .
3SG.MASC dance-跳 ASP.le two CLF hours ATTRIB.de dance-舞 .
'He danced for 2 hours.'

The example sentence (37) shows the true morpho-syntactic nature of separable verbs. Despite the fact that 跳舞 (*tiàowǔ*, to dance) is often seen as a single word (e.g., listed in dictionaries and taught in classrooms as a single word), in this example we see that it does not behave as a simple verb. In addition, when comparing (37) to (38), we can see that these two sentences share a lot of their structure.

(38) 他 看 了 两 个 小时 的 电影 。
Tā kàn le liǎng ge xiǎoshí de diànyǐng .
3SG watch ASP.le two CLF hours ATTRIB.de movie .
'He watched a 2 hour movie.' or 'He watched a movie for 2 hours.'

The verb 跳舞 (*tiàowǔ*, to dance) seems to split itself into two syntactic positions – the verb and its complement – which is the relation between 看 (*kàn*, to watch/to see) and 电影 (*diànyǐng*, movie) in (38). However, if we took the literal meanings of 跳 (*tiào*) and 舞 (*wǔ*), it would be closest to *to jump a dance*. Petrovčič (2016) lists many more of these literal oddities such as *to tie* + *a marriage* → to marry, *to run* + *a step* → to run, or *to see* + *a face* → to meet (etc.). This lack of compositionality explains why these verbs or expressions are often analysed as idiomatic expressions, equivalent to the likes of *to kick the bucket* (meaning *to die*).

It is important to note that, similar to what happens with idiomatic expressions, there seems to be a good amount of variability within separable verbs – including opaqueness of meaning, boundedness of morphemes, availability of ambiguity, or syntactic freedom (e.g., modification, extraction, etc.). Wang and Müller (2013) note that, in general, separable verbs can be separated by different kinds of adjuncts, and usually interact well with passivization and extraction (e.g., BA-constructions) while preserving their idiomatic meaning. Topicalization, on the other hand, seems to be much more restricted, if at all possible. On the matter of what can come between the two morphemes, Petrovčič (2016) lists aspectual particles, various kinds of complements (directional, resultative, quantitative, potential) as well as attributive modification as

possible intervening elements. But even though many of these syntactic behaviors and intervening elements can happen in principle, it is also important to stress that the acceptability of these phenomena cannot simply be explained by a sub-classification of separable verbs. The acceptability of different kinds of intervening elements is also very much tied to speaker variability (i.e., the same interactions are judged differently by different native speakers).

In this thesis, I follow Li and Thompson (1981)'s definition of separable verbs, which requires at least one of three properties to be present: a) either one or both constituents are bound morphemes; b) there is idiomacy (or non-composionality) in the meaning of the entire unit; or 3) there is confirmed inseparability or limited separability of the constituents.

One of the main goals of adding separable verbs to ZHONG was to be able to provide comparable semantics to sentences like (36) and (37). The first version of ZHONG was not able to show that 跳 (*tiào*) and 舞 (*wǔ*) were related (i.e., should form a single semantic predicate) when they were not contiguous. When both morphemes of a separable verb were also free morphemes, as is the case for 跳 (*tiào*) and 舞 (*wǔ*), ZHONG was often able to able to produce a viable parse for the sentence (from a syntactic perspective). However, from a semantic perspective it missed the fact that the meaning of 跳 (*tiào*) and 舞 (*wǔ*) was not possible to be derived compositionally. For cases where at least one morpheme was bound, e.g., 洗澡 (*xǐzǎo*, to bathe), this often meant that ZHONG could only parse the simplest sentences – i.e., those without any intervening elements. Unfortunately, some intervening elements (e.g., aspect particles) are extremely common, and this was hurting ZHONG's parsing coverage.

The analysis I propose here is, in fact, very similar in spirit to what is suggested by Wang and Müller (2013), but simplified in some ways. One of the main differences is that I do not attempt to differentiate between partially literal and full opaque (i.e., non-compositional) semantics. While it is evident that there are different degrees of compositionality (Li and Thompson, 1981; Wang and Müller, 2013), I was not entirely convinced of the benefits of trying to model these different levels of compositionality at this stage of the grammar. The second main difference of my analysis is that I chose to over-constrain what kind of intervening elements can appear between separable verbs. As mentioned above, the variability of what can or not be an intervening element tends to be a very sensitive subject (i.e., different native speakers have very

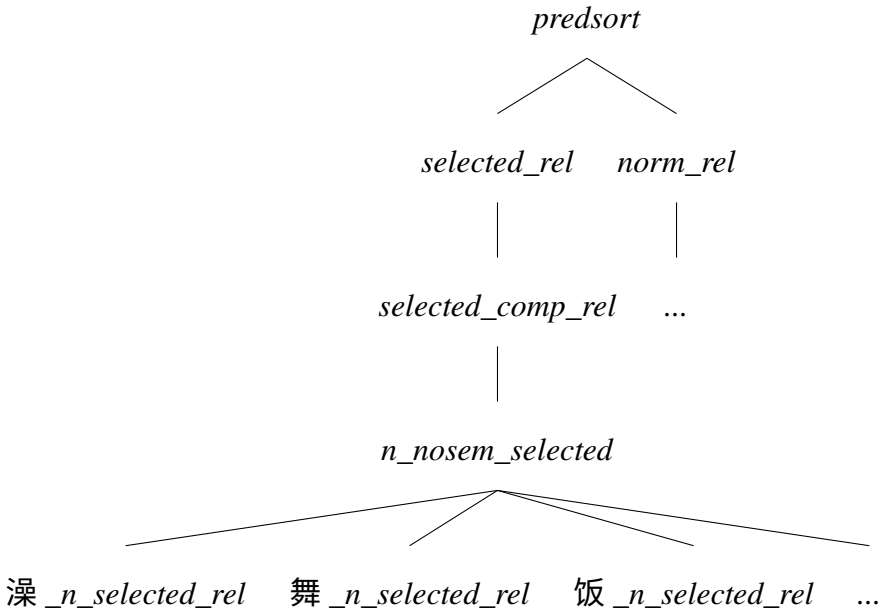
different intuitions about what is acceptable), and most of my informants were fairly conservative. The current implementation limits these intervening elements to those clearly undisputed among all speakers, and those most pervasive in the development data – i.e., currently, only aspect particles are allowed as intervening elements.

Admittedly, the current implementation of separable verbs still sidesteps much of the complexity described in the available literature. The goal of this implementation was not to have a complete coverage since this would require a much deeper analysis that would have been disproportionate in the context of this thesis. Instead, the current implementation was designed to capture the complexity early Mandarin learners are exposed to. This also solves some concerns regarding word segmentation (which will be discussed below), and enables ZHONG to provide consistent semantics for these verbs, regardless of whether or not they appear contiguously.

Similar to the analysis of Wang and Müller (2013), ZHONG's current implementation of separable verbs assumes that the first morpheme behaves as the verb (i.e., the head of the phrase), which is responsible for introducing the idiomatic predicate in the semantics. However, this verb selects for a special, semantically empty, non-optional complement. In other words, a semantically empty complement needs to exist in order for the idiomatic meaning to be available. The second morpheme of the separable verb is taken as a complement using the same rules as normal transitive verbs, upholding the parallelism discussed for examples (37) and (38).

The strict mutual selection between both morphemes of a separable verb is implemented through a hierarchy that sorts semantic predicates – exemplified in (39). Lexical types in ZHONG were essentially split in two: those that can be selected (i.e., `selected_rel`) and those that cannot (i.e., `norm_rel`). As can be expected, a large majority of lexical entries inherit from `norm_rel`, and currently only entries related to separable verbs inherit from `selected_rel`. Foreseeing that this distinction might also be useful in other parts of the grammar, `selected_rel` was specialized further. The first specialization is `selected_comp_rel` (defining lexical types that can be selected as complements of something). This type is then further specialized into `n_nosem_selected`, which defines semantically empty nouns that can be selected as complements – currently used only for separable verbs.

(39)



As can be seen in (39), under the type `n_nosem_selected` one can find types corresponding to the second morphemes (i.e., the complements) of separable verbs. For example, the type 澡`_n_selected_rel` corresponds to the second morpheme of 洗澡 (*xǐzǎo*, to bathe), 舞`_n_selected_rel` corresponds to the second morpheme of 跳舞 (*tiàowǔ*, to dance) and 饭`_n_selected_rel` corresponds to the second morpheme of 吃饭 (*chīfàn*, to eat).

In the lexicon, separable verbs require two entries, one for the first morpheme (i.e., the verb) that introduces the semantic predicate, and another for the semantically empty noun. Examples (40) and (41) show how these lexical entries look like for 洗澡 (*xǐzǎo*, to bathe).

(40)

```
洗_澡_sep_v_1 := v_np_sep_le &
  [ STEM < "洗" >,
    SYNSEM [ LOCAL.CAT.HEAD.CHAR [ FCHAR "洗", LENGTH one ],
             LKEYS [ --COMPKEY 澡_n_selected_rel,
                     KEYREL.PRED "_洗_v_澡_sep_1_rel" ] ] ].
```

(41)

```
澡_sep_n_1 := n_nosem_selected &
  [ STEM < "澡" >,
    SYNSEM [ LOCAL.CAT.HEAD.CHAR [ FCHAR "澡", LENGTH one ],
             LKEYS.KEYREL.PRED 澡_n_selected_rel ] ].
```

The first morpheme is the verb, inheriting from the type `v_np_sep_le` (where 'sep' stands for separable). This lexical entry has only the first morpheme listed in its `STEM` feature, which is what will be matched in the sentence's surface form. It is very much similar to other transitive verbs in the grammar, with the exception that it is able to select its complement, and that it provides a semantic predicate with only one argument (i.e., the subject), similar to intransitive verbs. Despite the fact that it syntactically behaves as a transitive verb, the complement is semantically empty – which is reflected in the semantics by the absence of an `ARG2`.

The selection of its special complement is done by the feature `--COMPKEY` which, for (40), is defined to be `澡_n_selected_rel`. This predicate can then be seen in (41) as the value for the feature `PRED` (i.e., for predicate). Even though (41) has a value for the feature `PRED`, this value does not get added to the semantic output of a parsed sentence. Only the first morpheme of separable verbs contributes its semantic predicate. The predicate of the second morpheme is used only for the selection process.

The final result of how these new lexical entries behave in a sentence can be seen in Figures 6.1 and 6.2. Figure 6.1 shows the parse tree and MRS output for the sentence '我洗澡。' (I bathe). In the syntactic tree, we can see that the lexical entries shown in (40) and (41) join through a `head-comp` (i.e., head-complement) rule. This is the same rule used to join regular transitive verbs with their complements. However, following what was discussed above, we can see that the semantic output has only the semantic predicate introduced by the first morpheme, (40), which in this case is `_洗_v_澡_sep_1_rel`.

Figure 6.2 shows the syntactic tree and the MRS for the sentence '我洗了澡。' (I bathed / I have finished bathing). The only difference between the two sentences is the intervening 了 (*le*) between the two morphemes – this particle marks the perfective aspect. And, as can be seen in

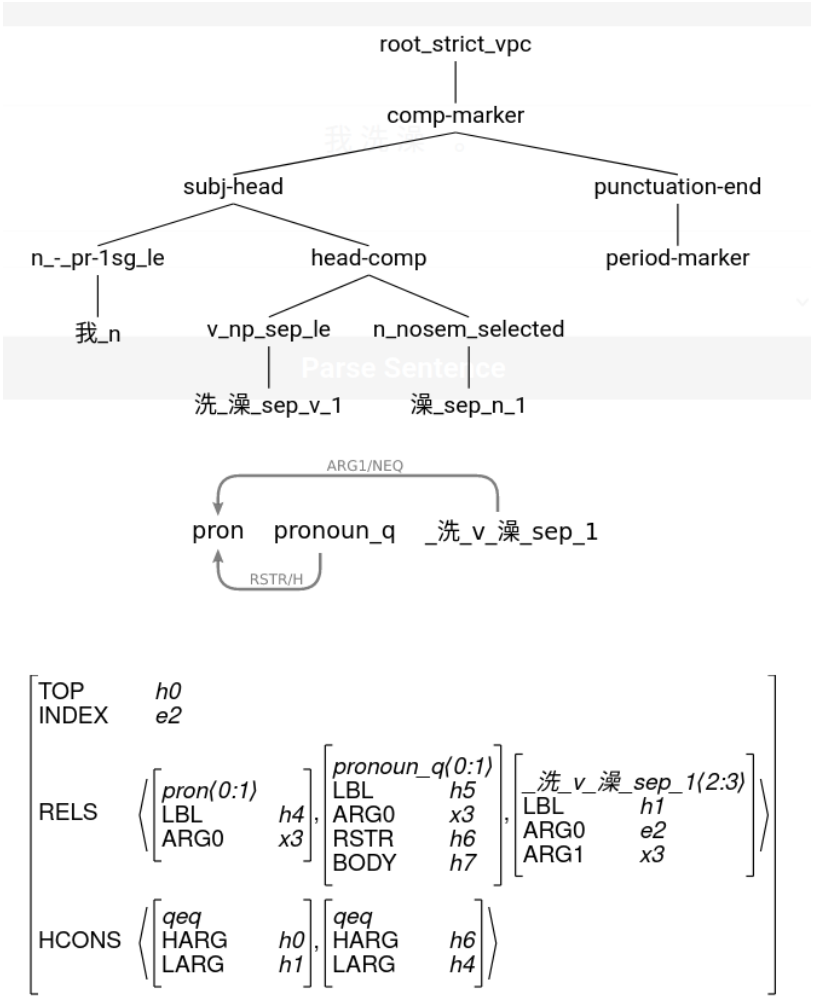


Figure 6.1: Syntactic and semantic outputs for the sentence: 我洗澡。(I bathe.)

Figure 6.2, the current implementation allows this sentence to parse without any problems. As intended, the semantic output is virtually the same to the one shown for Figure 6.1 – the only difference would be the aspect associated with the predicate `_洗_v_澡_sep_1_rel` (which is, unfortunately, not visible in this graphic depiction of MRS).

ZHONG currently has 38 separable verbs in its lexicon. This list is far from being exhaustive. These entries were selected because they were contained in the development data and, as pointed above, they are fairly restricted with regard to what kind of intervening elements are allowed. A future expansion of this list would invite further work into a possible clustering of separable verbs into subtypes – in particular, this would need to include an in-depth study of the kind of intervening elements that can appear between both morphemes, and whether sub-types of

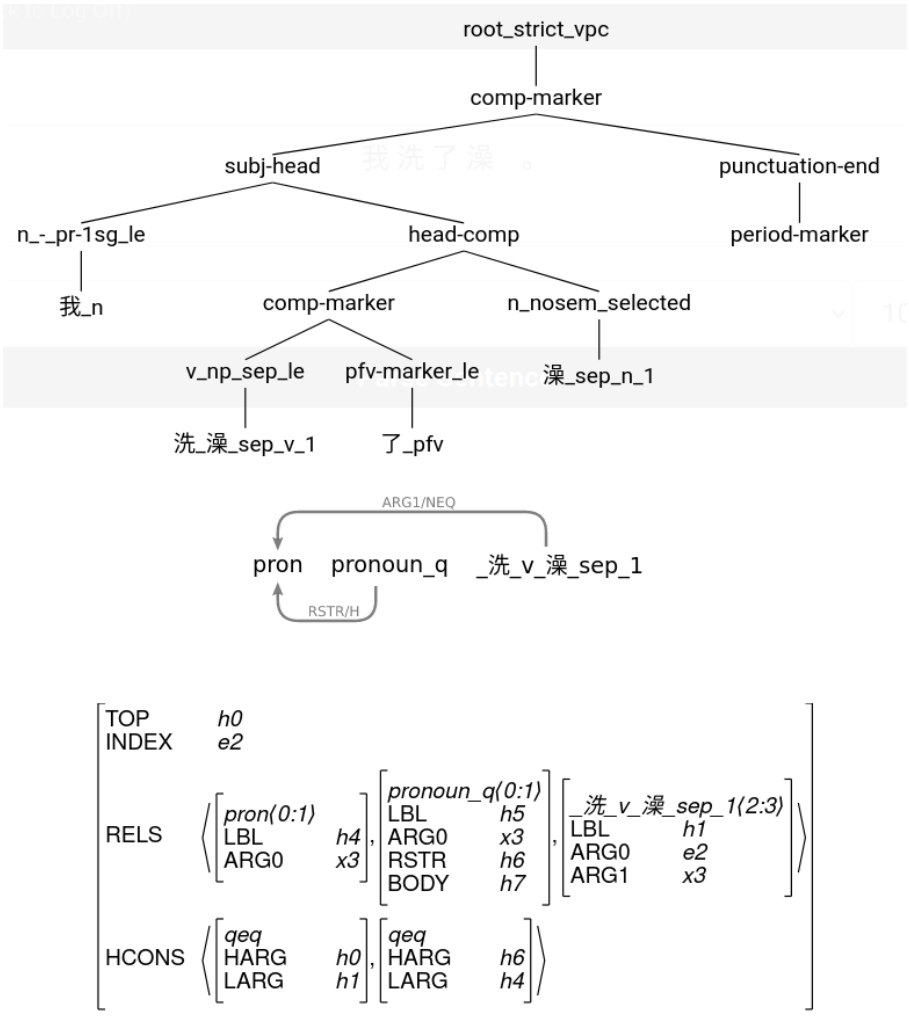


Figure 6.2: Syntactic and semantic outputs for the sentence: 我洗了澡。(I bathed.)

separable-verbs could be identified with regard to their syntactic behavior.

6.4 Interactions between Negation and Aspect

Another topic that received substantial attention during ZHONG's development was aspect – in particular, the interaction between aspect particles and negation.

Mandarin Chinese does not mark tense in verbs, as English and many other languages do. Instead, it uses a fairly rich number of verbal affixes (i.e., aspect particles) to mark relations between the time a situation occurs and the time it is brought to speech/writing (Li and Thompson, 1981). Li and Thompson (1981) identify four verbal aspects in Mandarin Chinese: perfective,

imperfective (durative), experiential, and delimitative. Out of these, the delimitative aspect does not use aspect particles, and instead is achieved by reduplication of the verb – this aspect will not be discussed further in this thesis. The other three verbal aspects are frequently associated with four particles (Li and Thompson, 1981): the perfective marker 了 (*le*), the experiential marker 过 (*guò*), and the imperfective (durative) markers 着 (*zhe*) and 在 (*zài*).

At the same time, Mandarin Chinese also has multiple negators that interact with aspect particles. Li and Thompson (1981) identify three main negative forms in Mandarin Chinese:[7] 不 (*bù*, no), 没 (*méi*, no) and 别 (*bié*, do not). Out of these, 别 (*bié*, do not) is used strictly for imperative constructions (Li and Thompson, 1981). Due to its very limited context, it is not usually discussed in relation to aspect. As such, 别 (*bié*, do not) will not be considered in this discussion.

Linguistic discussions concerning aspect (Li and Thompson, 1981; Ernst, 1995; Peck et al., 2013) and its interaction with negation (Teng, 1973; Lin, 2003; Xiao and McEnery, 2008) are long and largely unfinished. Fuller of accounts attempting to describe the behavior of aspect in Mandarin Chinese use semantic features such as dynamicity, duration, boundedness, telicity, and more to provide highly lexicalized descriptions of verb classes (Peck et al., 2013). These works attempt to draw distinctions between the way different verb classes interact with different aspect particles based on their internal semantic structure, and most often include also negators such as 不 (*bù*, no) and 没 (*méi*, no) in their discussion.

Unfortunately, similarly to what has been discussed for separable verbs, it would be impossible to provide a full discussion concerning the topics of verbal aspect and negation in Mandarin Chinese within the scope of this thesis. Both these topics are extremely broad, and have interactions across many other phenomena. A proper discussion would require its own thesis, and a definite resolution would still not be guaranteed.

My contributions to these topics are extremely focused, and attempt to draw broad generalizations from the interactions between aspect particles and negators themselves. Saving a couple of notable exceptions, motivated by frequency and by common mistakes associated with them, the work presented here does not attempt to classify verbs according to 'aspectual classes' (see:

[7] Sometimes 没 (*méi*, no) and 没 (有) (*méi(yǒu)*, no) are considered different forms, but they will be considered as a single form for the purpose of this discussion.

Peck et al., 2013). Instead, the goal of this section is to correctly constrain ZHONG regarding a few special interactions (i.e., co-occurrence) of particular aspect particles and particular negators – which hold true regardless of the verb class they occur with.

The solution proposed here is sufficient to solve some of the basic problems for which ZHONG was still unable to provide an adequate solution. It is a solution only to the level of complexity that it targeted, and it will most certainly require further fine-tuning as ZHONG continues to be developed.

Some of the problems raised here come from the fact that aspect is a very difficult topic to introduce to students. As a consequence, teachers and students alike often draw parallels between aspect particles and tense (e.g., erroneously equating the perfective marker 了 (*le*) to past tense). These parallels are not only incorrect by themselves, but also extend into further incorrect assumptions about the interaction between aspect particles and negators.

One central fact about this interaction is that both 不 (*bù*, no) and 没 (*méi*, no) are incompatible with the perfective marker 了 (*le*). See (42) to (46), below.

(42) 我 吃 了 猪肉 。
Wǒ chī le zhūròu .
1SG eat ASP.le pork .
'I ate pork.'

(43) 我 不 吃 猪肉 。
Wǒ bù chī zhūròu .
1SG NEG.bu eat pork .
'I don't eat pork.'

(44) * 我 不 吃 了 猪肉 。
Wǒ bù chī le zhūròu .
1SG NEG.bu eat ASP.le pork .
'I didn't eat pork.' (? intended)

(45) 我 没 吃 猪肉 。

Wǒ méi chī zhūròu .
1SG NEG.mei eat pork .

'I didn't eat pork.'

(46) * 我 没 吃 了 猪肉 。

Wǒ méi chī ASP.le zhūròu .
1SG NEG.mei eat ASP.le pork .

'I didn't eat pork.' (? intended)

The interactions between 不 (*bù*, no), 没 (*méi*, no) and 了 (*le*) shown above are agnostic to the type verb that is used. Except, of course, for verbs that are not compatible with the perfective aspect to start with, and hence would disallow the use of 了 (*le*). However, assuming a verb is compatible with the perfective aspect, it is a fact that 不 (*bù*, no) and 没 (*méi*, no) are always incompatible with 了 (*le*). The reason for this is that both 不 (*bù*, no) and 没 (*méi*, no) have inherent aspectual requirements that are incompatible with those inherent to 了 (*le*) (Teng, 1973; Ernst, 1995; Lin, 2003; Xiao and McEnery, 2008).

The perfective aspect introduced by 了 (*le*) denotes the completion of an action as a whole. The negator 不 (*bù*, no), however, cannot be associated with the perfective aspect (or telic situations, Ernst, 1995), and it is often characterized as negating the existence of a state. This explains why example (44) is nonsensical/ungrammatical.

The negator 没 (*méi*, no), on the other hand, is said to denote the semantic opposite of 了 (*le*) (Lin, 2003) – putting them in a complementary distribution. Concerning which aspectual label this would correspond to, Ernst (1995) labels a sentence like (45) as not-PRF (i.e., negative perfective), denoting that the completion of the action has not been achieved.

There are many other similar restrictions. The experiential aspect marker 过 (*guò*), for example, can only be negated using 没 (*méi*, no) – and is incompatible with 不 (*bù*, no). However, contrary to what happens for the perfective aspect, where 没 (*méi*, no) and 了 (*le*) cannot co-occur in the same sentence, this is what must happen if one wants to negate the experiential aspect. See examples (47) through (49).

(47) 我 吃 过 猪肉 。

Wǒ chī ASP.guo zhūròu .
1SG eat ASP.guo pork .

'I have eaten pork.'

(48) 我 没 吃 过 猪肉 。

Wǒ méi chī ASP.guo zhūròu .
1SG NEG.mei eat ASP.guo pork .

'I haven't eaten pork.'

(49) * 我 不 吃 过 猪肉 。

Wǒ bù chī ASP.guo zhūròu .
1SG NEG.bu eat ASP.guo pork .

'I haven't eaten pork.' (intended)

Similar interactions also happen with the imperfective (durative) markers 着 (*zhe*) and 在 (*zài*). These two markers behave similarly to the experiential aspect marker 过 (*guò*) insofar that they can only be negated by 没 (*méi*, no), and that the negation is achieved by including both the aspect particle and the negator in the same sentence.

Other types of selection that will only be briefly discussed here concern specific types of predicate. For example, according to Ross and Ma (2017), the large majority of adjectives and modal verbs – e.g., 会 (*huì*, to can), 可以 (*kěyǐ*, to be able to/allowed to), 能 (*néng*, to be capable of/possible to) – can only be negated with 不 (*bù*, no). Similarly, Ross and Ma (2017) also note that most stative verbs – e.g., 喜欢 (*xǐhuān*, to like), 爱 (*ài*, to love), 想 (*xiǎng*, to want), 怕 (*pà*, to fear), or 懂 (*dǒng*, to understand) – are more naturally negated by 不 (*bù*, no) – however corpus data clearly suggests that this second statement is not a strict rule.

As mentioned above, a broad characterization of verbs by aspectual classes is outside the scope of this thesis. However, the copula verb 是 (*shì*, to be) and the special stative verb 有 (*yǒu*, to have) are two notable exceptions. These two verbs not only show interesting aspect/negation constraints, but are also extremely frequent (both within and outside the development data).

And of particular interest for this thesis is also the fact that the aspect/negation constraints of 有 (*yǒu*, to have) is the source of a fairly common grammatical error, shown in (57), below.

The first of these two verbs is the copula verb 是 (*shì*, to be). This verb is incompatible with all four aspect particles, and also incompatible with 没 (*méi*, no). As such, it can only be negated using 不 (*bù*, no) – as can be seen in examples (50) trough (54).

(50) 迈克 是 美国人 。

màikè shì měiguórén .
Mike COP American .

'Mike is American.'

(51) 迈克 不 是 美国人 。

màikè bú shì měiguórén .
Mike NEG.bu COP American .

'Mike is not American.'

(52) * 迈克 没 是 美国人 。

màikè méi shì měiguórén .
Mike NEG.mei COP American .

'Mike was not American.' (?)

(53) * 迈克 是 了 美国人 。

màikè shì le měiguórén .
Mike COP ASP.le American .

'Mike was American.' (?)

(54) 去年 迈克 不 是 美国人 但是 今年 是 美国人。

qùnián màikè bú shì měiguórén dànshì jīnnián shì měiguórén
Last.year Mike NEG.bu COP American but this.year COP American

'Last year Mike was not American but this year he is American.'

The verb 有 (yǒu, to have), on the other hand, differs from the majority of other stative verbs (Ross and Ma, 2017) in the fact that it can only be negated by 没 (*méi*, no). In addition, 没 (*méi*, to not have) can also be used by itself, carrying the meaning of a negated 有 (*yǒu*, to have). These interactions are shown in examples (55) through (58).

(55) 我 有 钱 。
wǒ yǒu qián .
1SG have money .
'I have money.'

(56) 我 没 有 钱 。
wǒ méi yǒu qián .
1SG NEG.mei have money .
'I don't have money.'

(57) * 我 不 有 钱 。
wǒ bù yǒu qián .
1SG NEG.bu have money .
'I don't have money.' (intended)

(58) 我 没 钱 。
wǒ méi qián .
1SG NEG.mei money .
'I don't have money.'

The chosen method to deal with the issues discussed above, without necessarily having to do a full characterization of verbs by aspectual classes, was to encode the syntactic incompatibilities between aspect particles and negators in a new (i.e., improved) aspect hierarchy. ZHONG's original hierarchy dealt only with three aspect particles: the perfective marker 了 (*le*), the imperfective marker 着 (*zhe*), and the experiential marker 过 (*guò*) – leaving out the imperfective particle 在 (*zài*), 不 (*bù*, no) and 没 (*méi*, no). This hierarchy predicted incorrect parses, as it was not able to adequately reject many of the negative examples shown above (especially

those concerning negation). The new hierarchy, shown in in Figure 6.3, encodes the syntactic compatibility across all six function words.

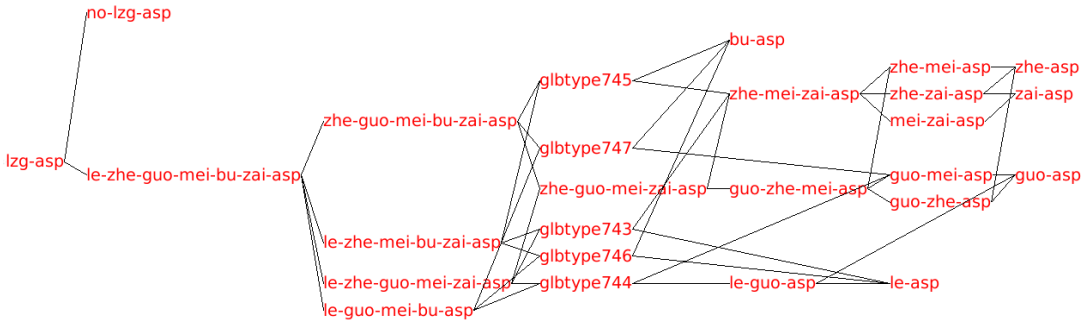


Figure 6.3: New Aspect Hierarchy (as produced by the LKB-FOS)

This hierarchy is used by verbs to restrict their individual compatibility with certain aspect/negators, but also encodes important constraints between these function words. The default for all verbs is `lzg-asp`, the top node of the hierarchy, which will allow the further subcategorization of verb classes by aspectual classes in the future. However, this hierarchy is also used by aspectual particles and by negators. On the verb level, certain verb classes (or, currently, specific verbs) can restrict the kind of particles that can be used with themselves by choosing an adequate value for the feature `LZG` (this stands for LE-ZHE-GUO, and was the feature name used in ZHONG's original aspect hierarchy).

For example, the value `le-zhe-guo-mei-bu-zai-asp` can be used for verbs that interact with all six function words. And the value `no-lzg-asp` can be used for verbs that cannot interact with any of these six function words (which is expected to be a fairly rare occurrence).

Aspectual particles and negators, on the other hand, use this hierarchy in two ways: i) to verify they are compatible with a verb (i.e., incompatibilities result in rejected parses); ii) to further constrain the value of the `LZG` feature of verbs to define what particles and negators can co-occur with the function words already present in a sentence.

A simple example of how this hierarchy works can be demonstrated using the discussion provided for examples (50) through (54), above. As was stated earlier, the copula verb 是 (*shì*, to be) is incompatible with all aspect particles, and it is also incompatible with 没 (*méi*, no). Out of the six function words represented by the new hierarchy, 是 (*shì*, to be) is only compatible with 不 (*bù*, no).

The example (59) shows the lexical entry for 是 (*shì*, to be). In this lexical entry, the feature LZG is set to bu-asp. Carefully inspecting the hierarchy shown in Figure 6.3, one can see that bu-asp is a leaf node – which effectively means that it will not be compatible with any other particle (except itself).

(59)

```
是_v_cop_1 := v_np_cop_le &
  [ STEM < "是" >,
    SYNSEM [ LOCAL.CAT.HEAD [ CHAR [ FCHAR "是", LENGTH one ],
                              LZG bu-asp ],
             LKEYS.KEYREL.PRED "是_v_cop_rel" ] ].
```

Let us now consider the lexical entry for 不 (*bù*, no), shown in (60). 不 (*bù*, no) constrains the LZG of its MOD to bu-asp. Both 不 (*bù*, no) and 没 (*méi*, no) attach to verbs as modifiers (i.e., adverbs). The content of the feature MOD defines what they can modify. In this case, 不 (*bù*, no) can modify verbs[8] with a LZG feature compatible with bu-asp. This is done through unification of types, as introduced in Chapter 3.

(60)

```
不_r := adv_-_neg_le &
  [ STEM < "不" >,
    SYNSEM [ LOCAL.CAT.HEAD [ CHAR [ FCHAR "不", LENGTH one ],
                              MOD < [ CAT.HEAD.LZG bu-asp ] > ]
             LKEYS.KEYREL.PRED _bu_x_rel ] ].
```

Since the copula verb 是 (*shì*, to be) has its LZG feature set to bu-asp, this means that it will be compatible with (i.e., can be modified by) 不 (*bù*, no). The unification of two equal types is the type itself.

In comparison, considering the lexical entry for the perfective particle 了 (*le*), shown in (61), the results of the unification would fail.

[8] In fact it can also modify adjectives, but this will be excluded from the discussion for simplicity.

(61)

```
了_pfv := pfv-marker_le &
  [ STEM <"了">,
    SYNSEM.LOCAL.CAT [ HEAD.CHAR [ FCHAR "了", LENGTH one ],
                       VAL.COMPS < [ LOCAL.CAT.HEAD.LZG le-asp] > ] ].
```

了 (*le*) sets the LZG feature of its complement to le-asp. Contrary to 不 (*bù*, no) and 没 (*méi*, no), aspect particles attach to verbs using a special comp-marker rule, where the verb is the complement of the aspect particle while remaining the head of the phrase. In the lexical entry shown in (61), one can see that 了 (*le*) forces its complement to have an LZG compatible with le-asp. However, when inspecting Figure 6.3, it should be clear that there is no type that is compatible with both le-asp and bu-asp. In other words, this means that the verb 是 (*shì*, to be) is not compatible with the aspect particle 了 (*le*) – any sentence that attempts this will be rejected. This is a desirable behavior, as shown in (53).

Using the same line of reasoning, it should also be clear that any verb modified with 不 (*bù*, no), which will have their LZG feature constrained to bu-asp, would also be incompatible with 了 (*le*) – which is a desirable outcome, as shown in (44). In fact, (*bù*, no) is incompatible with all aspect particles. From a syntactic perspective, aspect particles attach to the verb before negation, so it might be best to say that no aspect particle can interact with (*bù*, no).

Following the discussion presented for examples (55) through (58), one can see that the verb 有 (yǒu, to have) also has some interesting aspect/negation constraints. Considering the lexical entry for the verb 有 (yǒu, to have) provided in (62), with the lexical entry for 不 (*bù*, no), provided above, it should now be clear that the new hierarchy does not allow 不 (*bù*, no) to modify the verb 有 (yǒu, to have).

(62)

```
有_v_1 := v_np_le &
  [ STEM < "有" >,
    SYNSEM [ LOCAL.CAT.HEAD [ CHAR [ FCHAR "有", LENGTH one ],
                              LZG le-zhe-guo-mei-zai-asp ],
             LKEYS.KEYREL.PRED "_有_v_1_rel" ] ].
```

The verb 有 (yǒu, to have) sets its own LZG feature to le-zhe-guo-mei-zai-asp. This type is compatible with all aspect particles and with 没 (*méi*, no), but is not compatible with 不 (*bù*, no) – which is a welcome restriction, as shown in (57). There is no common type inheriting from both le-zhe-guo-mei-zai-asp and bu-asp – which would be the requirment imposed by 不 (*bù*, no) in order to be able to modify this verb.

These restriction are less strict for 没 (*méi*, no) – whose lexical entry shown in (63).

(63)

```
没_r := adv_-_neg_le &
  [ STEM < "没" >,
    SYNSEM [ LOCAL.CAT.HEAD [ CHAR [ FCHAR "没", LENGTH one ],
                              MOD < [ LOCAL.CAT.HEAD.LZG guo-zhe-asp ] > ],
             LKEYS.KEYREL.PRED _mei_x_rel ] ].
```

The lexical entry for 没 (*méi*, no) shows that it requires the verb it modifies to have a LZG value compatible with guo-zhe-asp. This value is incompatible with the value provided for a verb that is already marked with 了 (*le*) – which is desirable, as shown in (46). But it is compatible with the value provided for the verb 有 (yǒu, to have). The unification of the LZG values of 有 (yǒu, to have) and 没 (*méi*, no) is guo-zhe-asp.

This type, however, predicts that a verb that is modified by 没 (*méi*, no) can still take either the experiential marker 过 (*guò*) or the imperfective (durative) marker 着 (*zhe*) – which is the desired behavior, as shown in (48). In fact, 没 (*méi*, no) can also co-occur with the imperfect aspect particle 在 (*zài*) because this aspect particle constrains the LZG feature of the verb to

`zhe-mei-asp`, and the unification of `zhe-mei-asp` and `guo-zhe-asp` would be `zhe-asp`. This, in turn, would predict that a verb could be modified by 没 (*méi*, no) while taking both 在 (*zài*) and 着 (*zhe*) as aspect particles – which is, in fact, acceptable and shown in (64), with its syntatic and semantic representation shown in Figure 6.4.

(64) 迈克 没 在 看 着 她 。
màikè méi zài kàn zhe tā .
Mike NEG.mei ASP.zai see ASP.zhe her .
'Mike was not looking/staring at her.'

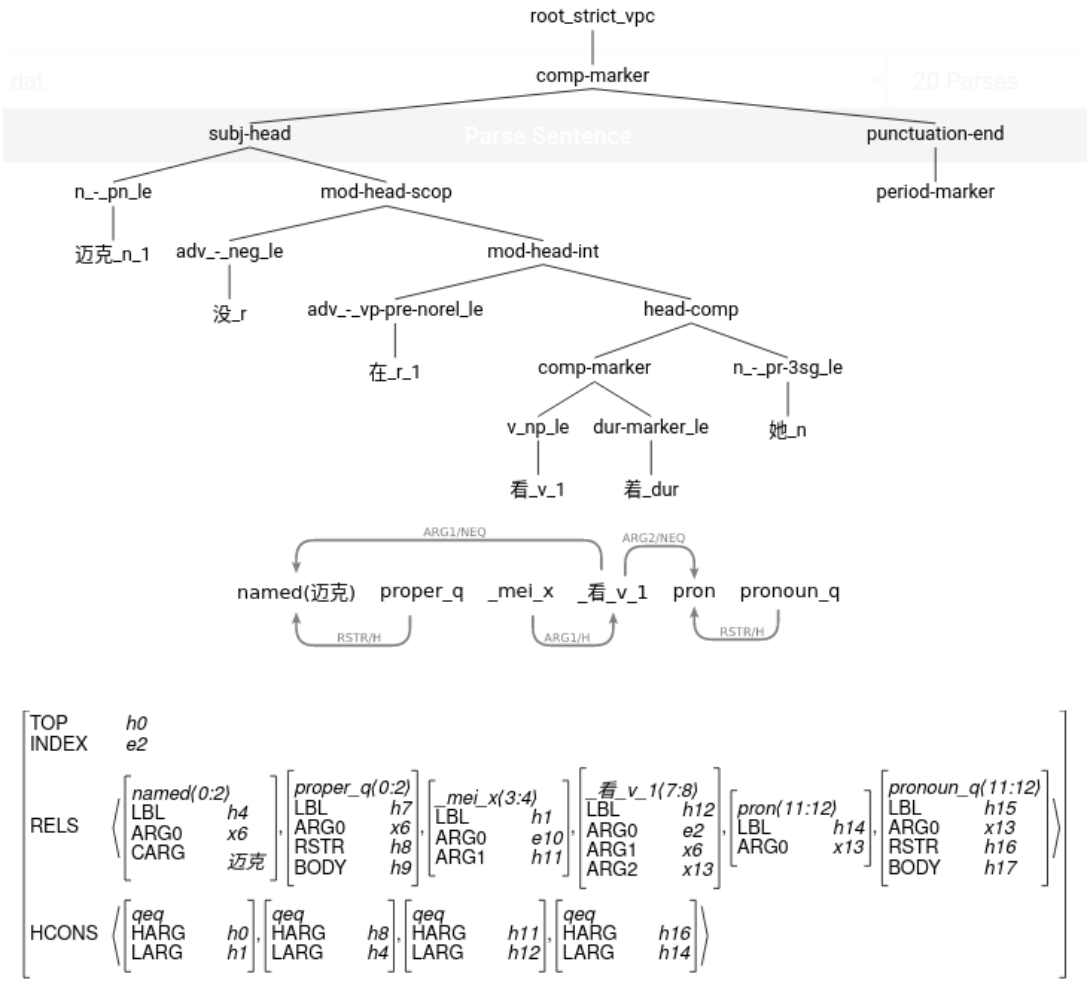


Figure 6.4: Syntactic and semantic outputs for the sentence: 迈克没在看着她 (Mike was not looking/staring at her.)

From the semantic output shown in Figure 6.4, and from the example lexical entries shown in (60), (61) and (63), it is possible to see that while negators introduce a predicate in the semantics, the aspect particles do not. Instead, they contribute to a semantic feature in the verbal

predicate. The nature of these aspectual semantic features will not be discussed here, as they were left mostly untouched (and incomplete), as released in ZHONG's first version.

It would be impossible to go over every interaction encoded in the new aspect hierarchy. What is most important to note is that the current hierarchy expanded ZHONG's ability to deal with a very complex matter. The hierarchy included in this version of ZHONG is largely compatible with the previous hierarchy, and all verbs that had specific aspect requirements encoded in their lexical entries (i.e., less than 2%) had equivalent restrictions from the new hierarchy assigned semi-automatically.

As previously stated, the solution presented here did not attempt to go into the classification of verbs by linguistically relevant features. Its main goal was to provide linguistic valid constraints and eliminate incorrect parses based mostly on the relation between negators and aspect particles. The strictness of this new hierarchy enables ZHONG to be on more solid ground to perform error detection of problems concerning these aspect/negation interactions – which will be discussed later in this chapter.

While a in-depth study of verbal aspect in Mandarin Chinese would be necessary to further progress this section of the grammar (and most probably requiring further tuning of the newly proposed aspectual hierarchy), it was also clear that this new hierarchy provides sufficient explanatory power to start a classification of verbs by their aspect compatibility. This is a promising direction for future work.

6.5 Other Extensions

It is impossible to describe in full detail all improvements made to ZHONG during this thesis. In this section I will enumerate a few other topics that received noteworthy contributions during this thesis without providing details of their implementation.

Regular Expression Pre-Processing

In order to generate a word-segmented sentence necessary to start the linguistic analysis, ZHONG currently relies on the Stanford's Word Segmenter for Chinese using the Chinese Penn Treebank

model (Tseng et al., 2005; Chang et al., 2008). However, the segmentation produced by these tools is not perfect. On the one hand, the Chinese Penn Treebank adopts a segmentation schema that is not consistent with some of the grammar's assumptions. This includes, for example, segmenting determiners and classifiers as a single token – e.g., 这个 (*zhège*, this/this one) instead of 这 (*zhè*, this) + 个 (*ge*, generic classifier). The Chinese Penn Treebank treats these two characters as a single token, while ZHONG expects these two characters to be analyzed as separate tokens. On the other hand, any strict segmentation schema ignores an interesting problem of Chinese syntax known as morphological ambiguity (Chen and Liu, 1992; Chang and Krulee, 1991). A very popular example of this ambiguity is the word 炒饭 (*chǎofàn*, fried rice), which can be single noun, but that can also be analyzed as two separate words 炒 (*chǎo*, to fry) and 饭 (*fàn*, rice), to produce a verb phrase – generating the ambiguity shown in examples (65) and (66).

(65) 迈克 要 炒饭 。
màikè yào chǎofàn .
Mike want fried-rice .
'Mike wants fried rice.'

(66) 迈克 要 炒 饭 。
màikè yào chǎo fàn .
Mike want fry rice .
'Mike wants to fry rice.'

In an attempt to solve many of these problems related to segmentation, DELPH-IN grammars have a preprocessing pipeline that includes a regular expression module (REPP, Dridan and Oepen, 2012). This module enables grammarians to preprocess the input to fit the grammar's theoretical assumptions. I have used this module extensively to improve ZHONG's ability to cope with incorrect word segmentation produced by the Stanford's Word Segmenter, and to keep as much morphological ambiguity as possible.

ZHONG's first version had only 17 such rules. The latest version has 205 rules. The large majority of these rules focus on the better segmentation of phenomena including: determiners,

numbers and classifiers; verbal and adjectival reduplication; prepositions and locative nouns; separable verbs; negation markers; auxiliary verbs and resultative constructions; aspect particles and other important function words.

Locative Nouns

ZHONG also received considerable improvements in the treatment of locative nouns. Mandarin Chinese has two main types of locatives: simple and synthetic (Li, 2013) – each with slightly different syntactic behaviors.

Simple locatives include words such as 东 (*dōng*, east), 西 (*xī*, west), 南 (*nán*, south), 北 (*běi*, north), 前 (*qián*, front), 后 (*hòu*, behind), 左 (*zuǒ*, left), 右 (*yòu*, right), 上 (*shàng*, above), 下 (*xià*, below), 里 (*lǐ*, inside) and 外 (*wài*, outside).

Synthetic locatives, are formed from by the juxtaposition of simple locatives with suffixes like 边 (*biān*, side) 面 (*miàn*, surface) – e.g., 前面 (*qiánmiàn*, front) 上边 (*shàngbian*, above), 里面 (*lǐmiàn*, inside), (etc.).

Simple locatives are treated as bound relational locative nouns that expect a non-optional nominal complement – see examples (67) and (68). Synthetic locatives, on the other hand, are unbound and can take an optional nominal complement – see examples (69) and (70).

(67) * 她 在 里 学习 。
tā zài lǐ xuéxí .
she PREP.in inside study .
'She studies inside.' (?)

(68) 她 在 房间 里 学习 。
tā zài fángjiān lǐ xuéxí .
she PREP.in room inside study .
'She studies inside the room.'

(69) 她 在 里面 学习 。

tā zài lǐmiàn xuéxí .
she PREP.in inside study .

'She studies inside.' (e.g., as opposed to outdoors)

(70) 她 在 房间 里面 学习 。

tā zài fángjiān lǐmiàn xuéxí .
she PREP.in room inside study .

'She studies inside the room.'

ZHONG's first version did not have a very thorough treatment of locatives. Both simple and synthetic locatives were analyzed as postpositions (as a consequence of the automated lexical acquisition). This is, however, not an acceptable analysis since locatives in Mandarin Chinese behave similarly to nouns in a variety of important contexts – e.g., they can be the subject of clauses, can head attributive 的 (*de*) constructions (which can only be headed by nouns), and can be the complement of verbs taking nominal complements – see examples (71) through (73). By treating locatives as postpositions instead of nouns, ZHONG was unable to parse a fair number of sentences.

(71) 里面 很 大 。

lǐmiàn hěn dà .
inside very big .

'The inside is very big.'

(72) 饭馆 的 里面 非常 漂亮 。

fànguǎn de lǐmiàn fēicháng piàoliang .
restaurant ATTR.de inside very beautiful .

'The inside of the restaurant is very beautiful.'

(73) 他 去 了 里面 。

Tā qù le lǐmiàn .
3SG go ASP.le inside .

'He went inside.'

The current analysis of locatives and all its interactions is far from complete, but this preliminary work has already helped boost the parsing coverage. In addition, the new analysis provides compatible semantic analyses for simple and synthetic locatives.

Numeric predicates

Numeric predicates were also missing from ZHONG's first version. In general, Mandarin Chinese has two main types of predicates – verbal and adjectival. However, in certain contexts, classifier phrases specified by a number can also serve as predicates – see examples (74) and (75).

(74) 这 本 书 二十 块 。
zhè běn shū èrshí kuài .
this CLS.books book twenty CLS.money .
'This book is twenty dollars.'

(75) 他 三十 岁 。
Tā sānshí suì .
3SG thirty CLS.age .
'He is thirty years old.'

Sentences using numeric predicates were extremely frequent in the development data and hence deserved some attention. Numeric predicates are being handled by pumping rules, which essentially transform classifiers phrases into verb phrases capable of taking a subject. In the semantics, the relation is captured by a copula predicate inserted by the pumping rule.

Reducing Spurious Ambiguity

Finally, my contributions to ZHONG also included some attention to spurious ambiguity. Ambiguity can be both good and bad. On the favorable side, there are truly ambiguous sentences (i.e., sentences expected to produce multiple different semantic representations). However, mostly due to the way the formalism is implemented, something called spurious ambiguity also exists. Spurious ambiguity is generated by artifacts in the implementation. Sometimes, these artifacts

can be considered 'bugs', but they are also often part of a conscious design choice – accepting the cost of ambiguity for a simpler/more elegant model.

I spent a non-trivial amount of time reducing spurious (nonessential) ambiguity in ZHONG. These improvements were fairly broad and are hard to pinpoint. However, improvements in the lexicon, modal verbs, question formation and attachment of negation are specially noteworthy.

6.6 Mal-Rules in ZHONG

In this section I will go over a non-exhaustive list of mal-rules added to ZHONG. These mal-rules were selected from the analysis of the NTU Corpus of Learner Mandarin (NTUCLM), presented in the previous chapter. Errors will be discussed in reference to Table 5.3. The example sentences used to illustrate the errors discussed here were also extracted from the same corpus.

An in-depth discussion of each detectable error and their respective mal-rule implementation would be extremely time consuming, and would not be necessarily interesting to the reader since many mal-rules share similar structures. Instead, the discussion will focus on a select number of errors that will serve as examples for how mal-rules were implemented in ZHONG. This discussion will also focus mainly on the syntactic aspects of mal-rule design, leaving the reconstruction of semantics largely implicit.

吗 (*ma*, question particle) redundancy

The first error to be discussed here is the use of the question particle 吗 (*ma*) in contexts where it is redundant (i.e., ungrammatical). This error corresponds to error ID 1 in Table 5.3. This was the most frequent error found in the NTUCLM. Minimal pairs exemplifying this error can be found in examples (76) through (79).

(76) 你 要 什么 ？
Nǐ yào shénme ?
2SG want QUEST.what ?
'What do you want?'

(77) * 你 要 什么 吗 ?

Nǐ yào shénme ma ?

2SG want QUEST.what QUEST.PART ?

'What do you want?' (intended)

(78) 你 有 没 有 中文 书 ?

Nǐ yǒu méi yǒu zhōngwén shū ?

2SG have NEG.mei have Chinese.language book QUEST.PART

'Do you have a Chinese textbook?'

(79) * 你 有 没 有 中文 书 吗 ?

Nǐ yǒu méi yǒu zhōngwén shū ma ?

2SG have NEG.mei have Chinese.language book QUEST.PART ?

'Do you have a Chinese textbook?' (intended)

Mandarin Chinese uses the question particle 吗 (*ma*) to transform propositions into polar (i.e., yes-no) questions. This particle, usually appearing without any other sort of syntactic evidence, often confuses L2 learners into assuming that it works similarly to a question mark (i.e., marking the existence of a question) – which is what happens with the particle か (*ka*) in Japanese. Unfortunately, as it can be seen in (77) and (79), this is not the case. In sentences where other question words are used, such as (76), the question particle 吗 (*ma*) should not be added.[9] A similar situation happens in (78), where the usage of a special syntactic construction with the form A-NOT-A (Wang et al., 2015b) already implies a polar question. For this reason, it would be redundant and ungrammatical to add the question particle 吗 (*ma*), as seen in (79).

The way ZHONG currently deals with this very common error is to use a special mal lexical entry for 吗 (*ma*), schematically equivalent to the one provided in (80). This mal lexical entry proposes an extra (i.e., a second) entry for 吗 (*ma*). Similar to the original entry for 吗 (*ma*), this mal sentence marker, modifies full sentences in a post-head position. This is shown in (80) by the feature $\left[\text{POSTHEAD } +\right]$. Similar to aspect markers, discussed in Section 6.4, sentence particles

[9]Here, I am assuming 什么 (shénme) is used as the interrogative pronoun *what*, and not as a plain pronoun meaning *something*.

attach to sentences using a special comp-marker rule, where the sentence is the complement of the marker while remaining the head of the phrase. The mal lexical entry shown in (80) can be seen selecting a sentence as its complement by the head feature (*verb*) with empty values for SUBJ and COMPS – i.e., a verb that has satisfied its COMPS and SUBJ requirements is one of the definitions of a sentence. Lastly, and most importantly, the complement of this mal lexical entry requires its complement to be already be interpreted as a question – shown here as SEM|MODE set to *quest* (although successfully defining this is actually a bit more complex within the grammar).

(80)

$$\left\langle \text{吗}, \left[\begin{array}{l} \textit{mal\_extra\_}\ \text{吗}\ \textit{\_qp} \\ \text{SYN} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \textit{marker} \\ \text{POSTHEAD}\ + \end{array} \right] \\ \text{VAL} \left[\begin{array}{ll} \text{SPR} & \langle\rangle \\ \text{COMPS} & \left\langle \left[\begin{array}{ll} \textit{verb} & \\ \text{SYN|VAL} & \left[\begin{array}{ll} \text{SUBJ} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{array} \right] \\ \text{SEM|MODE} & \textit{quest} \end{array} \right] \right\rangle \end{array} \right] \end{array} \right] \end{array} \right] \right\rangle$$

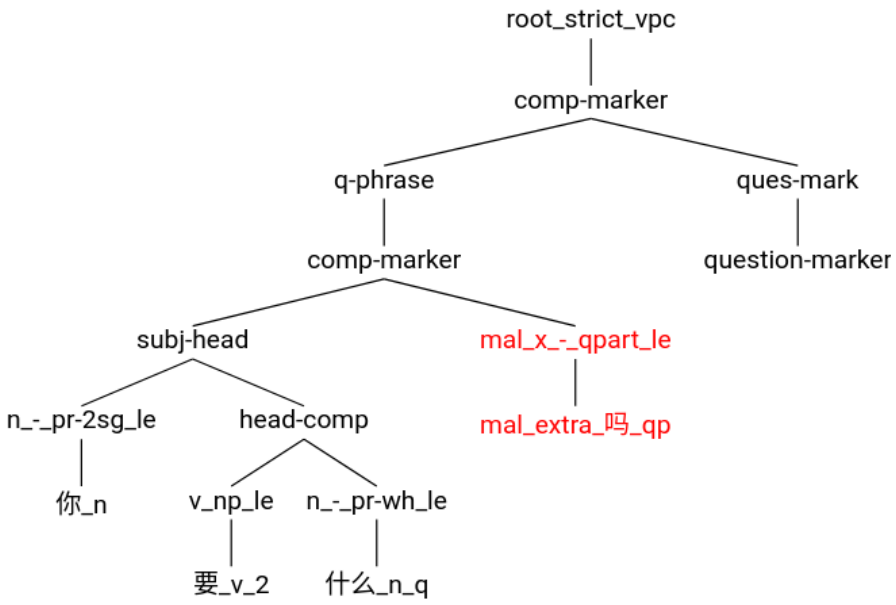


Figure 6.5: Example of mal-lexical entry for a redundant question particle 吗 (*ma*)

Using the mal lexical entry provided in (80), ZHONG is able to parse the ungrammatical sentence (77) – a syntactic representation of this parse is shown in Figure 6.5. All similar ungrammatical sentences (i.e., those where a redundant 吗 (*ma*) particle is added to a sentence that is already a full-fledged question), including the ungrammatical sentence shown in (79), can be detected under a similar analysis.

Usage of 和 (*hé*, and) vs. 也 (*yě*, also)

The second most common error in the NTUCLM is the use of the conjunction 和 (*hé*, and) to coordinate clauses. In the NTUCLM, this error is labeled as "Usage of 和 (*hé*, and) vs. 也 (*yě*, also)", error ID 2 in Table 5.3, because the adverb 也 (*yě*, also) fills much of the semantic function of what would be expected from a clausal conjunction in English. Consider examples (81) through (83).

(81) 我 学 工程 和 法文 。
wǒ xué gōngchéng hé fǎwén.
1SG study engineering and French.language .
'I study engineering and French.'

(82) 我 学 工程 ， 也 学 法文 。
wǒ xué gōngchéng ， yě xué fǎwén.
1SG study engineering ， also study French.language .
'I study engineering and also study French.'

(83) * 我 学 工程 和 学 法文 。
wǒ xué gōngchéng hé xué fǎwén.
1SG study engineering and study French.language .
'I study engineering and also study French.' (intended)

As can be seen from example (81), the conjunction 和 (*hé*, and) is suitable to coordinate nouns or noun phrases. However, in the presence of two clauses, as is shown in examples (82) and (83), 和 (*hé*, and) is not a suitable conjunction. Instead, a comma should be used – very

often followed by the adverb 也 (*yě*, also). This happens, most likely, due to language transfer issues with the English conjunction *and*, which can be used to coordinate both noun phrases and clauses.

To make a grammatical sentence, a comma must be used instead of 和 (*hé*, and). And when the subject of the first clause is the same as the subject of the second clause, the adverb 也 (*yě*, also) is necessary to make the sentence sound fluent.

The solution to this problem has already been hinted at. In order to catch this error, ZHONG now contains a mal lexical entry for 和 (*hé*, and) that has been modified in order to coordinate clauses. A schema for this mal lexical entry is shown in (84).

(84)

$$\left\langle \text{和}, \begin{bmatrix} mal\_clause\_\text{和}\_c \\ \textsc{syn} \begin{bmatrix} \textsc{head} & conj \end{bmatrix} \\ \textsc{sem} \begin{bmatrix} \textsc{lkeys}|\textsc{keyrel} & \begin{bmatrix} \textsc{l-index} & event \\ \textsc{r-index} & event \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle$$

The entry in (84) is essentially the same as the lexical entry for the comma used to coordinate clauses (Mandarin Chinese has two different commas, one to coordinate clauses '，' and one to coordinate/enumerate noun phrases '、'). This mal entry is also very similar to the original entry for (*hé*, and) – as they are both conjunctions. The main difference is that, for the mal lexical entry, the left and right entities being coordinated (L-INDEX and R-INDEX) must both be an *event* (i.e., clauses) – where the original entry sets these values to *individuals* (i.e., nouns or noun phrases).

Using the entry shown in (84), ZHONG is now capable of parsing sentences like the one shown in example (83) – a parse tree for this sentence is shown in Figure 6.6. In fact, the main problem of this sentence is the lack of comma, which means this same mal lexical entry would also be able detect a problem for a sentence like (85), signaling the lack of a necessary comma between clauses even in the presence of the adverb 也 (*yě*, also). A parse tree for (85) is shown in Figure 6.7.

(85) * 我 学 工程 和 也 学 法文 。

wǒ xué gōngchéng hé yě xué fǎwén.
1SG study engineering and also study French.language .

'I study engineering, and also study French.' (intended)

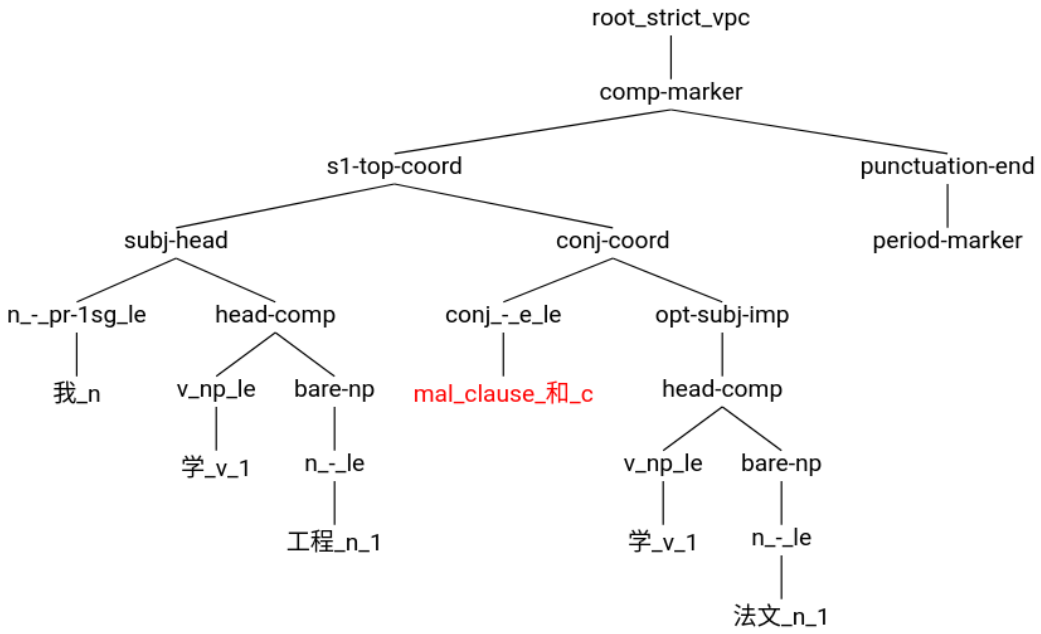


Figure 6.6: Example of mal-lexical entry for 和 (*hé*, and) as a clausal conjunction

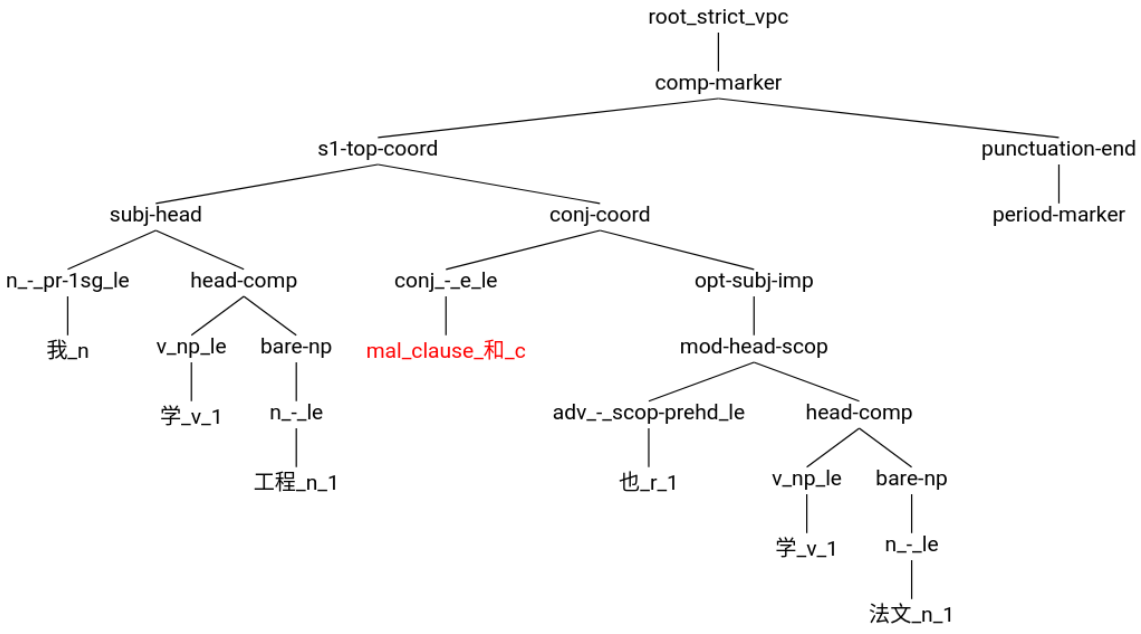


Figure 6.7: Second example of mal-lexical entry for 和 (*hé*, and) as a clausal conjunction

Usage of 是 (*shì*, to be) with adjectival predicates

The third error I will discuss is a common problem related to adjectival predicates in Mandarin Chinese. Unlike English, Mandarin Chinese does not always need a verb to head the main predicate of a complete sentence. This error corresponds to ID 4 in Table 5.3. Examples (86) through (88) illustrate some minimal pairs to understand the problem in question.

(86) 她 很 美 。
Tā hěn měi .
3SG.FEM very beautiful .
'She is beautiful.'

(87) * 她 是 美 。
Tā shì měi .
3SG.FEM COP.be beautiful .
'She is beautiful.' (intended)

(88) * 她 是 很 美 。
Tā shì hěn měi .
3SG.FEM COP.be very beautiful .
'She is beautiful.' (intended)

In short, adjectival predicates in Mandarin Chinese do not need (and should not use) the copula verb 是 (*shì*). However, it is important to point out that this might be a slight overstatement. This is a phenomena where restrictions vary slightly from speaker to speaker. However, in a prescriptive environment, such as a Mandarin Chinese classroom, sentences like (87) and (88) are usually considered mistakes. Instead of using the copula verb, adjectival predicates should be modified by a degree adverbial (i.e., a degree specifier) instead. 很 (*hěn*, very) is, by far, the most common degree specifier, and even though it is often glossed as *very*, it actually carries a fairly neutral degree – which can be seen in the translation provided for (86). Other common degree specifiers include 非常 (*fēicháng*, extremely), 真 (*zhēn*, truly) and 有点儿 (*yǒudiǎnr*,

somewhat). The presence of degree specifiers effectively license adjectival predicates, becoming capable of taking a subject directly without any intervening verb – as can be seen in (86).

The simplest way to address this type of error was to create a mal lexical entry for a dummy copula 是 (*shì*), that behaves like a transitive verb by selecting an adjective phrase as its complement. A schema of this lexical entry is shown as (89). This lexical entry is fairly simple. It is essentially a copula verb that raises its subject and links it to the subject of its complement. In this mal lexical entry its complement is restricted to adjective phrases (with or without a degree specifier). Following the restrictions of normal adjectival predicates, the subject of this mal lexical entry can be a noun phrase, but it can also be full clause, as seen in examples (90) and (91).

(89)

$$\left\langle \text{是}, \left[\begin{array}{l} \textit{mal\_apred\_}\,\text{是}\,\textit{\_cop} \\ \text{SYN} \left[\begin{array}{l} \text{HEAD}\ \textit{verb} \\ \text{VAL} \left[\begin{array}{ll} \text{SUBJ} & \langle \ldots \rangle \\ \text{COMP} & \langle \text{AP} \rangle \end{array} \right] \end{array} \right] \end{array} \right] \right\rangle$$

(90) 吃 肉 很 贵 。
chī ròu hěn guì .
eat meat very expensive .
'Eating meat is expensive.'

(91) * 吃 肉 是 很 贵 。
chī ròu shì hěn guì .
eat meat COP.be very expensive .
'Eating meat is expensive.'

Using the mal lexical entry shown in (89), ZHONG is now able to parse ungrammatical sentences like (87), (88) and (91). The syntactic representation produced for (87) is provided in Figure 6.8. Similar to other mal-rules, this analysis generalizes for all other sentences where adjectival predicates are preceded by the copula 是 (*shi*).

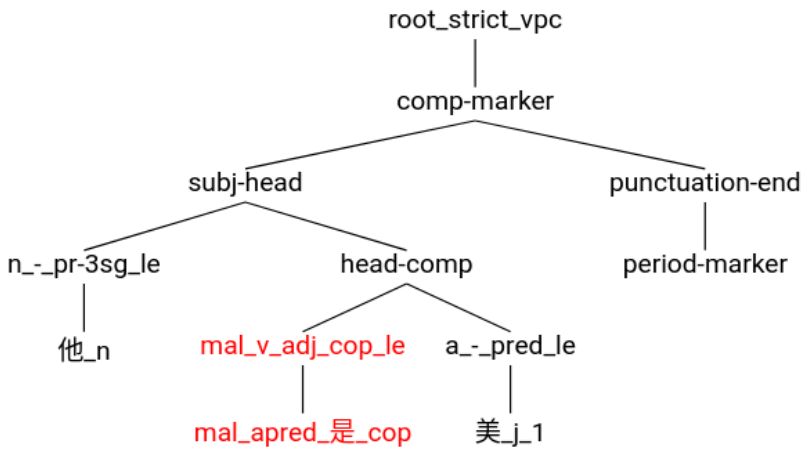


Figure 6.8: Example of mal-lexical entry for 是 (*shì*) taking adjectival predicates

Bare adjectival predicates

Another common error related to the one discussed above is the use of bare adjectival predicates – i.e., adjectival predicates without a degree specifier. This error corresponds to ID 8 in Table 5.3, and is exemplified through the minimal pair shown as (92) and (93).

(92) 她 很 美 。
Tā hěn měi .
3SG.FEM very beautiful .
'She is beautiful.'

(93) * 她 美 。
Tā měi .
3SG.FEM beautiful .
(intended) 'She is beautiful.'

As discussed above, adjectival predicates should, in principle, be modified by a degree specifier. This rule does not hold in some contexts, such as comparative or contrastive constructions, but it is a general rule followed in low proficiency prescriptive environments.

Because there is no lexical entry on which to support the error detection (i.e., this error shows consists of a missing word), this error is currently detected through a robust grammar

root. Roots define what can be the output of a grammar. Usually, this is equivalent to the definition of a complete sentence, however roots can also be used to license sentence fragments.

Normal grammar roots ensure that all the requirements to license a full sentence are satisfied. Sentences that generate a parse without a compatible root end up being rejected in the latest stage of the parsing process. The root conditions for adjectival predicates share a great number of requirements with other roots (e.g., being compatible with forming a main clause, having the subject and complements filled or optional, etc.).

One main difference for adjectival predicate sentences is that the strict root for this kind of sentences also ensures that the specifier (SPR) is empty (i.e., filled or optional) – which is not a requirement for roots headed by verbs. In ZHONG, this is also what guarantees that adjectival predicates have a degree specifier. (94) shows the schema for a robust root for adjectival predicates where the requirement for the specifier (SPR) to be empty is changed to a requirement that it has not yet been filled. This ensures that this root is only able to license sentences with adjectival predicates that do not have a degree specifier.

(94)
$$\left[\begin{array}{l} \textit{root\_rbst\_nospr\_adj} \\ \left[\begin{array}{ll} \textsc{head} & \textit{adj} \\ \textsc{mc} & + \\ \textsc{syn} & \left[\begin{array}{ll} \textsc{subj} & \langle\rangle \\ \textsc{val} & \left[\begin{array}{ll} \textsc{comp} & \langle\rangle \\ \textsc{spr} & \langle \ldots \rangle \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right]$$

Using the root in (94), ZHONG can now parse sentences like the one shown in (93). The syntactic representation for this sentence is shown in Figure 6.9.

Usage of 有点儿 (*yǒudiǎnr*, somewhat) vs. 一点儿 (*yīdiǎnr*, a bit)

A slightly different type of error is the confusion between the words 有点儿 (*yǒudiǎnr*, somewhat) and 一点儿 (*yīdiǎnr*, a bit). This error corresponds to ID 7 in Table 5.3. The word 一点儿 (*yīdiǎnr*) can be translated as *some*, *a piece* or *a bit*. In ZHONG, this is actually treated

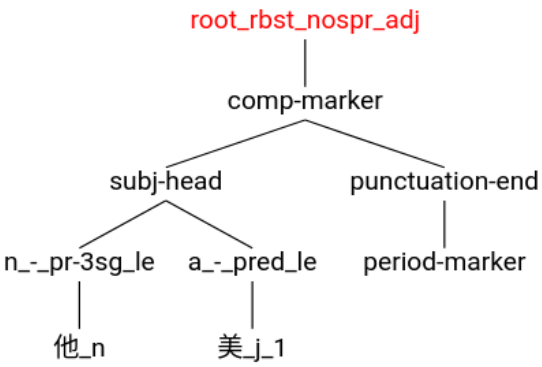


Figure 6.9: Example of robust root to allow bare adjectival predicates

compositionally, from the numeral 一 (*yī*, one) and the measure word 点儿 (*diǎnr*, piece). The word 有点儿 (*yǒudiǎnr*, somewhat), on the other hand, is a degree specifier – suitable to build sentences like (95).

Early learners of Mandarin Chinese often have trouble remembering this distinction, especially since *a bit* can be used as a degree adverb in English (e.g., *This is a bit expensive*). This confusion leads students to try to build sentences like the one in (96). This is not, however, an acceptable sentence. 一点儿 (*yīdiǎnr*, some) should only be used as a classifier phrase, as shown in (97).

(95) 南大 的 宿舍 有点儿 贵 。
nándà de sùshè yǒudiǎnr guì.
NTU POSS.PART dormitory somewhat expensive .
'NTU's dormitories are a bit expensive.'

(96) * 南大 的 宿舍 一点儿 贵 。
nándà de sùshè yīdiǎnr guì.
NTU POSS.PART dormitory some expensive .
'NTU's dormitories are a bit expensive. (intended)'

(97) 我 要 一点儿 蛋糕 。

Wǒ yào yīdiǎnr dàngāo .

1.SG want some(one+piece) cake .

'I want a bit of cake.'

This error can actually be considered as an instance of a broad class of misspelling mistakes. The source of this problem is most likely a mix of orthographic similarity and transfer problems arising from the English translation.

Lee et al. (2016a) identify various sources for Chinese spelling errors. The most important are linked to problems arising from a confusion among words that are phonologically or visually similar but semantically distinct. In Mandarin Chinese, spelling mistakes are tighly linked to the input method used to write a sentence. Hand-written input often generates more errors that are visually similar, while keyboard input (which often relies on phonological inputs for characters) tends to generate more spelling errors with a phonological source.

In the NTUCLM, a number of other common spelling mistakes were also found. These include, for example, misspelling the verb 有 (yǒu, to have) as 友 (*yǒu*, friend / friendly). 友 (*yǒu*) is a bound character and cannot stand on its own. It is, however, a fairly frequent character – appearing in words like 朋友 (*péngyou*, friend) and 室友 (*shìyǒu*, roommate). An example of this error is shown in (98).

(98) * 他 友 一 个 室友 。

Tā yǒu yī gè shìyǒu .

3SG.FEM friend one CLS roommate .

'He has a roommate. (intended)'

This spelling mistake can be argued to have both orthographic and phonological basis – the characters 有 and 友 not only share some orthographic resemblances, but also have the same pronunciation. Another common mistake in the NTUCLM was mispelling the word 老师 (*lǎoshī*, teacher) as 老帅 (*lǎoshuài, old + handsome*). This error would be be said to have an orthographic basis because the characters 师 (*shī*) and 帅 (*shuài*) differ orthographically by only one stroke, but have different pronunciations.

Orthographic errors can be easily solved by creating mal lexical entries that is essentially the same as the correct spelling of a word (including any semantic predicates they may introduce), differing only in their STEM value – i.e., the string that is matched to the sentence's surface form.

Figure 6.10 shows the syntactic representation for example (96) where 一点儿 (*yīdiǎnr*, a bit) is interpreted as a misspelling of 有点儿 (*yǒudiǎnr*, somewhat). Figure 6.11 shows the syntactic output for example (98), where the character 友 (*yǒu*, friend / friendly) is analyzed as a misspelling of the verb 有 (yǒu, to have).

ZHONG currently has more than 30 errors that can be largely considered spelling mistakes. Most of these errors were identified from a careful analysis of errors that did not have a label during the annotation efforts of the NTUCLM – shown as ID 20 in Table 5.3.

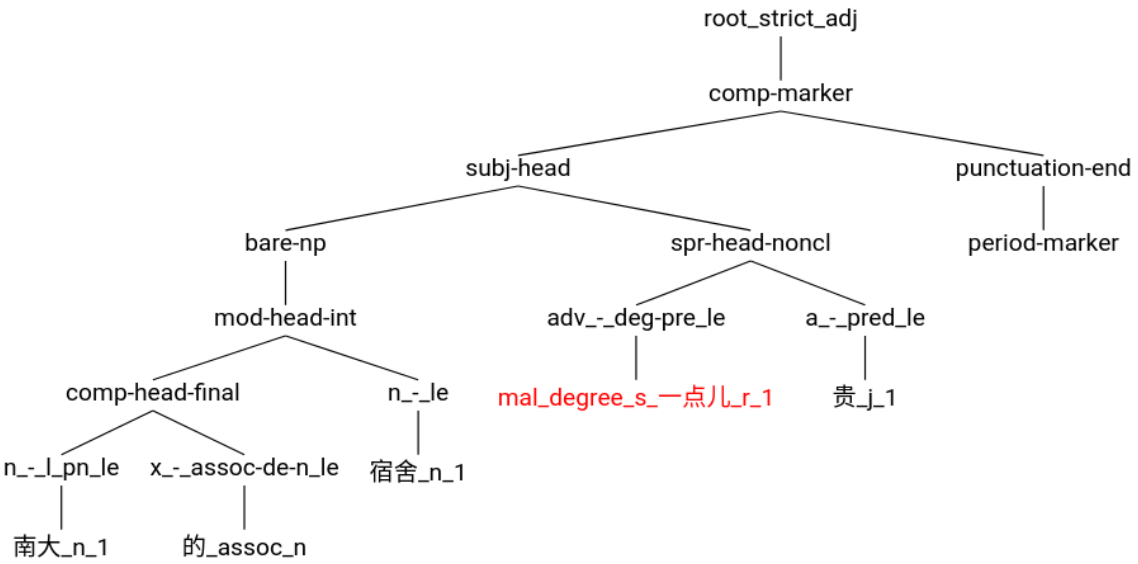


Figure 6.10: Example of mal-lexical entry for 一点儿 (*yīdiǎnr*, a bit) as a degree specifier

Usage of 不 (*bù*, no) vs. 没有 (*méiyǒu*, no)

The next error I will discuss is an extension of the discussion introduced in Section 6.4, concerning interactions between negation, aspect and a few special verbs. In particular, the constraint that stopped the negator 不 (*bù*, no) from modifying the verb 有 (yǒu, to have) – summarized in the discussion of examples (55) through (58). This error corresponds to ID 15 in Table 5.3.

While early learners of Mandarin Chinese have only a very limited knowledge of aspect

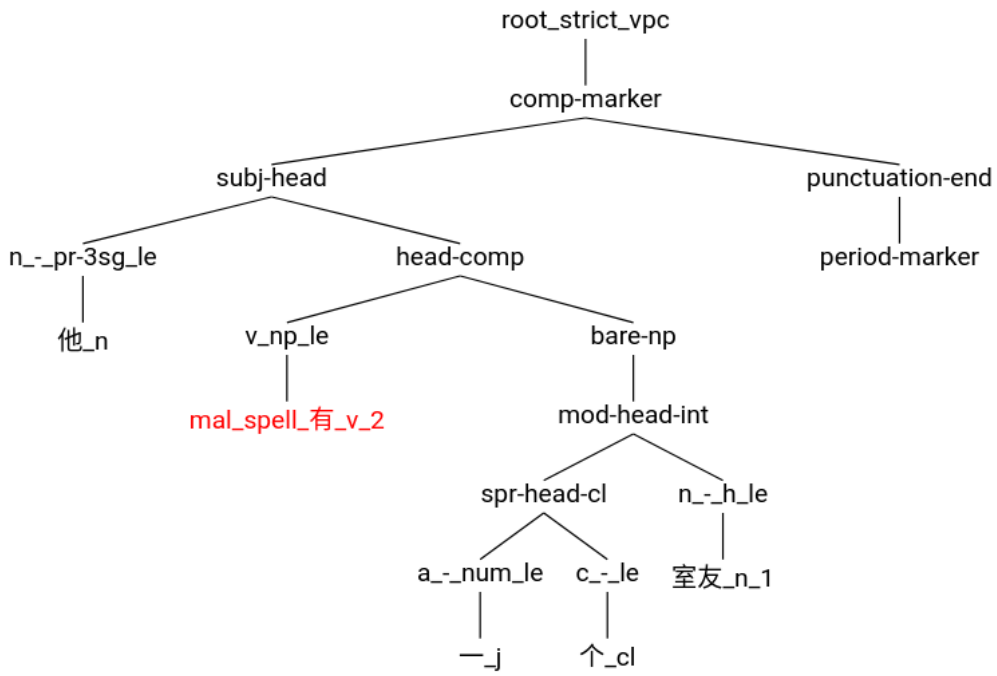


Figure 6.11: Example of a mal lexical entry capturing 有 (*yǒu*, to have) misspelled as 友 (*yǒu*)

particles (limiting the presence of errors concerning these particles), attempting to use 不 (*bù*, no) to negate 有 (yǒu, to have) is a fairly common error in the NTUCLM.

In order to parse sentences like (57), repeated here as (99), ZHONG currently has a entry for 不 (*bù*, no) that mimics the aspect requirements of 没 (*méi*, no) – the only negator compatible with 有 (yǒu, to have). This entry is shown in (100). These aspect requirements have been discussed above, and will not be repeated. However, in addition to aspect requirements, this entry has one more essential feature – it can only be used to modify the verb 有 (yǒu, to have), which is exemplified through feature $\left[\text{EXISTV } +\right]$. This feature was first introduced to ZHONG by Wang et al. (2015b), for the implementation of A-NOT-A constructions, and it is used only to select the verb 有 (yǒu, to have).

(99) * 我 不 有 钱 。
wǒ bù yǒu qián .
1SG NEG.bu have money .
'I don't have money.' (intended)

(100)

$$\left\langle 不, \left[\begin{array}{l} \textit{mal\_}\ 不\ \textit{\_r} \\ \textsc{syn} \left[\begin{array}{ll} \textsc{head} & \textit{adverb} \\ \textsc{val} & \textsc{mod} \left\langle \left[\begin{array}{ll} & \textit{verb} \\ \textsc{existv} & + \\ \textsc{lzg} & \textit{guo-zhe-asp} \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

Using an entry like the one shown in (100), ZHONG is now able to parse sentences like (99). The syntactic representation for (99) is shown in Figure 6.12.

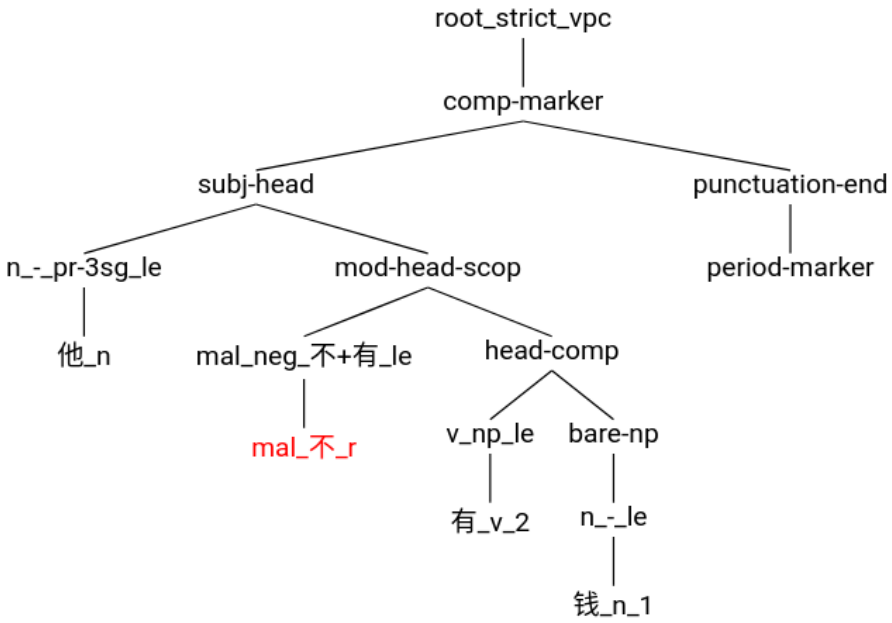


Figure 6.12: Example of mal-lexical entry for 不 (*bù*, no) negating the verb 有 (*yǒu*, to have)

Bare Nominal Predicates

The final syntactic mistake I will discuss here is another common error that was not originally in the 20 error tags used to tag the NTUCLM – the use of bare nominal predicates. This error was identified from the analysis of errors that did not have a specific label – bundled as 'Other Errors' or ID 20 in Table 5.3.

In Mandarin Chinese, even though adjectival predication (discussed above) should not use the copula verb 是 (*shì*), nominal predication requires it – see (101) and (102). Regardless, most

likely due to interference with adjectival predication, many students attempt to build sentences like the one shown in (102) – effectively using noun phrases as predicates. This is, however, ungrammatical.

(101) 我 是 大学生 。

Wǒ shì dàxuéshēng.

1SG COP.be university.student .

'I am a university student.'

(102) * 我 大学生 。

Wǒ dàxuéshēng.

1SG university.student .

'I am a university student.' (intended)

ZHONG currently addresses this error through the use of a mal pumping rule, shown in (103). The role of this pumping rule is to transform any fully specified noun phrase (shown on the right of the rule) into something similar to an intransitive verb (shown on the left of the rule). The valence of the noun phrase is transformed to expect a subject, and its head value changed to *verb* through the introduction of a copula verb predicate in the semantics. This pumping rule essentially transforms any noun phrase into the equivalent of a verb phrase headed by 是 (*shì*, to be).

(103)

$$\begin{bmatrix} \textit{mal\_np\_as\_vp} \\ \text{SYN} \begin{bmatrix} \text{HEAD } \textit{verb} \\ \text{VAL} \begin{bmatrix} \text{SUBJ} & \langle \text{NP} \rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \longrightarrow \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{HEAD } \textit{noun} \\ \text{VAL} \begin{bmatrix} \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

By making use of the mal pumping rule shown in (103), sentence (102) can be licensed by the tree provided in Figure 6.13. Here, one can see that 大学生 (*dàxuéshēng*, *university student*)

starts off as a noun phrase before being pumped into a verb phrase, capable of taking 我 (*wǒ*, *I*) as its subject.

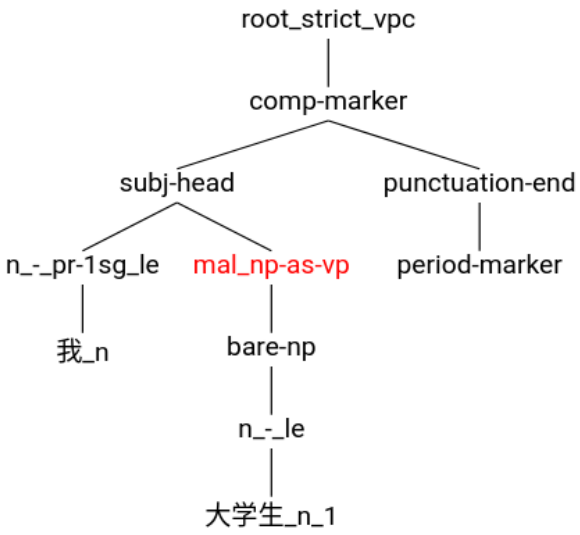


Figure 6.13: Example of a mal-rule enabling bare nominal predicates

6.7 Diagnosing errors through semantics

In the final section of this chapter, I will discuss semantic errors. These are common errors that cannot or should not be identified using mal-rules because they do not necessarily represent a syntactically deprecated sentence.

The NTUCLM contains a few of this kind of errors. One such example is the lexical conflation between the concepts 中国 (*zhōngguó*, China) and 中文 (*zhōngwén*, Chinese language) – which corresponds to error ID 5 in Table 5.3. I will use examples (104) through (106) to discuss this error.

(104) 我 说 中文 。
Wǒ shuō zhōngwén .
1SG speak Chinese.language .
'I speak Chinese.'

(105) ? 我 说 中国 。

Wǒ shuō zhōngguó .
1SG speak China .

'I speak Chinese.' (intended)

(106) 我 说 中国 。

Wǒ shuō zhōngguó .
1SG say China .

'I say "China".'

Although sentences such as (105) are not strictly ungrammatical, as shown by (106), it is too often used with the meaning contained in (104). Students use the word 中国 (*zhōngguó*, China) with the intended meaning of 中文 (*zhōngwén*, Chinese language).

Even though this cannot be treated as a syntactic error, it would still be possible to identify frequently used non-prototypical complements using mal lexical entries. For the example in question, (107) provides a special entry for the verb 说 (*shuō*, to talk). Based on the NTUCLM, students often use the word 中国 (*zhōngguó*, China) as the complement of 说 (*shuō*, to talk), instead of the desirable and most likely intended 中文 (*zhōng wén*, Chinese Language).

(107)

$$\left\langle \text{说}, \left[\begin{array}{l} \textit{mal\_talk\_china} \\ \left[\begin{array}{ll} \textsc{head} & \textit{verb} \\ \textsc{val} & \left[\textsc{comp} \left\langle \left[\begin{array}{ll} \textit{noun} \\ \textsc{form} & \text{中国} \end{array} \right] \right\rangle \right] \end{array} \right] \end{array} \right] \right\rangle$$

Technically speaking, the mal lexical entry shown in (107) would be able to selectively flag sentences for which 中国 (*zhōng guó*, China) is the complement of the verb 说 (*shuō*, to talk). In the lexical entry, this is achieved by the feature FORM inside the complement. However, since the grammar would then have two competing entries for the verb 说 (*shuō*, to talk), this would generate (spurious) ambiguity for the sentence (105)/(106). In addition, the same rule would not be able to detect a very similar problem, shown as (108).

(108) ? 你 的 中国 名字 很 好 。

Nǐ de zhōngguó míngzì hěn hǎo .

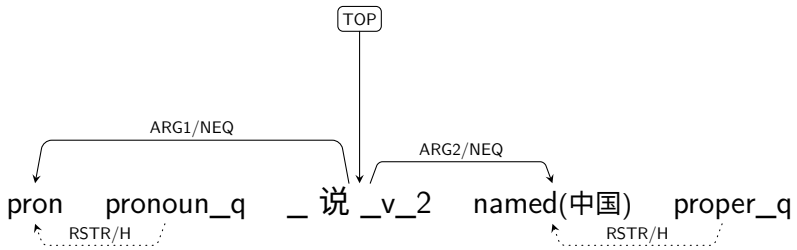
2SG POSS.PART China name very good .

'Your Chinese name is nice.' (intended)

With regards to (108), even though one could most certainly argue for the grammaticality of a forced noun-noun compound between the words *China* and *name*, it is not clear what it could mean. In any case, if the intended meaning is *Chinese name*, (108) must also considered a infelicitous (although not strictly ungrammatical) sentence. This problem cannot, however, be captured with mal lexical entry shown in (107).

An alternative way to detect this kind of errors, without generating unnecessary ambiguity, would be to use the semantics produced by grammar. Consider the simplified semantic representation for the sentence shown in (105)/(106), shown in (109) as a DMRS dependency graph (Copestake, 2009).

(109)



This semantic representation shows that 中国 (*zhōngguó*, China) is the ARG2 of 说 (*shuō*, to speak, to say) – i.e., *what is said*. Instead of creating a *mal* lexical entry for 说 (*shuō*, to speak, to say), a simple semantic check can be used to see if 中国 (*zhōngguó*, China) is used as the ARG2 of the verb 说 (*shuō*, to speak, to say). Given the deep semantic analysis performed by ZHONG, the semantic arguments are also easily detectable in the presence of discontinuous arguments (e.g., topicalization, modification, intervening aspect particles, etc.) – which can be a problem when using shallow text based methods. This method would also be capable of identifying problematic noun-noun compounds, as discussed for (108).

This type of error detection relies on a theoretically sound grammar, and not on mal-rules. In addition to the error discussed above, many of the error classes identified in the NTUCLM

are, in fact, better suited to be identified through semantic checks. Because of this, they were left outside the scope of this thesis. Error classes in the NTUCLM that would be best detected by semantic checks include, for example, problems concerning the scope of negation (Error ID 10), problems concerning the choice of sortal classifiers (Error ID 11) and the syntactic order of nominal 的 (*de*, possessive marker) modification (Error ID 17).[10] Designing semantic checks to detect these and other similar errors is an excellent direction for future work.

6.8 Summary

In this chapter, I gave an overview of various lexical and syntactic improvements made to ZHONG, including detailed analyses of separable verbs, and of key interactions between negation and aspect particles. The chapter concluded with a detailed account of many mal-rules and mal lexical entries designed to capture a variety of common mistakes identified in the NTU Corpus of Learner Mandarin. The list of errors described here is not exhaustive. ZHONG currently has more than 60 mal types (including mal lexical entries and rules), and it would have been impossible to describe them all in detail. A full account of this work can be found in the grammar, which is released under an open-source license.

The work presented in this chapter will be evaluated in experiments that measure ZHONG's improved parse coverage as well as its new-found ability to detect a variety of common grammatical errors made by early learners of Mandarin Chinese. The results of this evaluation will be presented in Chapter 9.

Finally, I would like to note that, due to the wide and interdisciplinary scope of this thesis, it was impossible to provide a discussion that exhausted the linguistic complexity of any of the covered topics. Instead, the improvements described in this chapter focused on two goals: a) to equip ZHONG with enough detail to deal with the language complexity expected of early learners of Mandarin Chinese; and b) to provide deep enough linguistic analyses to allow the design of mal-rules capable of diagnosing common errors made by the same population. As such, ZHONG is still very much a work in progress, and will welcome and foster work in Mandarin Chinese theoretical linguistics for many years to come.

[10] Error IDs are provided in reference to Table 5.3

Chapter 7

Learner Treebanks and Parse Ranking Models

This chapter reports on the creation of the Tembusu Treebank, a multilingual learner treebank, containing data from the NTU Corpus of Learner English, the NTU Corpus of Learner Mandarin Chinese and the Mandarin Education Corpus. It describes the process of tagging, as well as its exploitation to create new mal-rule enhanced parse-ranking models (for both the ERG and ZHONG). These new models are the first large scale effort of this kind, and aim to improve the selection quality of parses using mal-rules, which I show translates into improved error detection and diagnosis in Chapter 9.

7.1 Tembusu Treebank: a Multilingual Learner Treebank

The tembusu is a large evergreen tree, native to Southeast Asia, and it is *unofficially* recognized as Singapore's national tree – where the data for this treebank was collected. To the best of my knowledge, the Tembusu Treebank is the first of its kind: a multilingual treebank using precision parsers enhanced by mal-rules.

Despite its uniqueness, the Tembusu Treebank is in the same spirit of a few existing projects: a project named Syntactically Annotating the Language of Learner English (SALLE, Ragheb and Dickinson, 2012, 2014), as well as two other projects that overtly follow in SALLE's footsteps – the Universal Dependencies for Learner English (Berzak et al., 2016) and a similar

although much smaller project for Mandarin Chinese (Lee et al., 2017).

These three projects use syntactic dependency-style analysis to hand-annotate learner data. Even though these projects can, most certainly, be useful to inform tasks such as grammatical error detection and/or correction, it becomes clear that they are mainly concerned with increasing the robustness of statistical parsers, increasing their ability to reasonably deal with non-canonical language. In other words, to allow parsers to produce reasonable results when dealing with ungrammatical text produced by L2 learners of a given language (a type of non-canonical language), statistical parsers need to include this kind of text as training data. And although these goals are linked with the goals of this dissertation, they do not fully align.

The three projects mentioned above focus on discussing and establishing a reasonable layer of dependency annotations when presented with sentences that would not be able to be annotated using standard guidelines (designed for the canonical use of language). None of the projects attempt to diagnose or annotate the source or kind of errors present in the data. And despite the fact that the Universal Dependencies for Learner English project (Berzak et al., 2016) does provide a corrected version of each ungrammatical sentence (tagged using the standard universal dependency guidelines – and probably useful when compared with their ungrammatical counterparts), none of the three projects explicitly elaborate on how these annotations can be used to benefit the task of error detection, error correction, or as a means to provide corrective feedback. In general, these projects are essentially working towards certain classes of errors (or non-canonical language) being ignored by parsers by attempting to '*reduc[e] the impact of grammatical errors in automatic annotation*' (Berzak et al., 2016).

Some of the main differences between the Tembusu Treebank and the above projects are:

- the Tembusu Treebank uses a precision grammar and a web system to annotate trees, while other projects use hand-annotations (i.e., direct human labeling). Both methods have advantages and disadvantages: using a precision grammar assumes a theoretical model of the error has been previously developed, and it can provide deeper morphosyntactic and semantic information about each annotated tree. However, when a precision grammar is used, it is not possible to provide annotations for sentences with phenomena or errors not covered by a grammar (something possible when annotating by hand).

- because labels in the Tembusu Treebank are directly tied to the formal descriptions of a specific grammar (including mal-rules), the annotations provided by the Tembusu Treebank can be used to describe which constraints are being violated by an ungrammatical sentence (i.e., why a sentence ungrammatical). The three projects mentioned above label their data in a way similar to that of a grammatical sentence – and hence would be difficult to describe exactly where or why a sentence is ungrammatical;

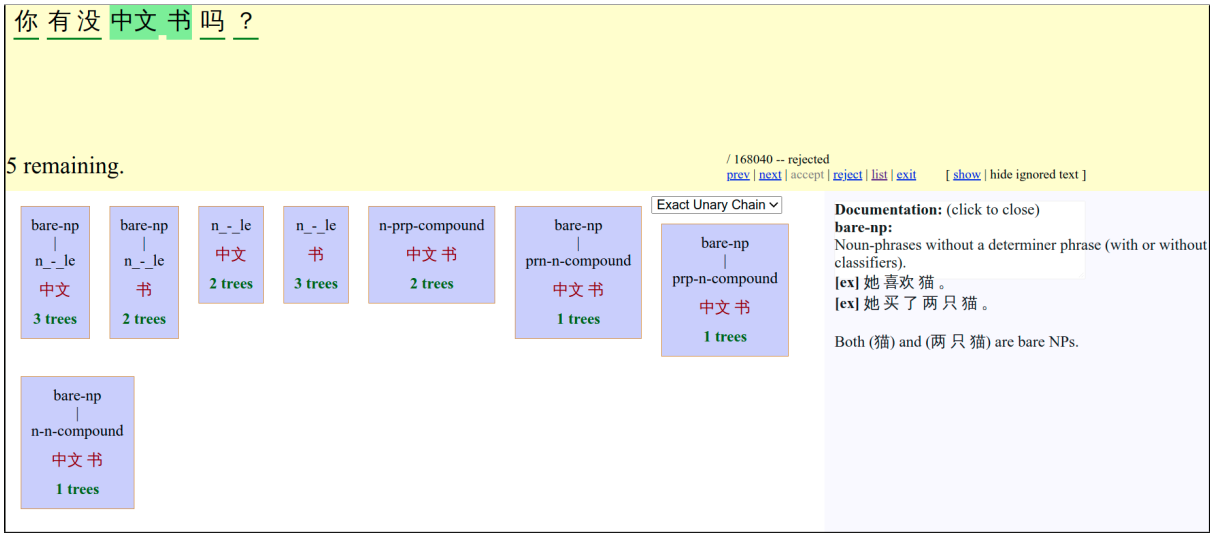
- because it is produced from precision grammars, the Tembusu Treebank is uniquely suited to train these grammars without compromising their flexibility and precision. The inherent upside of having detailed annotations is the fact that these annotations can be used by simpler systems (e.g., using automatic many-to-one mappings of fine-grained linguistic labels into more coarse labels). However, the reverse is not true. As such, while the data provided by the Tembusu Treebank could be converted into Universal Dependencies (with some amount of work to produce adequate mappings), the reverse would probably not be possible to be done in an unambiguous way;

FFTB: an Enhanced Version

The Tembusu Treebank used an enhanced version of the Full Forest Treebanker tool (FFTB, Packard, 2015) introduced in Chapter 4.3. These enhancements included small changes to be able to securely serve the FFTB as a web-service (so the results of remote annotation could be centralized in a server), and improvements to the user-interface of the FFTB, providing in-tool access to grammar documentation, and to make mal-rules visually distinct from other rules within a grammar.

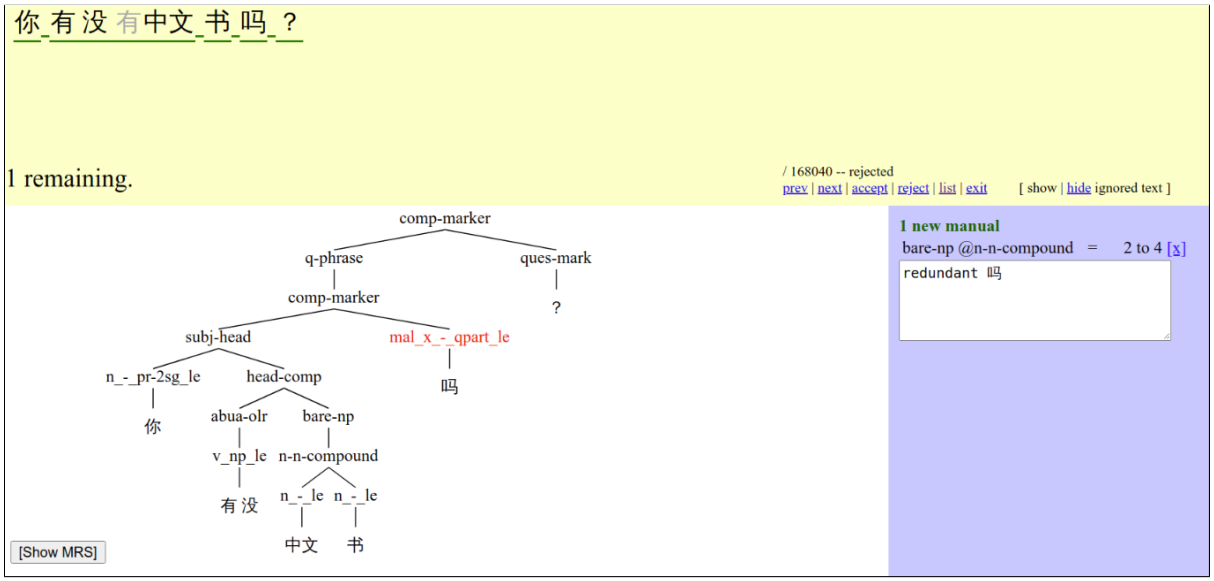
Using the FFTB, the annotation process consists mainly of choosing constituent boundaries and the rules through which different constituents are licensed by a specific grammar. This requires considerable knowledge of the grammar being used in the treebank. Even though grammars try their best to adopt neutral, language agnostic names for their rules, most grammars still have idiosyncrasies in the naming or in the analysis of certain constructions that need to be understood by the the annotators. The enhanced version of the FFTB, used to build this treebank, tries to help annotators by providing the documentation for each type when hovering over either

the tree or a discriminant. This documentation is extracted from the grammar, using the Linguistic Type Database (LTDB, Hashimoto et al., 2007, 2008). However, if this documentation is unavailable in the grammar, this feature becomes irrelevant.



Available discriminants for the constituent spanning '中文书' (zhōngwén shū, Chinese book)

Figure 7.1: Enhanced Full Forest Treebank: grammar documentation;



Mal lexical entry for the incorrect use of question particle 吗 (ma);

Figure 7.2: Enhanced Full Forest Treebank: mal-rule highlight

Figures 7.1 and 7.2 provide screenshots for the enhanced version of the FFTB tool. Figure 7.1 shows the view for the selection of discriminants, along with the documentation provided, by ZHONG, for the type *bare-np*. Figure 7.2 shows the final tree of a sentence, high-

lighting a mal-rule noting the incorrect use of the question particle 吗 (*ma*). This error was discussed in Section 6.6, and captures two clashing ways of forming polar questions in Mandarin Chinese.

7.1.1 Treebanking English Learner Data

The English portion of the Tembusu Treebank currently includes 4,900 sentences from the NTU Corpus of Learner English (NTUCLE), introduced in detail in Section 5.2. More specifically, the data comes from the NTUCLE-X version of NTUCLE – taking full advantage of improved sentence selection and segmentation, as well as data that was unavailable in the original version. The treebank used the latest available version of the English Resource Grammar, known as the 'make-over' version (ERG-MO). This grammar was created using the 'trunk' branch of ERG's SVN repository[1] (Revision 29199).

To build this treebank, I employed the help of five student assistants (four undergraduate students and one graduate student), all majoring in Linguistics and Multilingual Studies. All four undergraduate students had previously attended a Syntactic Theory course, where they were introduced to Head-Driven Phrase Structure Grammar. Students with a good final-grade in this course were invited to participate. The four students went through a treebanking training exercise, which will be described in greater detail below. The fifth (graduate) student was already experienced with HPSG theory and implementation, and was also familiar with some of the naming conventions and idiosyncrasies of the ERG – making the treebanking process much easier.

Even though all four students had successfully completed (with excellent grades) a course on the theoretical inner-workings and assumptions of HPSG, treebanking sentences with a real grammar is a very different experience.

When developing a grammar, a grammarian needs to decide on the best analyses for different linguistic phenomena. As such, real grammars (in this case, the ERG) have their own assumptions, which are not always intuitive and need to be learned. For example, the destination (e.g., 'to Beijing') in a sentence such as 'She went to Beijing' is treated as an adjunct in

[1] `http://svn.delph-in.net/erg/trunk`

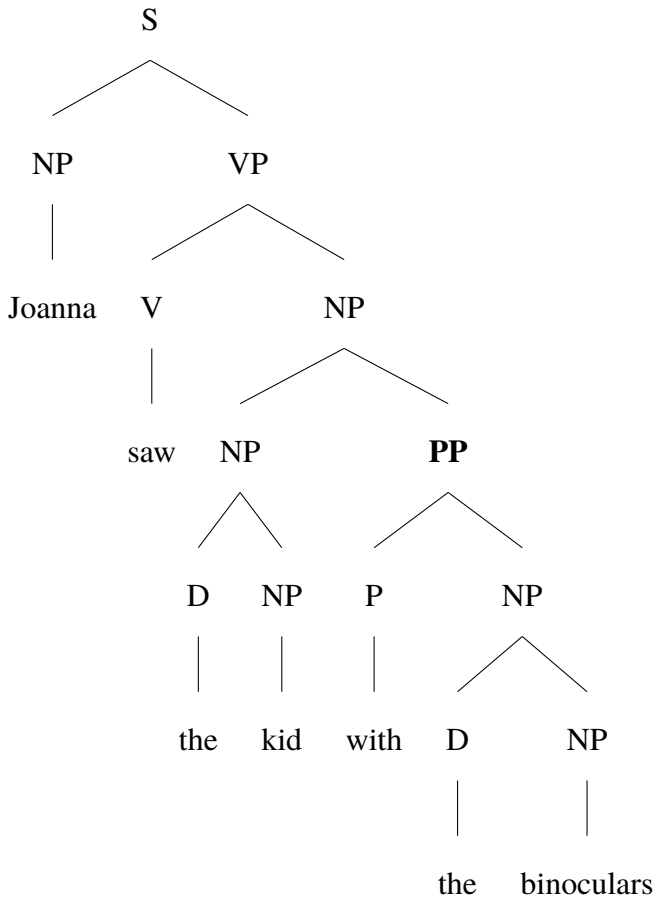
the ERG – while other grammarians/grammars might treat it as a complement. Much of the variety that can be found in implemented grammars mirrors current linguistic discussions. It is not always clear what is the best analysis for certain linguistic phenomena. As such, treebankers need to spend some time learning many of the intended analysis for a variety of phenomena.

In addition to this layer of idiosyncrasies all grammars possess, the general task of treebanking also deals with a very important dimension of language (and a very important problem in Natural Language Processing): ambiguity. Even considering that treebankers have full-knowledge of a grammar's inner-workings, the task of treebanking a sentence includes resolving any inherent ambiguity that sentence may have. (110) shows and example of ambiguity in PP attachment.

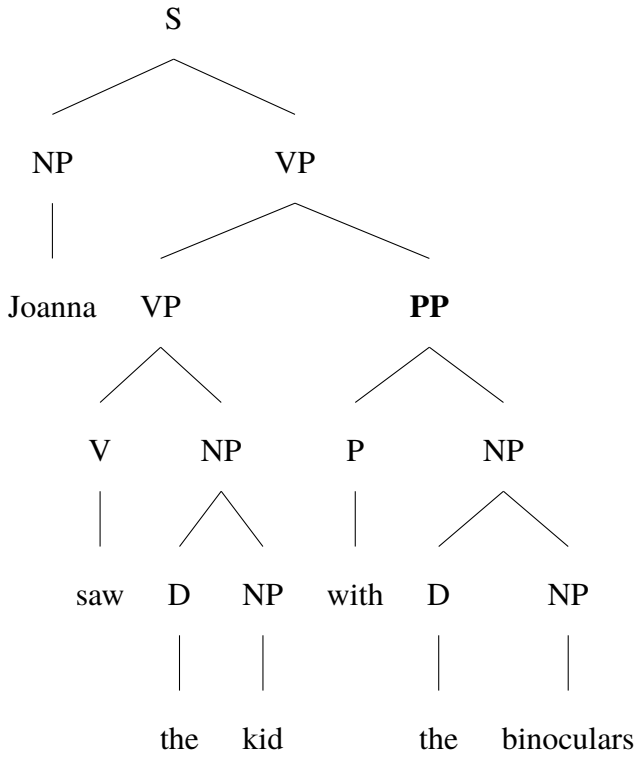
(110) Joanna saw the kid with the binoculars.

Considering (110) as an example, and (111) and (112) as two possible analyses, the semantic ambiguity should become clear. In the reading provided by (111), the kid that Joanna saw was carrying or using the binoculars. While the reading captured by (112) describes a situation where the binoculars were used as an instrument in the act of seeing (i.e., it was Joanna carrying the binoculars).

(111)


S
NP
VP
Joanna
V
NP
saw
NP
PP
D
NP
P
NP
the
kid
with
D
NP
the
binoculars

(112)


S
NP
VP
Joanna
VP
PP
V
NP
P
NP
saw
D
NP
with
D
NP
the
kid
the
binoculars

Whenever there is available context, this decision of how to resolve ambiguity should be done with this context in mind. Sometimes, common-sense knowledge also plays a part in this disambiguation process. For example, one could argue that (112) makes more sense, especially given the nature of the verb and the common sense knowledge of what binoculars are used for. However, more often than not, the context is not enough to resolve all available ambiguity. In the example shown as (113), it is arguably more difficult to decide if the purchase happened on Tuesday or if the concert will happen on Tuesday. Part of the training sessions of a treebanking exercise includes coming up with guidelines about what to do in the presence of this type of ambiguity, especially when common sense and/or context cannot help solve it.

(113) Yasu bought the tickets for the concert on Tuesday.

Finally, as it was discussed above, one of the main differentiators of the Tembusu Treebank is the fact that it uses mal-rules. This is also a task without much precedent. In this case, a new set of guidelines was also developed to help students understand how to use mal-rules, including situations where they should not be used even if they were available. In the presence of multiple ways of correcting a sentence, treebankers were instructed to select the most natural correction (similar to what is done for disambiguation – using context and common sense knowledge). When mal-rules were available but none of the possible corrections was diagnosing/reconstructing a plausible interpretation given the context, treebankers were instructed to reject these sentences. This was done to prevent the overuse of very broad-spectrum mal-rules that appear in a variety of different contexts unless they specifically target a plausible error in the sentence.

Dr. Daniel Paul Flickinger – the main developer and maintainer of the ERG – kindly agreed to be a part of the training provided to treebankers. His participation was essential to present and discuss some of the main assumptions and idiosyncrasies of the ERG, as well as to better understand some of the phenomena that were not fully implemented in the ERG. During the first instructional session, the FFTB tool was demoed using the ERG. During this stage, treebankers also learned how to use the Linguistic Type Database, introduced in Section 4.3. Accessing the grammar documentation along with previously treebanked sentences is an essential part of the treebanking process.

The instructional set used during the training phase of the treebank contained 500 sentences from NTUCLE-X – spanning approximately 21 student assignments. Sentences were treebanked sequentially, ensuring that treebankers had access to the context surrounding each sentence. This instructional set of 500 sentences was split into four subsets (0-a through 0-d), as shown in Table 7.1. All subsets were tagged by all five main treebankers (A-E). However, the first subset (0-a), was also tagged by Dr. Flickinger.

		Treebankers					
Datasets		**A**	**B**	**C**	**D**	**E**	**DPF**
Subset 0-a	(216 sents)	216	216	216	216	216	216
Subset 0-b	(95 sents)	95	95	95	95	95	
Subset 0-c	(95 sents)	95	95	95	95	95	
Subset 0-d	(94 sents)	94	94	94	94	94	
Total	(500 sents)	500	500	500	500	500	216

Treebankers A through D were undergraduate students. Tagger E was a graduate student. DPF stands for Dr. Daniel Paul Flickinger, the main developer of the ERG.

Table 7.1: NTUCLE-X Treebank - Instructional Set

The first subset was tagged as a group effort. Dr. Flickinger provided gold annotations for each tree which were used to compare to the trees produced by the students. During the span of a week, that included multiple sessions, the subset 0-a was tagged as a group. This subset was used for demoing the tools and the treebanking process, including an in-depth discussion of ERG's naming conventions, idiosyncrasies and limitations. Dr. Flickinger's contribution was limited to this first subset of sentences.

Following Dr. Flickinger's contribution, the three remaining subsets (0-b, 0-c and 0-d) were used to further train the treebankers through a process of adjudication.

In all annotation tasks (i.e., not limited to treebanking) there is room for errors that need to be accounted for and, preferably, minimized as much as possible. Concerning treebanking specifically, where there is room for multiple interpretations of the same sentence, adjudication exercises can be very important to bring all treebankers to the same *wavelength* – i.e., be aware of the thought process of the other treebankers. By forcing two or more treebankers to discuss discrepancies and to come up with a single tree among them (i.e., adjudication), the annotation process can become more streamlined, and some types of discrepancies tend to dissipate as the

treebanking process continues and as treebankers are forced, over and over, to resolve similar discrepancies.

Despite this being a fairly small project, perhaps too small to achieve the best possible smoothing between treebankers, adjudication exercises were used both to train the treebankers, and to serve as a measure of expected quality of the annotation taking place.

The rest of the instructional set was tagged using a pyramid style adjudication exercise, inspired by a format of the strategy card-game *Magic: The Gathering*[2] known as Two-Headed Giant – where teams of two players are forced to work together since victory can only be achieved as a pair.

In the first stage of this exercise, each of the four undergraduate treebankers were asked to adjudicate each of the three remaining subsets with a different treebanker (hence exhausting all pair-wise combinations between treebankers). Table 7.2 shows how the three subsets were used to pair all four treebankers. For example, treebanker A adjudicated with treebanker B for the first subset (0-b), with treebanker C for the second (0-c), and with treebanker D for the third (0-d).

All subsets were tagged individually first. Then, differences between trees were computed automatically. In each adjudication meeting, both treebankers had to go over sentences where there was at least one difference between their trees. For each sentence with at least one difference, treebankers had to either agree on a common tree, or agree that a suitable tree was not available.

This not only allowed all treebankers to get to know each-other, but also to learn more about the grammar and to assimilate the treebanking guidelines. Most of these meetings were supervised by me (a few were supervised by the experienced graduate student, also referred as treebanker E). The supervision ensured that all discussions were productive and instructional – which was especially important when the disagreement came from the lack of understanding of certain rules or phenomena. During these discussions, treebankers were also often reminded to follow the outlined guidelines for this treebank. For example, when context alone was not enough to not clear up ambiguity – see example (113) –, treebankers should choose to attach

[2]https://magic.wizards.com/en/game-info/gameplay/formats/two-headed-giant

prepositional phrases to higher nodes, such as main verbs, instead of lower – i.e., prefer trees like (112), instead of (111).

Subset 0-a	No Adjudication
Subset 0-b	Treebankers: A vs B Treebankers: C vs D Treebankers: (A + B) vs (C + D)
Subset 0-c	Treebankers: A vs C Treebankers: B vs D Treebankers: (A + C) vs (B + D)
Subset 0-d	Treebankers: A vs D Treebankers: B vs C Treebankers: (A + D) vs (B + C)

Table 7.2: NTUCLE-X Treebank - '2-Headed Giant' Adjudication of the Instructional Set

The second stage of this adjudication exercise was only possible because each subset of sentences had been tagged by all four treebankers. In this stage, as depicted in Table 7.2, treebankers were paired together with the person they adjudicated each subset during the first stage, and made to adjudicate each subset of sentences one more time, except this time this was done in pairs. Table 7.2 also shows how this was done. Let us look at an example of how this was processed: during the first stage, treebanker A adjudicated the subset 0-b with treebanker B, and treebanker C adjudicated the same subset with treebanker D; in the second stage, treebankers A and B formed a team and treebankers C and D formed another team; the results of the first stage of adjudication were compared between both teams, and the goal of this second stage was to, once again, go over sentences where there were any discrepancies between both teams. In other words, stage two was an adjudication of the data produced in the first stage of adjudication.

The two stage nature of this exercise was known to all treebankers from the start, and was done to ensure that treebankers did not simplify the adjudication task to the selection of any one tree. Due to the existence of a second stage of adjudication, treebankers had an extra incentive to find the best possible tree for each sentence, since not getting the correct tree would most likely mean they would have to discuss it again in the second stage.

Despite being a nice design for training purposes, adjudicating a full treebank in this two-stage fashion would be expensive and time consuming. Having four treebankers per sentence is often not feasible for larger corpora. As such, this two-stage approach was mainly conceptualized for training purposes. The remainder of the treebanking process used adjudication in a more

traditional way – to measure a form of inter-treebanker agreement, and to ensure treebankers kept following the guidelines throughout the process.

Table 7.3 shows a summary of the entire treebanking process. As already mentioned above, a total of 4,900 sentences were tagged. The first dataset (ID 0) was the largest, with 500 sentences, and was used as the instructional set. All other sets had 200 sentences each. The remaining 22 sets were tagged by either one or two treebankers. In total, 1,700 sentences ($\approx 35\%$) were tagged by two treebankers. Whenever a set was treebanked by two people, it was also adjudicated before moving on with further sets.

Table 7.3 shows that five treebankers were involved – including treebanker E, an experienced graduate student. This was necessary because two of the four treebankers (treebankers A and C) were unable to commit for the full length of the project. Despite not having actively participated in the two-stage adjudication training, Treebanker E was already an experienced treebanker, and had also participated in many of the adjudication meetings. It was important for the treebanking process to guarantee that adjudication sessions happened at various stages of the process, to ensure that quality remained stable throughout the entire process.

Measuring the Quality of the Treebank

Evaluating the quality of the treebank process is not a simple endeavor, especially since there is no gold standard to measure against. One could argue that a human annotated treebank is, in itself, a gold standard for future attempts to automatically select the best parse/tree for a give sentence. However, as it was discussed above, the task of treebanking includes common-sense reasoning, as well as true ambiguities that prevents this task from being treated as having a single correct answer. In most cases, the task is picking the *best possible analysis* from a pool of plausible analyses. As such, it is difficult to discuss quality in a very explicit way.

The common practice to measure the quality of a treebank relies on portions of the treebank that have been tagged by more than one person. The same metrics used by computational parsers are applied to double annotated subsets of the corpus, producing a measure of how much two treebankers' choices overlap, without necessarily defining which one is the correct tree. This was the main reason why roughly 35% of this treebank was tagged by two people and distributed

Datasets		Treebankers					
ID	**Size**	**A**	**B**	**C**	**D**	**E**	**Overlap**
0	500	500	500	500	500	500	500
1	200	200	200				200
2	200			200	200		200
3	200	200					
4	200		200				
5	200					200	
6	200				200		
7	200	200					
8	200		200				
9	200					200	
10	200				200		
11	200	200				200	200
12	200		200		200		200
13	200					200	
14	200		200				
15	200					200	
16	200				200		
17	200					200	
18	200		200				
19	200					200	
20	200				200		
21	200				200	200	200
22	200		200			200	200
Total	4900	1300	1900	700	1900	2300	1700

Table 7.3: NTUCLE-X Treebank - Summary

along the treebanking process (instead of overlapping only at the beginning or at the end).

The metric used to measure the overlap between two treebankers is derived from the standard PARSEVAL metric, with its first canonical algorithm proposed in Black et al. (1991). Plainly put, PARSEVAL is useful for constituency-based parsers, and is able to calculate how much the constituents defined by two different parse trees overlap.

The implementation used for this thesis follows Collins (1997) in the definition of Labeled Precision, where a constituent is only deemed equivalent if: a) it spans over the same set of words in the sentence; and b) has the same label. In addition to this Labeled Precision metric, an additional metric is provided where a constituent only needs to span over the same set of

words in the sentence to be considered equivalent (i.e., the label may or may not match) – this is referred as Unlabeled Precision.

When used to evaluate a computational parser, one of these trees is deemed as canonical ('gold' or 'target') parse, and the tree produced by the parser is evaluated against that tree.

In its canonical form, PARSEVAL precision (labeled or unlabeled) is calculated using the formula below. For labeled precision, matching constituents need to match both in yield (i.e., span of words contained by the constituent) and the constituent label. For unlabeled precision, only the yield is relevant.

$$\text{Precision} = \frac{\text{No. of generated constituents that also exist in the GOLD tree}}{\text{Number of constituents in the generated tree}}$$

Precision essentially measures the percentage of constituents in the generated tree that exist in the gold standard. But another important and related metric is the recall (also labeled or unlabeled) – which can be calculated using the formula below.

$$\text{Recall} = \frac{\text{No. of generated constituents that also exist in the GOLD tree}}{\text{Number of constituents in the GOLD tree}}$$

Recall can be seen as a measure of completeness or, in other words, what is the percentage of the gold tree that is matched by constituents in the generated tree.

However, when no tree is canonical – as is the case for treebank adjudication – this algorithm needs to be slightly modified, so it is not biased to any particular tree. Agreement is, essentially, a measure between precision and recall. The used formula was:

$$\text{Labeled Agreement (LA)} = \frac{2 \times \text{Number of labeled constituents equivalent between both trees}}{\text{Number of the sum of constituents in both trees}}$$

This formula ensures that if two trees are exactly the same, then the denominator (Number of the sum of constituents in both trees) is exactly the same as the double of the numerator (Number of labeled constituents equivalent between both trees) – yielding a score of 1. If there is no labeled constituents considered equivalent in both trees, then it produces a score of 0. And

partial overlap of the two trees will yield a score between 0 and 1, proportional to the amount of overlap, but not biased by any of the two trees.

The formula for unlabeled precision is very similar, differing only in the method to define two constituents as equivalent. The used formula was:

$$\text{Unlabeled Agreement (UA)} = \frac{2 \times \text{Number of unlabeled constituents equivalent between both trees}}{\text{Number of the sum of constituents in both trees}}$$

Scores for these two metrics were computed for each sentence before being adjudicated and then averaged across all sentences in a given set. The results can be seen in Table 7.4. The unlabeled precision is, naturally, slightly higher than the labeled precision (by roughly 5%). It is possible to observe a slight tendency to increase the overlap in later sets, which shows that treebanking does have a learning curve. An average agreement score of 73.1% for labeled agreement and 78% for unlabeled agreement is in line with what was expected. These numbers are comparable to those provided by Tanaka et al. (2005) – reporting 83.5% for a similar metric of labeled agreement across annotators, when building a treebank with a fairly large HPSG grammar of Japanese. In comparison, the ERG is a much larger grammar, capable of generating a lot of ambiguity. The NTUCLE is also a corpus with a lot of convoluted sentences. Many of these sentences are also often long and difficult to disambiguate, even following a set of guidelines. This helps explain why these are fairly good results, even if slightly lower than those presented by Tanaka et al. (2005).

In total, 76.3% of the 4,900 sentenced in the English portion of the Tembusu treebank were able to get a suitable parse (i.e., 23.7% of all sentences were rejected).

7.1.2 Treebanking Mandarin Chinese Learner and Educational Data

The Mandarin Chinese portion of the Tembusu Treebank currently includes 5,648 sentences from two corpora: the NTU Corpus of Learner Mandarin and the Mandarin Education Corpus, both introduced in Chapter 5.

Many of the issues, processes and assumptions discussed above, for English, also hold true

ID	Size	Overlap					LA	UA
0	500	**A**	**B**	**C**	**D**	**E**	0.681	0.747
1	200	**A**	**B**				0.738	0.778
2	200			**C**	**D**		0.690	0.730
11	200	**A**				**E**	0.773	0.812
12	200		**B**		**D**		0.773	0.820
21	200				**D**	**E**	0.775	0.816
22	200		**B**			**E**	0.761	0.807
	1,700						0.731	0.780

Table 7.4: NTUCLE-X Treebank - Agreement of Overlapped Sets

for this portion of the treebank. However there are some key differences.

The Mandarin Chinese portion of the treebank was developed with the help of five undergraduate student assistants, whose proficiency level of Mandarin Chinese was self-reported to be very high. In general, Singaporean students tend to underestimate their Mandarin Chinese proficiency, and prefer working with English data. Students willing to work with Mandarin Chinese are often quite reliable users of this language.

All five students had also completed a Syntactic Theory course (introducing HPSG) with good grades. The students were introduced to the same tools as the English treebankers, and went through a training exercise – although admittedly a much simpler exercise than the one conducted for English. This was considered enough given that ZHONG is far less complex than the ERG. The training exercise contained around 50 sentences (both grammatical and ungrammatical), which were tagged as a group effort while showcasing ZHONG's main design decisions. Students were also free to ask questions during the treebanking process, which was sometimes necessary since the grammar remained in active development during the treebanking process, and students had to keep adapting to new treebanking decisions.

Each student treebanked between 2,200 and 2,800 sentences. Most sentences (circa 89.4%) were tagged by two or more students. Table 7.5 shows the summary of this treebanking process, including the distribution of sentences per student and how many students tagged each set. Table 7.5 also shows the Labeled Agreement (LA) and Unlabeled Agreement (UA) scores, following the same definitions provided above, for English.

Pair-wise agreement scores for sets tagged by three students can be found in Tables 7.6 and 7.7. The results reported in Table 7.5 were calculated by averaging the pair-wise agreement scores among all annotators of a particular set.

ID	Size	Overlap					LA	UA
tufs_cmn_01	200	**A**	**B**				0.870	0.897
tufs_cmn_02	200			**C**	**D**	**E**	0.795	0.840
tufs_cmn_03	200	**A**	**B**			**E**	0.880	0.905
tufs_cmn_04	200			**C**	**D**		0.817	0.848
tufs_cmn_05	200			**C**	**D**	**E**	0.839	0.900
tufs_cmn_06	200	**A**	**B**				0.877	0.928
tufs_cmn_07	200			**C**	**D**		0.839	0.867
tufs_cmn_08	137	**A**	**B**			**E**	0.874	0.892
cmnedu_01	200	**A**	**B**			**E**	0.824	0.873
cmnedu_02	200			**C**	**D**		0.779	0.820
cmnedu_03	200	**A**	**B**			**E**	0.851	0.884
cmnedu_04	198			**C**	**D**		0.801	0.834
hsksc_01	175	**A**	**B**			**E**	0.832	0.882
hsksc_02	200			**C**	**D**		0.775	0.832
hsksc_03	81	**A**	**B**			**E**	0.691	0.736
hsksc_04	200			**C**	**D**		0.791	0.826
hsksc_05	200	**A**	**B**			**E**	0.788	0.813
hsksc_06	157			**C**	**D**		0.767	0.794
ntuclm_test_01	200	**A**	**B**			**E**	0.794	0.817
ntuclm_test_02	87			**C**	**D**		0.624	0.642
ntuclm_train_01	200			**C**			-	-
ntuclm_train_02	200	**A**	**B**			**E**	0.874	0.900
ntuclm_train_03	200			**C**			-	-
ntuclm_train_04	200	**A**	**B**			**E**	0.871	0.897
ntuclm_train_05	200			**C**			-	-
ntuclm_train_06	200	**A**	**B**			**E**	0.884	0.912
ntuclm_train_07	200			**C**	**D**		0.808	0.832
ntuclm_train_08	200	**A**	**B**			**E**	0.859	0.885
ntuclm_train_09	200			**C**	**D**		0.533	0.543
ntuclm_train_10	213	**A**	**B**			**E**	0.721	0.733
Total	**5,648**	**2,806**	**2,806**	**2,842**	**2,242**	**2,806**	**0.808**	**0.893**

Table 7.5: MEC and NTUCLM Treebanks - Agreement Summary

The average LA score was around 80.8%, and the average UA score was around 89.3%. These scores are quite good, and slightly higher than what was achieved for English. This difference in agreement scores is expected as ZHONG is far less complex than the ERG and the treebanked sentences are much simpler than those found in the NTUCLE.

ID	Overlap		LA	UA
tufs_cmn_02	**C**	**D**	0.779	0.830
tufs_cmn_02	**C**	**E**	0.826	0.865
tufs_cmn_02	**D**	**E**	0.780	0.826
tufs_cmn_03	**A**	**B**	0.913	0.936
tufs_cmn_03	**A**	**E**	0.869	0.894
tufs_cmn_03	**B**	**E**	0.859	0.885
tufs_cmn_05	**C**	**D**	0.879	0.938
tufs_cmn_05	**C**	**E**	0.819	0.884
tufs_cmn_05	**D**	**E**	0.819	0.877
tufs_cmn_08	**A**	**B**	0.914	0.930
tufs_cmn_08	**A**	**E**	0.844	0.863
tufs_cmn_08	**B**	**E**	0.863	0.883
cmnedu_01	**A**	**B**	0.902	0.953
cmnedu_01	**A**	**E**	0.770	0.817
cmnedu_01	**B**	**E**	0.800	0.848
cmnedu_03	**A**	**B**	0.911	0.949
cmnedu_03	**A**	**E**	0.818	0.852
cmnedu_03	**B**	**E**	0.824	0.850
hsksc_01	**A**	**B**	0.828	0.871
hsksc_01	**A**	**E**	0.869	0.922
hsksc_01	**B**	**E**	0.798	0.853
hsksc_03	**A**	**B**	0.720	0.748
hsksc_03	**A**	**E**	0.680	0.734
hsksc_03	**B**	**E**	0.675	0.727
hsksc_05	**A**	**B**	0.805	0.827
hsksc_05	**A**	**E**	0.767	0.792
hsksc_05	**B**	**E**	0.792	0.821

Table 7.6: MEC Treebanks - Agreement of Overlapped Sets

It is, however, important to note that certain sets show a much lower score than the average (e.g., ntuclm_test_02, ntuclm_train_09 and ntuclm_train_10). This can be explained by the fact that the NTUCLM includes both grammatical and ungrammatical sentences, and most problematic sentences had been clustered in these three sets. It became clear, through this data, that the treebankers had some difficulty tagging problematic sentences.

According to the guidelines provided, treebankers should only accept ungrammatical sentences when a mal-rule that could explain the error was available. However, due to idiosyncrasies of the FFTB treebanking tool, the names of lexical entries are not shown during the

ID	Overlap		LA	UA
ntuclm_test_01	**A**	**B**	0.879	0.905
ntuclm_test_01	**A**	**E**	0.741	0.765
ntuclm_test_01	**B**	**E**	0.761	0.780
ntuclm_train_02	**A**	**B**	0.928	0.947
ntuclm_train_02	**A**	**E**	0.836	0.864
ntuclm_train_02	**B**	**E**	0.857	0.888
ntuclm_train_04	**A**	**B**	0.921	0.943
ntuclm_train_04	**A**	**E**	0.859	0.884
ntuclm_train_04	**B**	**E**	0.833	0.864
ntuclm_train_06	**A**	**B**	0.917	0.944
ntuclm_train_06	**A**	**E**	0.858	0.884
ntuclm_train_06	**B**	**E**	0.878	0.909
ntuclm_train_08	**A**	**B**	0.867	0.887
ntuclm_train_08	**A**	**E**	0.841	0.869
ntuclm_train_08	**B**	**E**	0.871	0.900
ntuclm_train_10	**A**	**B**	0.670	0.681
ntuclm_train_10	**A**	**E**	0.679	0.694
ntuclm_train_10	**B**	**E**	0.812	0.823

Table 7.7: NTUCLM Treebank - Agreement of Overlapped Sets

treebanking process (i.e., only the lexical type is shown). Unfortunately, some mal-rules (e.g., lexical entries for orthographic mistakes) could only be recognized as such through the name of the lexical entry, creating some confusion and often leading to the sentence being rejected.

Another source of errors were sentences with awkward semantics, e.g., the Mandarin Chinese equivalent of "I am France." or "This is the office's professor.". While these sentences most likely reveal a problem in the student's knowledge of Mandarin Chinese, they are not technically ungrammatical, and the guidelines indicated that they should have been tagged as normal. However, some students failed to understand this issue very well, rejecting many of these sentences because there were no available mal-rules to 'correct' such issues.

A key difference between the English and the Mandarin Chinese portions of this treebank was the fact that the Mandarin Chinese portion was not adjudicated between students. Since students were treebanking while ZHONG was still in active development, this meant that some sentences would need to be re-tagged when ZHONG's development ceased. This need to re-tag sentences could happen because previously rejected sentences could gain a viable parse

(e.g., through a new mal-rule), because extra ambiguity was added to the grammar and a new tree would need to be selected, because word segmentation in a sentence could change (i.e., be fixed), or even because a certain rule or lexical entry could have been renamed or deleted – which would mean that saved trees using that rule would be incompatible with the final version of the grammar.

As treebanking provides useful insight into grammar development, the normal cycle is to treebank, fix the grammar and update the treebank, often several times (Oepen et al., 2004). The students thus provided a first pass treebank, which was updated as the grammar improved.

This initial treebank was used to inform ZHONG's development through a careful analysis of the comments that were left during the tagging process. Once ZHONG reached its final version, I re-treebanked each sentence, while taking into consideration the previous trees and comments left by the previous annotators.

A summary of the final results of the Chinese portion of the Tembusu treebank can be found in Table 7.8. About 75.3% of the 5,642 sentences from the Mandarin Chinese portion of this treebank was successfully annotated. However, considering only sentences for which the grammar was able to produce a parse, then this number increases to 87%.

ID	Size	Treebanked	Parsed	Treebanked/ Parsed
tufs_cmn	1531	0.685	0.808	0.848
cmn_edu	798	0.896	0.949	0.945
hsksc_01	175	0.897	0.971	0.924
hsksc_02	200	0.665	0.860	0.773
hsksc_03	81	0.568	0.765	0.742
hsksc_04	200	0.580	0.705	0.823
hsksc_05	200	0.425	0.630	0.675
hsksc_06	157	0.357	0.586	0.609
ntuclm_train	2013	0.824	0.924	0.891
ntuclm_test	287	0.805	0.916	0.878
Total	5642	0.753	0.865	0.870

Table 7.8: Final Chinese Treebank - Summary

7.2 Mal-Rule Enhanced Parse Ranking Models

As introduced above, one of the main reasons to create the Tembusu Treebank was to gather and annotate data to train new mal-rule enhanced parse-ranking models for both the ERG and ZHONG. These models differ from existing models by one key factor: they were trained with data containing mal-rules. This should, in principle, allow these models to perform better at tasks such as parse-selection and error diagnosis when using grammars with mal-rules enabled.

One of the current issues of using grammars with mal-rules enabled but with a model not trained using mal-rules is the fact that the model does not know the relative likelihood of mal-rules when compared with the other available rules in the grammar. Mal-rules are essentially initiated with a neutral weight (neither likely nor unlikely to happen), when many other rules show up with negative weights inside the model. This effectively makes the grammar opt for parses using mal-rules even when other grammatical parses are still available. Training a parse ranking model on a treebank containing mal-rules allows the model to store the right relative weights of all rules, including mal-rules. With enough data, theoretically, this means that the grammar should be able to prefer parses without mal-rules whenever a plausible parse is available.

It is important to note, however, that unification based grammars always produce the same number of parses for the same sentence. One of the goals of treebanking data is to provide information on how to rank the available parses in order of likelihood, based on real data. As such, in the presence of an ungrammatical sentence, using a mal-rule enhanced parse-ranking model should help guess the most likely problem in that sentence, while preserving the ability to provide multiple corrections for a single sentence (if available).

With this in mind, the data in the Tembusu Treebank was used to train two separate maximum entropy parse ranking models, one for English and another one for Mandarin Chinese. These models were trained using preexisting binaries (FFMASTER and FFWORKER) released as part of the ACE Tools, introduced in Section 4.3.[3]

[3]The grandparenting level was set to 3 for both models, following what had been done for previous ERG models: that means that up to three nodes above the subtree were included when making features, see Toutanova et al. (2005).

7.3 Summary

This chapter described the creation of the Tembusu Treebank, a new multilingual treebank of English and Mandarin Chinese learner data. Employing the help of a team of students at NTU, this treebank used mal-rule enhanced versions of the ERG and ZHONG to tag a total of 7,983 sentences (4,246 sentences for Mandarin Chinese, and 3,737 sentences for English).

This treebank was used to train two new parse ranking models, which are the first HPSG-based models trained using mal-rule enhanced parsers (at least for these two languages). These new models are hypothesized to correct some of the bias of previous models, and to help the grammars perform better at tasks such as parse-selection and error diagnosis when mal-rules are enabled. These hypotheses were evaluated through a series of experiments which will be reported in Chapter 9.

With a few exceptions for Mandarin Chinese, the data contained in the Tembusu Treebank can and will be released under an open license, and should be incorporated mainstream by the respective grammar projects. The English portion of this treebank could be merged with the Redwoods treebank (Oepen et al., 2002),[4] – a treebank in development for almost 20 years, used to produce parse ranking models for the ERG, but that currently lacks ungrammatical sentences. ZHONG does not currently have a public treebank, but the open section of Mandarin Chinese data contained in the Tembusu Treebank will serve this purpose – and will be released as a companion treebank for ZHONG.

[4] `https://github.com/delph-in/docs/wiki/RedwoodsTop`

Chapter 8

iTELL: Suite of Applications

This chapter introduces iTELL – a suite of web-based applications exploiting deep computational parsers in **i**ntelligent **T**echnology-**E**nhanced **L**anguage **L**earning environments. This chapter will introduce two different applications: the LCC-APP, an online writing support system designed to help students with English Scientific Writing; and CALLIG, a collection of language learning games designed using elements of improvisation comedy.

8.1 The LCC-APP: an academic writing support system

The iTELL's LCC-APP was developed in collaboration with a team of English lecturers from the Language and Communication Centre (LCC), NTU. As a pedagogical tool, the LCC-APP aimed to fulfill three major requirements: a) a system focusing specifically on the needs of NTU engineering students in their English Scientific Writing; b) a system capable of detecting problems that go beyond grammaticality, such as stylistic guidelines recommended by LCC; and c) a system capable of detecting errors with sufficient granularity to provide meaningful corrective feedback, helping students understand the problem, to explore possible solutions, and to decide on the best corrections on their own.

The LCC-APP was inspired by, and builds on, previous work by Suppes et al. (2014) and Flickinger and Yu (2013) who have showed significant results in the use of corrective feedback produced by computational parsers to evaluate the grammatical correctness of written English. Even though their work focused on elementary-school and middle-school English language ed-

ucation, they target a very similar problem: the difficulty of correcting and providing useful feedback to large numbers of students in a timely manner.

After more than three years of collaboration with the LCC, the end result was the iTELL's LCC-APP – an open-source system exploiting mal-rules, computational parsers, and NLP techniques to perform linguistically aware error detection. This application was designed to provide the highest possible quality of corrective feedback for a range of common errors made specifically by NTU's engineering student population. A full description of this system can be found in Morgado da Costa et al. (2020).

Motivation

The motivation behind the development of the LCC-APP was to assist undergraduate engineering students in a mandatory course on English Scientific Writing at Nanyang Technological University, in Singapore. Its main goal was to alleviate some of the challenges LCC lecturers face while teaching a cohort of over 2,000 undergraduate engineering students per year. These challenges revolve around correcting and providing timely, high-quality feedback on student assignments, so students can learn from the feedback and iteratively improve their assignments throughout the duration of the course. Unfortunately, due to the size of the student cohort and the number of available lecturers, this is often a very difficult task. Without an automated system, like the LCC-APP, students rarely receive feedback on their drafts before their final submissions, leaving them with little incentive to work through and incorporate the tutors' feedback.

A possible solution to this problem was found in the development of a writing support system capable of detecting common errors and of providing individual corrective feedback to students throughout the writing process. The LCC-APP is the culmination of these efforts.

The LCC-APP was designed to be a pedagogical tool. As such, the system's objective was not to correct errors. Instead, the goal was to identify issues and provide constructive feedback that prompts students to consider whether corrections are needed. This should, in principle, allow students to have a more meaningful participation in the error correction process, learning while actively identifying errors and choosing from multiple ways which are often available to correct different classes of problems. This is aligned with the ideas behind indirect corrective

feedback, discussed in Section 2.2.

System Architecture

The LCC-APP is fully developed on top of existing open-source platforms. At its core, it is a web system developed using Python, Flask and SQL. It was designed to be easy to customize, making it easy to add or block error checks. It is hosted on Github[1] under the iTELL's repository, and it is developed under an MIT open-source license.

The LCC-APP functionalities can only be accessed through a password protected user authenticated module. This was done to prevent misuse of the app, because the application manipulates and stores sensitive student data. During this PhD, the LCC-APP was used solely by NTU students and only while enrolled in a communication course at LCC, NTU.

Submission and Preprocessing

Students can interact with the LCC-APP in two ways: by submitting a document (which was, by far, the most used method), or by submitting individual sentences.

Currently, document submission is limited to documents using the *docx* format. This was a design decision based on the quality of text extraction possible when using *docx*, when compared with other formats. Since sentence boundaries are obviously extremely important in error detection, extracting text from formats like *pdf* would hurt the system's performance immensely and, therefore, be less useful to students. The *docx* format is non-proprietary and can be produced from most word processors – including Microsoft Word, Pages, as well as free document processors such as Google Documents and LibreOffice Writer.

Figure 8.1 shows the document uploading page of the LCC-APP. This figure also shows an important decision students had to make before uploading their documents. Following the approved IRB protocols for this research project, every time students uploaded a document to the LCC-APP they were forced to decide on whether or not to make their document available for further research through its release under a Creative Commons 0 license.[2] Whenever students

[1] `https://github.com/lmorgadodacosta/itell`
[2] `https://creativecommons.org/share-your-work/public-domain/cc0/`

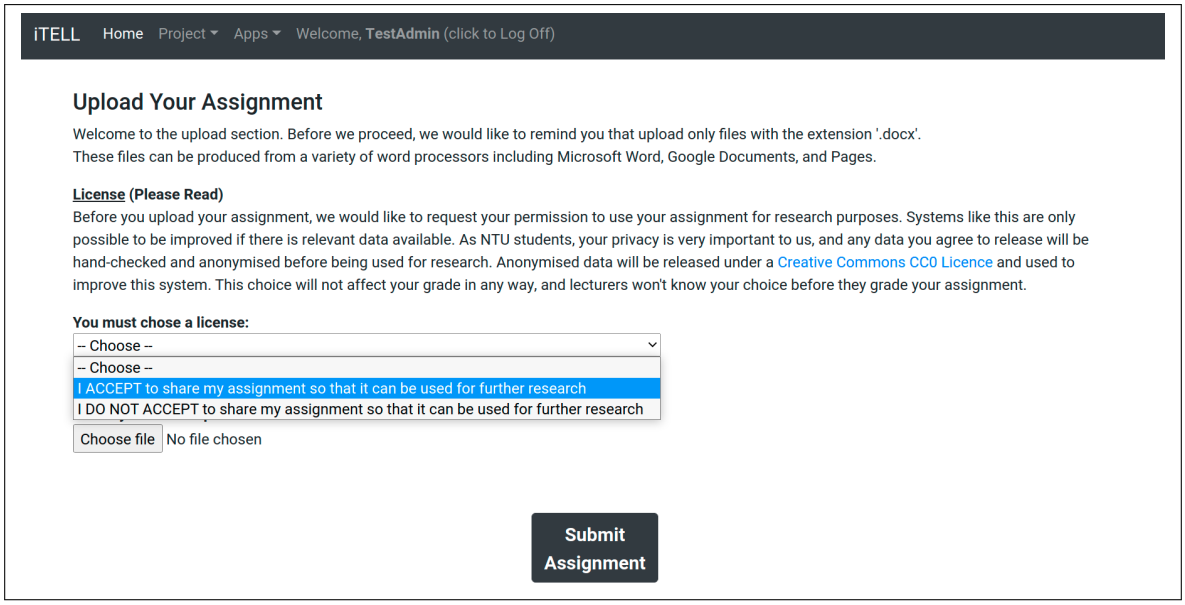


Figure 8.1: Online Error Detection System - Document Upload

agreed to release their assignment, this assignment was anonymized and added to the NTUCLE, already discussed in Section 5.2.

The preference for this very liberal license was to safeguard students' privacy when adding these documents to the NTUCLE. Since all assignments are completely anonymized before being released to the public, using a license that requires attribution would effectively contradict the students' right to anonymity. The most important concern was to allow students to protect their data. This was done through forcing an informed decision, that guaranteed their anonymity and ensured that they would not be adversely affected should they choose not to share their data. In the end, between 40% and 50% of students chose to release their assignments.

After a document is submitted, the file is converted to *xml* using Python's Mammoth[3] library. Sentence segmentation is done using the Python Natural Language Toolkit (Bird et al., 2009).

The final step before using the ERG to parse the documents is a stage of content detection. In this stage, a large list of hand-written rules attempt to define which parts of the document deserve being checked. This step was continuously improved throughout the development of the system, and was not available in early versions. The lack of this content detection stage in early versions was the source of many false positives, where the system flagged parts of the document (e.g.,

[3]https://pypi.org/project/mammoth/

content inside tables, figure legends, etc.) as errors when they were not. A similar problem came from mid-sentence citations. Given the nature of the documents being processed, mid-sentence citations were extremely common, but this turned out to be a challenge for ERG. To solve this problem, the LCC-APP is now able to ignore mid-sentence citations before sending a sentence to be processed by the ERG. In addition, the LCC-APP also relies on the expected structure of the document (such as section headers) to ignore parts of the document that are not expected to be full sentences – e.g., tables, figure legends, bibliography (etc.). An example of the results of this stage is shown in Figure 8.2, where the ignored portions of the document are shown in light gray. The process of ignoring mid-sentence citations happens behind the scenes, and is imperceptible in this document view.

The alternative method of interaction with the LCC-APP is through a mode where users can submit a single sentence for inspection. Figure 8.3 shows this mode of interaction. This differs from the previous mode in two essential ways: i) only a single sentence is expected as input and ii) it must be released under a CC0 license. Users are adverted of this license restriction, and encouraged to use the document submission module if they rather not share their data.

Once the system gathers a list of sentences to be checked (i.e., a list with a single sentence if the second mode of interaction was used), sentences are then processed against a list of grammatical and stylistic checks.

Grammatical and Stylistic Checks: a two-grammar approach

The LCC-APP currently performs 80 grammatical and stylistic checks. A core part of this system uses mal-rules that already existed within the English Resource Grammar, designed to identify both ungrammatical and stylistically deprecated sentences (Bender et al., 2004; Flickinger and Yu, 2013; Suppes et al., 2014).

Using the ERG allowed the LCC-APP to have a detailed granularity in error detection, capable of providing very specific feedback. While data-driven systems usually group broad classes of errors together (e.g., “problems with determiners”), the level of linguistic detail of the ERG allows it to differentiate different problems associated with a certain class of errors by the specific linguistic rule it violates (e.g., the omission of an article for single countable nouns; the

Figure 8.2: LCC-APP: awareness of document structure

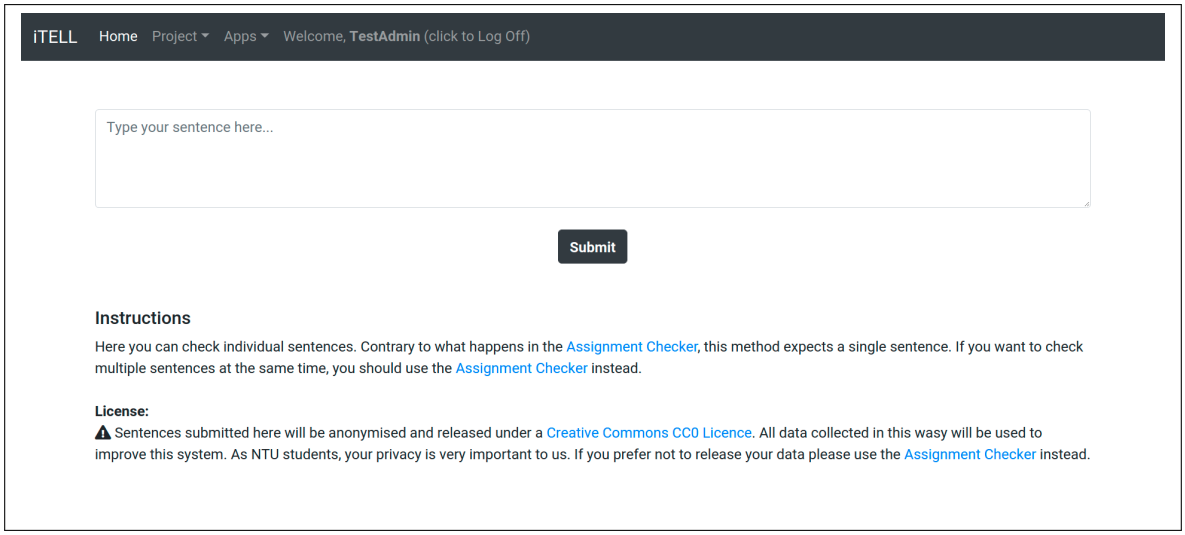


Figure 8.3: Online Error Detection System - Single Sentence Submission

use of indefinite articles with uncountable nouns; or the use of the wrong form of the indefinite article 'a/an', etc.)

The LCC-APP currently uses 70 of over 250 mal-rules available in the ERG. These mal-rules were carefully selected using the results of annotating the NTU Corpus of Learner English (Winder et al., 2017), presented in Section 5.2.

In addition to mal-rule checks, the LCC-APP can also detect 10 broad classes of stylistic errors using Natural Language Processing techniques, focused mainly on enforcing guidelines provided by LCC lecturers. These include, for example, checking the sentence length, the use of contractions, the use of overly formal or overly casual words, the use of rhetorical questions, and proper capitalization of words. This does not mean, however, that mal-rules are not able to detect errors that could be defined as *stylistic*. In fact, four out of the 70 mal-rules included in the LCC-APP could be considered as stylistic since they deal mostly with problems related to punctuation (i.e., run-on sentences, commas splitting the subject and the predicate without intervening adverbial phrases, the use of commas without being followed by a space, and the use of question marks in sentences that are not in question form). Mal-rules can be used to check anything that is analyzed as a linguistic token. In the case of the ERG, punctuation is analyzed as any other word, allowing this grammar to make judgments on the appropriate use of punctuation.

While many of these checks would most likely be useful beyond the context of NTU classrooms, some of them are definitely specific to either the population represented by NTU students, or to the sensitivities of the lecturers that helped guide the development of this system. Some errors that might be specific to NTU's student population include, for example, the overuse of words like *stuff* and *aforementioned*. While errors particular to individual lecturers' sensitivities may include the use of first person pronouns (which LCC students are told to avoid), or the use of words such as *tackle* or *hassle*, which are classified as overly casual by LCC lecturers but which might not hurt the sensitivities of lecturers elsewhere. If ever used in a different context, each check can be easily enabled or disabled from the LCC-APP.

There were two main reasons to use only a select number of mal-rules available in the ERG. The first of these reasons was that each mal-rule was linked to specific corrective feedback. The design of corrective feedback messages took a considerable amount of time. Not only were multiple LCC lecturers involved in the process, but it was also not always clear what would be the best form of a particular feedback message. This topic will also be discussed in light of the results presented in Chapter 9.

The second reason why only a a few mal-rules were selected for the LCC-APP was the fact that even though the ERG had had substantial work on mal-rule development, it did not include a parse-ranking model trained on mal-rules. This has been discussed in Chapter 7 as one of the motivations to build such model as part of this thesis.

Ambiguity in mal-rule selection is a real and very interesting problem. This was actually discussed as an advantage of using mal-rules over statistical methods. As discussed in Section 2.2, mal-rules are often able to reconstruct multiple corrections for a single ungrammatical sentence. This also means that a sentence may have multiple possible diagnoses to choose from, depending on its intended meaning. In previous applications using the ERG, such as the work presented in Suppes et al. (2014), the ambiguity generated by multiple corrections for a single ungrammatical sentence was greatly reduced by limiting the vocabulary available in the grammar. And while this may be a viable choice for elementary-school and middle-school English language education, it is not a viable choice at the university level – where unrestricted access to vocabulary is necessary.

However, when the vocabulary is unrestricted, the number of corrections can be very high – and many of these corrections may seem completely implausible at first glance, which is often explained by ambiguity at the word level (e.g., multiple parts-of-speech for a single word, multiple verb frames, etc.). In addition to implausible corrections, without a mal-rule enhanced parse-ranking model, parses using mal-rules are often ranked higher than other parses that predict the sentence is grammatical – which is a problem that will be shown and discussed in greater detail in Chapter 9.

Unfortunately, building such a model takes a considerable amount of time, and this model was not available for the most part of my PhD – which meant that the LCC-APP had to make use of different techniques to reduce the rate of misdiagnoses. The main technique was employing a two-grammar approach, which will be discussed below. The second technique was to select only mal-rules that had shown a high chance of being correctly selected when parsing an ungrammatical sentence.

This evaluation used the NTUCLE to first determine which mal-rules were being used when parsing this corpus. Mal-rules that did not appear while parsing the NTUCLE were not considered for the LCC-APP. A second step of this evaluation that led to further filtering was an informal analysis of the precision of each individual mal-rule. For each mal-rule that appeared while parsing the NTUCLE, up to 100 sentences were randomly selected (i.e., some mal-rules appeared less than 100 times in the corpus). These sentences were hand-checked to see if each particular mal-rule was able to provide a reasonable diagnosis for that particular sentence. This provided an approximate measure of precision of how the grammar was using each mal-rule. Because the English Resource Grammar lacked a parse-ranking model with mal-rules, many mal-rules were too often incorrectly used. In the end, rules with a precision below 70-80% were excluded, leaving only 70 mal-rules. The time it took to develop corrective feedback messages for each mal-rules was also a factor to decide how many mal-rules were kept.

The two-grammar approach consists of parsing each sentence through a pipeline with two different versions of the ERG: the first is the standard release of the ERG, that should only be able to parse sentences considered 'proper' English; and the second is the mal-rule enhanced version of the same grammar, which can use mal-rules to parse ungrammatical sentences.

Sentences are processed serially in the LCC-APP. For each sentence, the system first tries to use the standard release of the ERG. If it succeeds in obtaining a parse for that sentence, then the sentence is considered grammatical (and it does not get parsed by the mal-rule enhanced version of the grammar). But when the standard release of the ERG fails to produce a parse for a sentence, the mal-rule enhanced version is used to try to get a parse with mal-rules. It is also possible that the mal-rule enhanced parser is not able to parse the sentence, meaning that the sentence is likely ungrammatical but that there are no mal-rules specifically available to catch the error(s) contained in the sentence.

This parsing step is done using PyDelphin which presents an easy-to-use python API for ACE, both introduced in Chapter 4. The main reason sentences are processed serially is the fact that parsing with the ERG can be very computationally demanding. Currently, the LCC-APP limits the memory used by each sentence to 3Gb. This means that a server should have 3Gb of ram available for each instance of LCC-APP running (i.e., for each student uploading a document at the same time). This, of course, comes with a waiting cost to the students, who usually need to wait a few minutes before receiving feedback on their assignment. Should computational resources not be an issue, it would not be too difficult to allow the LCC-APP to use more resources to parse a document faster.

After going through the two-grammar step, each sentence also goes through the collection of NLP checks designed to capture other stylistic problems. The number of problems associated with a sentence depends on the number of mal-rules contained in the final parse and the number of stylistic checks it failed – making it not at all unusual for a sentence to have more than one problem associated with it.

Automatic Corrective Feedback

After sentences go through the grammatical and stylistic checks, the system provides a corrective feedback message for each error found in each submitted sentence.

The corrective feedback messages were designed in collaboration with the team of LCC lecturers. Since the system was to be used in their classrooms, LCC lecturers felt strongly about taking ownership of the design of the feedback messages.

Following the discussion introduced in Section 2.2, it is generally true that, in the context of second language writing, indirect corrective feedback is preferred to more direct versions. And this was also the position of the LCC lecturers involved in the development of the LCC-APP.

However, while this indirect/meta-linguistic corrective feedback is made possible through the fine-grained linguistic-based analysis behind the system's error detection algorithms, this was not not a simple task. One of the hardest problems encountered while trying to correct a sentence is, once again, ambiguity. Even though the ERG is able to produce multiple sets of corrective feedback (i.e., multiple corrections of the same sentence), displaying too many possible corrections of a sentence would, most likely, be overwhelming to students. This ambiguity in the correction process is, in fact, raised by Ashwell (2000) as a motivation to use indirect corrective feedback. Ashwell states that indirect corrective feedback is preferred over more direct methods because it is relatively easy to identify errors but much harder to know how to correct them. Similar findings are presented by Lee (2004) where it is discussed that human lecturers are also prone to make assumptions leading to spurious and even erroneous corrections. In Lee (2004)'s study, it is reported that only about half of teacher's corrections were accurate – labeling 40% unnecessary and up to 3% inaccurate/erroneous.

As it was discussed above, since the ERG did not have a mal-rule enhanced parse-ranking model, the LCC-APP's performance in choosing the most likely correction was also far from perfect. This had an impact in the design of the corrective feedback messages – i.e., they had to be explicitly non-committal.

The LCC lecturers were also happy with enforcing that the correction process required the judgment of students, including whether a corrective feedback message needed to be addressed or not. A good example of sentences which might not require changes would be sentences flagged as 'overly long'. Such sentences should, in principle, be checked (and possibly rewritten) but since the check is merely making a decision based on the number of words in the sentence, this may not be a serious problem for some sentences. This decision should ultimately be made by the student, who needs to make sure a sentence reads well regardless of its length.

For the time being, the LCC-APP uses the original parse-ranking model provided by the ERG (i.e., a model without mal-rules) to choose the most likely correction out from all available

corrections for a sentence. Even though sometimes this diagnosis is wrong (i.e., it does not contain the student's intended meaning), it is still enough to help students locate the error in the sentence. In other words, even if the feedback suggests an unlikely correction for an error, the fact that the system flags a problematic sentence that needs to be revised is often enough, since students are able to identify many classes of errors on their own when their attention is directed.

This line of thought was also the main motivation to include a generic unspecified error message when the system detects a problem but no specific mal-rule had been designed for that particular error. This happens only when both the standard release of the ERG and the mal-rule enhanced version of the ERG fail to produce a parse for a given sentence. Being rejected by both grammars is a strong indicator that a sentence is not linguistically sound without actually knowing what error(s) it contains. Following what was said above, it was assumed that this unspecified error label can still be useful since students are often able to identify errors independently just by being alerted to the possibility of their occurrence. Students were aware of the fact that the system was still in development, and they could always choose to discuss a diagnosis with a lecturer or peer, which acted as a safety net to occasional misdiagnoses.

Whenever possible, corrective feedback messages include a placeholder that is replaced by the word or words in the vicinity of the errors detected in order to help the students identify the location of the error. This is essentially the constituent defined by a mal-rule.

Here are some examples of the original corrective feedback messages used by the LCC-APP:

- *This sentence may have a verb which does not **agree** in person (e.g., 'I', 'you', 's/he') and number (singular/plural) with its subject: {{placeholder}}. Please check the sentence and ensure that the verb agrees with its subject.*
- *You may be using 'a' (an indefinite article) before something that cannot be counted and does not have a plural form (an uncountable noun such as 'research'): {{placeholder}}. Please check your sentence for uncountable nouns and remove any 'a' that comes before them.*
- *This sentence may contain **subjective or informal words** or expressions: {{placeholder}}. You may want to replace these words and expressions with more formal and objective alternatives.*
- *This sentence is much **longer than the average sentence**. It may be difficult for readers to read the sentence and understand it after reading it once. There is also a higher risk of making grammar*

mistakes in such a long sentence. You may want to consider breaking up the sentence to make it easier for the reader to follow the text.

- *You have used **'there'** in this sentence. Please check if it should be **'their'** instead and make the change if necessary.*

Finally, Figure 8.4 shows the end result produced by the system. The student's submitted document is converted to HTML (preserving images and most of the original styling of the document), and sentences that were considered problematic are highlighted in either red or yellow. These two colors are used to mark different levels of confidence and severity of the identified problems. For example, using overly casual or overly formal words or expressions triggers a yellow warning (since this is mostly a question of style and, in most contexts, not a very serious problem). In contrast, the lack of agreement between subject and predicate triggers a red level warning – most definitely requiring the student to change the sentence in some way.

When the single sentence mode is used, the same feedback is provided but delivered in a slightly different way – see Figure 8.5.

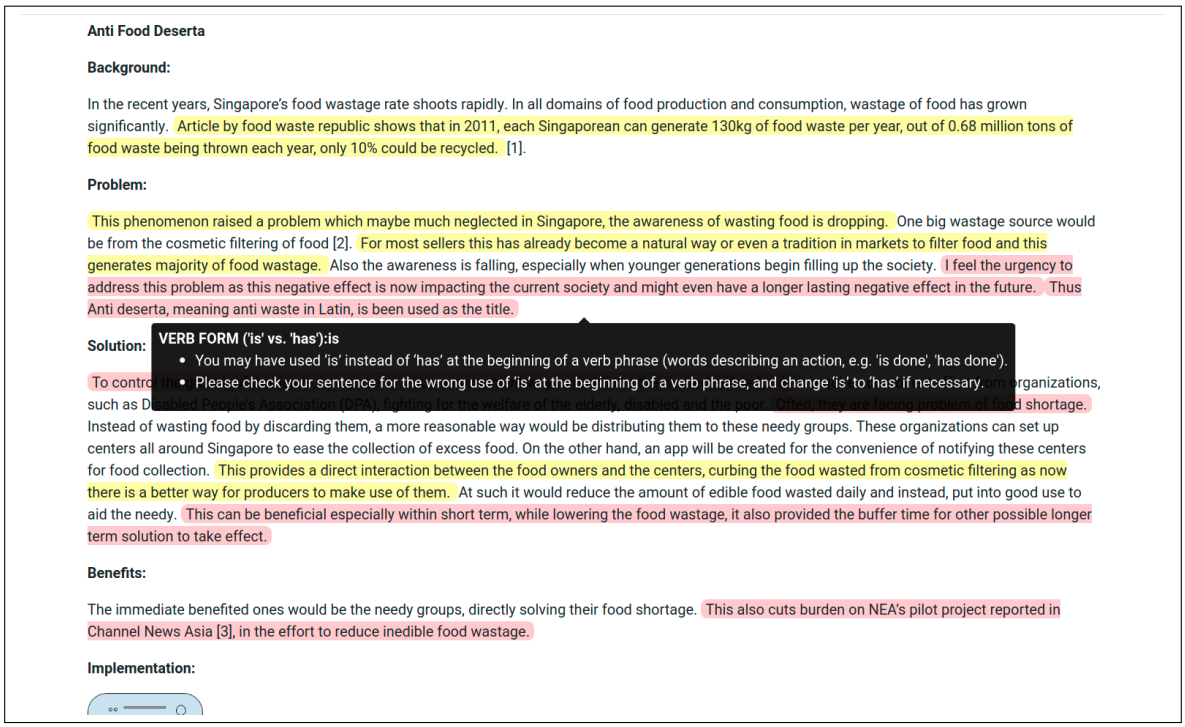


Figure 8.4: Online Error Detection System - Feedback

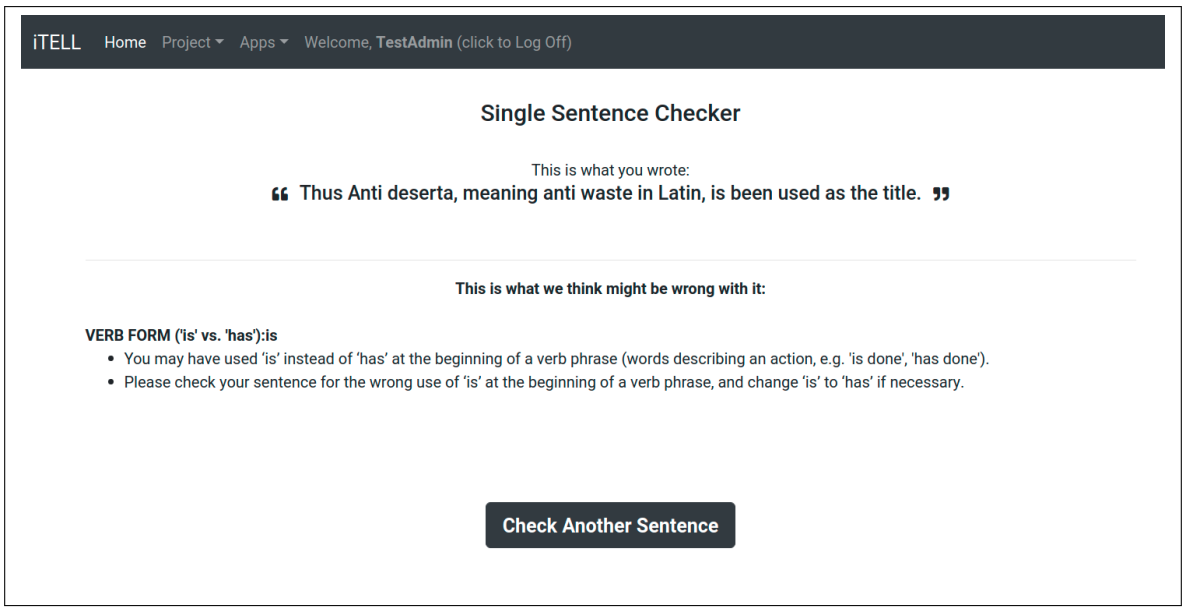


Figure 8.5: Online Error Detection System - Single Sentence Feedback

Discussion and Future Work

The LCC-APP is a system designed to help NTU students improve the writing quality of their assignments through automated corrective feedback. And, to this end, as it will be made clear through an experiment detailed in Chapter 9, it was a successful endeavor.

However, the LCC-APP was also designed as a proof-of-concept. It was a part of a larger project hoping to explore how computational parsers and *mal-rules* can be used in various pedagogical settings, and for different languages (i.e., English and Mandarin Chinese).

As a proof-of-concept, the LCC-APP served multiple purposes. The first was to corroborate the idea that mal-rules can be successfully used in higher levels of language education. Previous studies had shown that this technology could be implemented and was useful when dealing with lower levels of language proficiency – but the execution of these early experiments required a very constrained setting, including greatly limiting the vocabulary a the grammar could use. The LCC-APP shows that, with enough care, the same technology can be scaled to unrestricted levels of proficiency – as is the case in a university setting.

The LCC-APP also showed that it is possible to work with language lecturers, who are often concerned about the use of technology in classrooms, in the development of a system that abides

by their pedagogical principles and standards. While many institutions of higher learning are pushing towards more technology to be used in classrooms, it is extremely important to work directly with lecturers to identify and understand the factors that will ultimately decide if new technologies will be embraced by lecturers, or included in curricula only to satisfy administrative requirements.

At the same time, developing the LCC-APP was also useful to flag some important issues and questions. The first of these issues was the need for mal-rule enhanced parse-ranking models. The lack of such models greatly impacted the design of the LCC-APP (e.g., it was forced to use a two-grammar approach). In turn, this led to new research directions of this thesis, including the creation of a new mal-rule enhanced treebank which was eventually used to train a new parse-ranking model for the ERG. Results of these experiments will be further discussed in Chapter 9. The lesson learned from this experience also pushed the early development of a treebank along with the design of mal-rules for ZHONG. It has been made increasingly clear that mal-rules and mal-rule enhanced parse-ranking models need to be developed in parallel.

Finally, the LCC-APP also served to open the door to new and exiting research opportunities – such as the design of corrective feedback. This thesis did not set out to investigate the intricacies of corrective feedback. The goal of my research agenda was to provide technology capable of delivering corrective feedback. However, it became clear through the discussion with LCC lecturers, and further confirmed by the results of a student survey discussed in Chapter 9, that there is little consensus of what should be the right form of corrective feedback. This problem is not concerned specifically with *automatic* corrective feedback, and it is most certainly a problem that also happens during other forms of feedback (e.g., hand-written feedback delivered to students with their assignments). Adding corrective feedback to a problematic sentence does not ensure students are able to understand how to correct that sentence. The form and quality of the feedback is a topic as important as the ability to automate error detection. However, since the opportunity to monitor if students were able to work with the provided feedback was often missed before the LCC-APP was introduced, this problem was likely unknown to LCC lecturers. A system like the LCC-APP is a great tool to brave further into research concerning the quality and design of feedback. And that is a research direction I am very interested to pursue

in the future.

The next planned stages of development for this system include adding support for other input formats, namely LaTeX – which is widely used in academic writing, specially in the fields of engineering. There are multiple tools available for extracting text from LaTeX, and this addition would make this system more valuable outside classrooms.

Also on the agenda, is the development of more mal-rules to increase the classes of errors the LCC-APP can detect. As it will be made clear in Chapter 9, the LCC-APP is still unable to diagnose a large portion of errors. Since the previous work on ERG mal-rules focused on a very different population of students, it is not surprising that there are errors the ERG was still unable to catch.

Finally, another area definitely worth pursuing is concerned with the proper selection of a diagnosis for problematic sentences. Even with the creation of a new mal-rule enhanced parse-ranking model, the LCC-APP is likely to encounter sentences where is it difficult to predict the student's intended meaning – which can generate seemingly implausible suggestions for correction. One way to deal with this problem would be to take more advantage of the ambiguity generated from mal-rules to offer a small number of different options to correct the same sentence. Thus far, the LCC-APP chooses a single (highest ranked) parse to be the source of feedback. However, this is not strictly necessary. The LCC-APP could provide different sets of feedback for the same sentence, allowing the users to see more than one way of correcting the same problematic sentence. This can also help students to better locate the error, and to raise awareness that a sentence can be corrected in multiple ways.

8.2 CALLIG: Computer Assisted Language Learning using Improvisation Games

In this section, I introduce another iTELL application named CALLIG – a Computer Assisted Language Learning (CALL) web system inspired by Improvisation Games (IG). A full description of this system can be found in Morgado da Costa and Sio (2020). This application was developed in collaboration with Dr. Joanna Sio, a linguist and professional comedian, who

provided valuable insights into some of the gaming elements of this application, in hope of transforming language games into engaging humorous experiences.

Improvisation games are structured activities with built-in constraints where improvisers are asked to generate a lot of different ideas and weave a diverse range of elements into a sensible narrative spontaneously. CALLIG shows how computer-based language games can be created combining improvisation elements and language technology. In contrast with traditional language exercises, improvisational language games are open and unpredictable. CALLIG encourages spontaneity and witty language use. And it also provides unique opportunities for collecting useful data for many NLP applications.

CALLIG consists of a series of language games, integrating the principles of improvisation comedy with existing language technology and language resources, in order to create a fun language learning environment. CALLIG is integrated in iTELL to be able to access the ERG's grammatical error detection and diagnosis technology, which is being shared with the LCC-APP.

The main motivation for this project was to create a platform that could explore improvisation principles as a dimension to gamify certain aspects of second language learning for advanced learners of English. Another important motivation to develop CALLIG was the ability to collect new kinds of data that are extremely rare, which can facilitate research in certain niche fields of linguistics and psychology, such as humor and creativity.

Improvisation

Improvisation is a type of performance where performers create the content of the performance as it is performed. There is no predetermined content. Everything is made up on the spot. Such performances can be of music, theater or dance, to name a few possibilities.

Improvisational comedy is a branch of improvisational theater. There are two main types of improvisational comedy: long form and short form. Long form consists of a sequence of improvised scenes. A few suggestions would be elicited from the audience for inspiration, which act as the launching pad for the show. These scenes are often related. The thread that links them is discovered and developed as the performance progresses. Short-form improvisational

comedy consists of games (generally a few minutes in length). Each game has its own built-in constraints. For example, in the game 'Numbers', players can only speak in sentences with a given number of words. Every game requires inputs from the audience, e.g., an occupation, a location, an emotion, a number (etc.), that are used to further constrain the game or scene.

En even though the term 'improvisational comedy' is often used, one of the rules in improvisation is that improvisers do not try to be funny in a performance, contrary to what one would expect. The comic effect produced is a side-effect. In improvisational comedy, the suggestions and the constraints in the games are often incongruous and the comedy often comes from the unexpected connections that improvisers make to link seemingly unrelated ideas together. This is believed to be one of the sources of humor in improvisational comedy.

Improvisation promotes, among other things, collaboration, spontaneity, risk-taking and creative language use. The applied value of such an art form has not gone unnoticed. The techniques, the principles, tools, practices, skills and mind-sets developed in improvisation have been used for non-performance purposes, such as language learning and corporate training. Many of the major players in tertiary education have improvisation programs for business schools or for communication training (e.g., Duke, UCLA, MIT and Stanford[4]).

Improvisation elements in CALLIG

Improvisation games are regularly performed as theater performances, involving not just witty language use, but also physicality and most often than not, collaboration with multiple players. For CALLIG, only verbal improvisation is relevant. And, at its current stage, it includes only single-user games, though collaboration is on the implementation agenda. There are a lot of online resources for improvisation games,[5] though not all games can be directly usable, and most need to be adapted or designed anew to fit CALLIG's constraints.

Excluding physicality and collaboration (for the time-being), both existing and future games in CALLIG contain the following improvisation elements: (i) spontaneity; (ii) random suggestions; (iii) creativity.

Improvisation performances are spontaneous. In a performance, improvisers have to react

[4] `http://www.npr.org/2012/12/05/166484466/it-s-improv-night-at-business-school`
[5] E.g., `http://improvencyclopedia.org/`

and respond on the spot. Any delay in response due to over-thinking is considered bad improvising. In CALLIG, spontaneity is attained by having a time limit within which the user must finish the task. The time limit differs in different games depending on the difficulty level. Different time limits were tested with multiple users to decide on a length that is long enough to create tension but not too short to finish the task at hand.

In an improvisation performance, suggestions are elicited from the audience and are incorporated into the performance to highlight both the unscripted nature of the performance and the skills of the performers. In CALLIG, each game begins with a randomly generated prompt to guide the user's input. The prompts could be random words, phrases, numbers, etc. In an improvisation performance, the performers can ask for many suggestions and select among them. In CALLIG, users can also refresh and get a new prompt if they do not like the one they are given.

Improvisation activities are celebrated for their creativity. Creativity contains many aspects. Currently, CALLIG focuses on two cognitive processes, which exist in a lot of improvisation games: remote association and divergent thinking. Remote association is the process of putting associative elements into new combinations that are in some way useful, or that satisfy specific requirements (Mednick, 1962). The more mutually remote the elements of the new combination, the more creative the process or solution. Divergent thinking is the process of generating multiple related ideas or solutions for a given topic or problem. (Guilford, 1967). Divergent thinking occurs in a spontaneous, free-flowing, "non-linear" manner. In improvisation training, improvisers are told to stop filtering themselves. This inhibition of self-judgment enhances the ability to generate a large number of ideas.

All of CALLIG's games are designed to require remote association and divergent thinking. Users are invited to connect words/phrases in unusual ways, forcing them to generate uncommon ideas.

Improvisation in language learning

The most effective learning occurs when the learners are free to explore and discover with the support of scaffolds (the learning paradox) (Sawyer, 2011a). Similarly, in teaching, teachers

must allow themselves the freedom to explore within plans, routines and structures (the teacher paradox) (Sawyer, 2011b). This makes improvisation an excellent tool in teaching and learning. Improvisation contrasts with the traditional way of teaching as transmission of knowledge and skills. Instead of a prescribed curriculum and a fixed execution plan, improvisation celebrates openness and unpredictability. On the other hand, improvisation is never completely free, it occurs within a network of structures, rules and frameworks (Sawyer, 2011b).

Each short-form improvisational comedy game comes with its own set of rules and restrictions, these constraints provide a nice platform to anchor and scaffold teaching and learning. Furthermore, improvisational comedy games are highly malleable. The constraints can be customized for various training programs, especially those pertaining to language. In addition to providing contexts for witty language use, improvisation games also provide possibilities of testing particular language skills, for instance, they can be adapted for the teaching of linguistics, covering areas in phonetics, syntax, semantics and pragmatics (Sio and Wee, 2012). Improvisation activities provide varied contexts of language use that do not appear in traditional language classrooms. The entertaining nature of such games makes language learning less repetitive and more enjoyable. CALLIG can thus function as a useful complement to regular classroom teaching and learning.

Following some of the points raised in Section 2.5 concerning the current trends and limitations of gamification in language learning, CALLIG shows an unique ability to train both aspects of language structure and language skills in the domain of semantics and pragmatics, with unrestricted language, which are more suitable for advanced second language learners. These skills include but are not limited to the understanding of lexical semantics, semantic association, conceptual retrieval, different registers of language use and witty language use. Improvisational games are also engaging and fun to play because of the accidental generation of humor.

All these provide users strong intrinsic motivation to use CALLIG for language learning.

Available Games

CALLIG is developed on top of iTELL's back-end, and using only open-source resources. At its core, it is a modular web system developed using Python, SQL, Flask[6], and Bootstrap.[7] The system is fully open-source, and easy to expand in scope. The use of flexible web technologies such as Bootstrap also ensures that it can easily be played on mobile devices. It is hosted on Github[8] under the iTELL repository, and it is developed under an MIT open-source license.

CALLIG currently includes four games. The first available game is called **Sex with Me**, and it is a one-liner game. The player is given a prompt with the form: *'Sex with me is like a/an [object]!'*. The goal of the game is to justify why *'sex with me'* is like the randomly generated object. The player has to come up with a justification and type it in the answer box within 40 seconds.

The object is randomly generated using a mix of curated lists and random nouns extracted from the Princeton WordNet (introduced in Chapter 4. Definitions are provided for words that are extracted from the Princeton WordNet, because they can be quite uncommon. The player can read the definition by hovering the cursor over the word. An example prompt is shown in Figure 8.6.

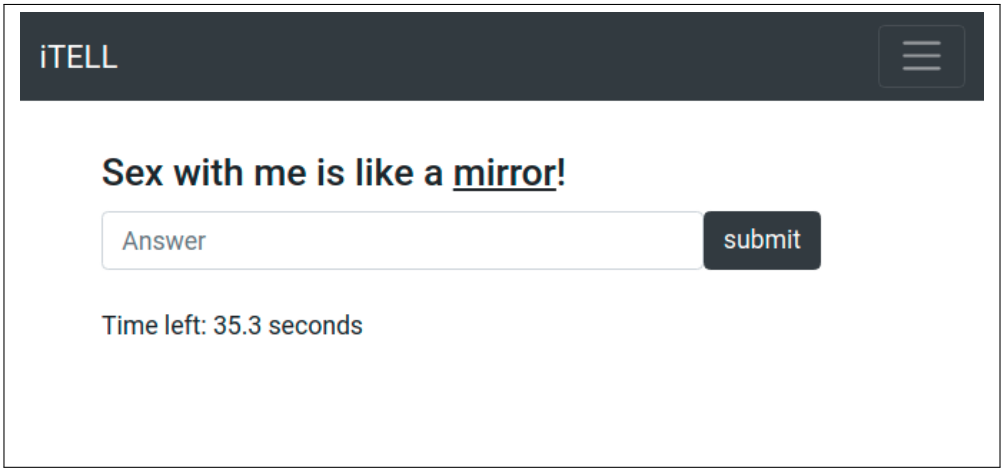


Figure 8.6: Example prompt for the game: **Sex with Me**

Possible replies to the prompt *Mirror* shown in Figure 8.6 include: *'It only works with the*

[6] http://flask.pocoo.org/
[7] https://getbootstrap.com/
[8] https://github.com/lmorgadodacosta/itell

lights on', '*You can't miss it if you go to the bathroom*' or '*You shouldn't spend too much time looking at it, or you will start to doubt yourself*'.

This game, though a bit risqué, is fun and challenging. It requires the player to quickly find features shared by both sex and the object in question. The output is often humorous due to the unlikely combination.

The second game is called **Haiku on Demand**. Haiku is a short form of Japanese poetry, containing 3 lines and comprising 17 syllables: 5 (1st line), 7 (2nd line) and 5 (3rd line). The 3rd line often contains an observation about a fleeting moment in nature. It is meant to be simple, direct and intense. It focuses on the juxtaposition of images and a sudden revelation at the end with a sense of enlightenment.

In this game, a random poem title is generated by the system. The generation of the title follows one of multiple predefined patterns using a mix of parts-of-speech and frequency information. For example, one of such patterns is the combination of a determiner, an adjective and a noun into a noun phrase (e.g., *'my oversized urinal'*, *'the hysterical assumption'*). Another of such patterns is a modified verb phrase, comprised of a uninflected verb and an adverb (e.g., *'conjugate cold-bloodily'*, *'internalize pungently'*).

After the random title is generated (with definitions provided by the Princeton WordNet), the user is then prompted to input the three lines of the haiku. A custom-made syllable-checker is ran after the Haiku is completed to confirm that the input has the desired number of syllables. If the input deviates from the haiku requirements (e.g., the user provides a first line with 6 syllables), the line is rejected and the user is prompted for another input until the requirement is met. The user has to come up with a haiku of the given title within 90 seconds. An example prompt is shown in Figure 8.7.

An example answer for the prompt in shown in Figure 8.7 could include:

I can't stop buying
words, sentences and pages
till death do us part

In this game, the player needs to show competence in the English sound system (i.e., being able to count syllables). It also exercises remote association, as it forces the players to search

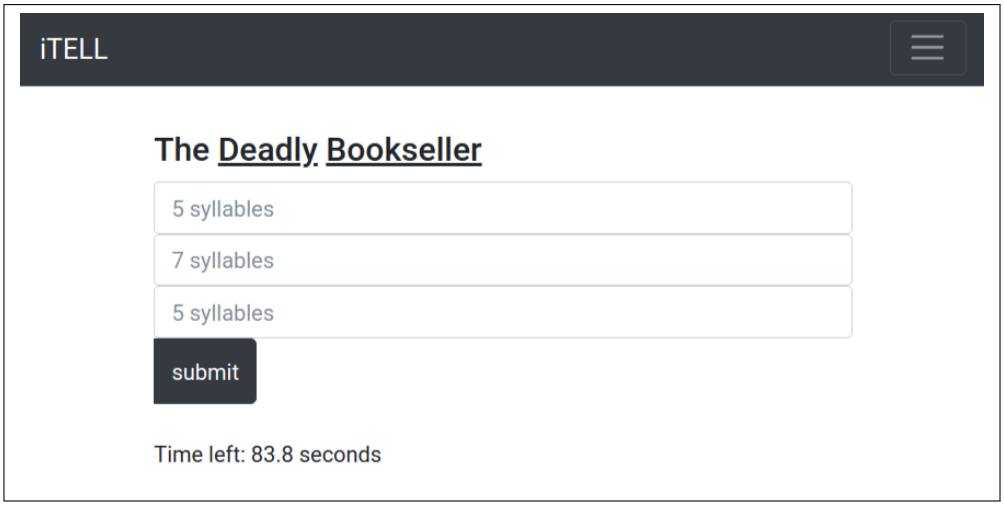


Figure 8.7: Example of **Haiku on Demand** being played

for shared-ground between often nonsensical titles.

The third game is called **Wicked Proverbs**. A proverb is a well-known piece of wisdom that advises people on how to live properly, for example *'The squeaky wheel gets the grease.'* (intended meaning: those who complain will get attention). Proverbs exist in all languages, but are often language/culture specific (e.g., similar messages are often expressed using different concepts).

The goal of this game is to invite the user to create a proverb-style piece of wisdom using randomly generated words and provide an explanation of its meaning. These game also provides a random prompt that tries to contextualize the origin of the proverb (e.g., *there is an an old Chinese saying...*, *a leprechaun once told me...*, *my grandmother always used to say...*, etc.) Similar to the previous games, definitions to the words are also provided. Based on the feedback of beta-testers, this game is timed at 90 seconds for a proverb plus its explanation. This is enough time but only if the user does not think too much about it (which is always undesirable). Figure 8.8 shows an example of this game's prompt.

An example answer for the prompt in shown in Figure 8.7 could be: (**Prompt:**) *There is an old Chinese saying that goes like this...* (**Proverb:**) *Going to a marriage counselor when you have problems with your husband is the same as wrapping socks around your neck to cure influenza.* (**Explanation:**) *Neither work but at least you can say you've tried it!*

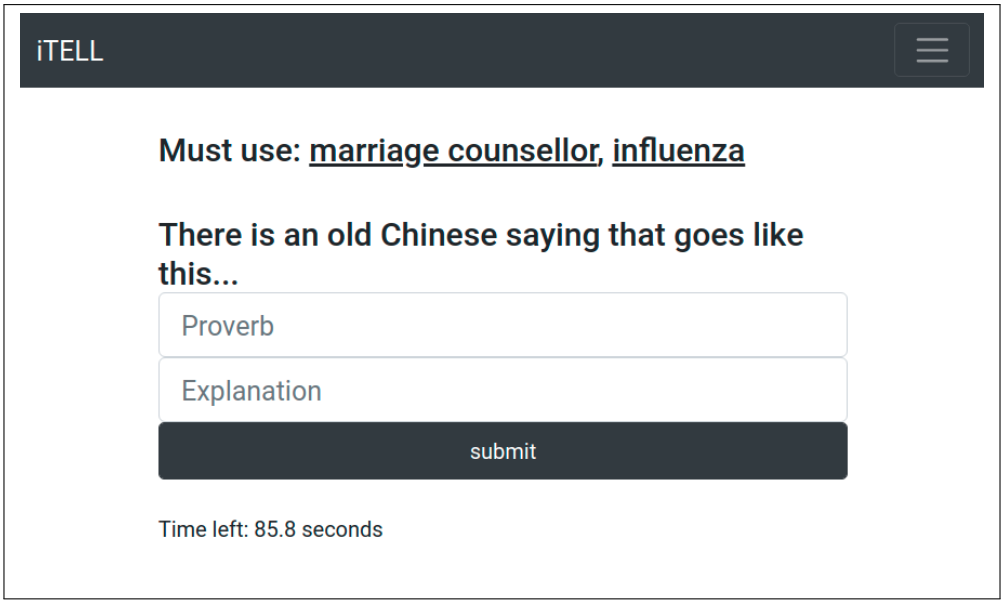


Figure 8.8: Example of **Wicked Proverbs** being played

This game also exhibits remote association as the user has to connect the given words into a grammatical and meaningful proverb. The game also involves divergent thinking when the user is invited to generate multiple proverbs with the same set of "must-use" words.

The fourth and final fully implemented game is called **Forced Links**, and it is an association game. The user is given two unrelated words as a prompt and is asked to come up with a chain of words that would connect the two given words within 25 seconds. The two related words given as the prompt can be nouns or adjectives. There is no restriction on the part of speech of the linking words nor on the number of linking words. There is, however, an imposed restriction that players cannot undo submitted words, preventing players from fixing their answers. Figure 8.9 shows an example of this game's prompt.

An example answer for the prompt in shown in Figure 8.9 (*toothy* → *physics department*) could be:

toothy → *Freddie* → *Queen* → *Don't Stop me now* → *rocket ship* → *physics department*

This is essentially a game based on semantic association. And it requires players to search for patterns of relatedness among different words, which are often different among different speakers.

Each game in CALLIG has an introduction page with instructions on how the play the game,

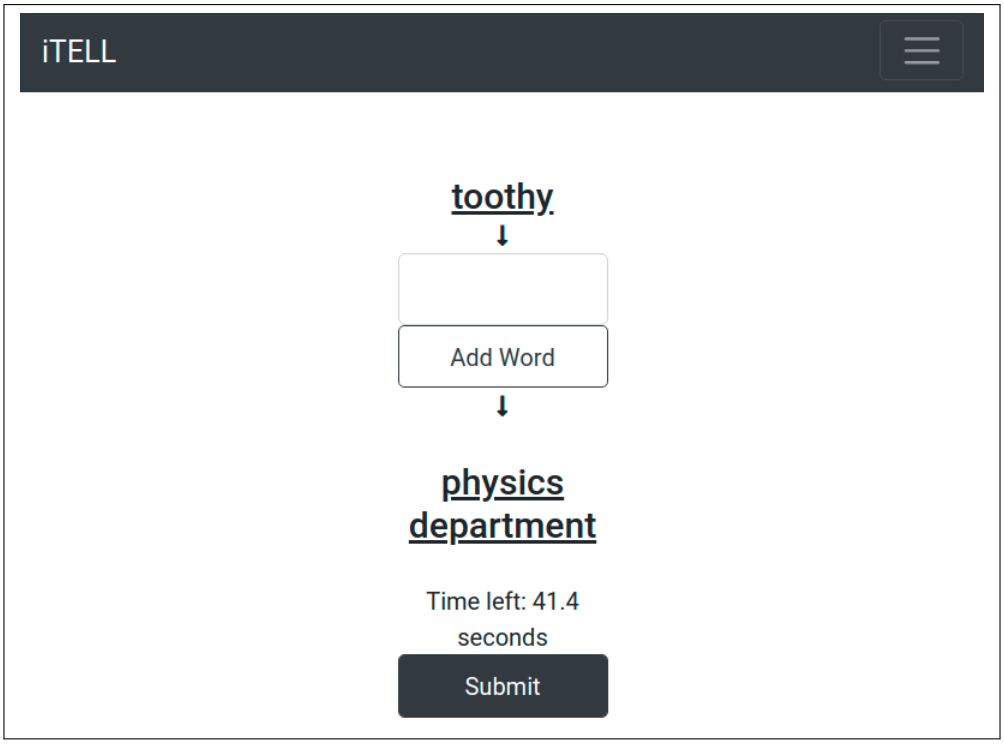


Figure 8.9: Example of **Forced Links** being played

as well as a randomized sample of responses by previous players that are shown in an automatic carousel on the top of the page. These responses include information about the author (username) and the time it took them to complete that particular game, which can be used as a competitive measure among players (i.e., being able to come up with a witty response under time pressure can be seen as an achievement). Figure 8.10 shows the introduction page for the game **Wicked Proverbs**.

Simulation of Audience Suggestions

In an improvisation performance, suggestions are elicited from the audience. In CALLIG, suggestions are randomly generated. These two types of suggestions are not identical. Suggestions elicited from an audience are almost always interesting (and potentially amusing) since audience members suggest ideas they want to see developed on stage. Furthermore, the host of the game has the option of choosing a suggestion among the many given. This also gives the option of getting rid of undesirable suggestions. Within CALLIG, suggestions (e.g., for the title of the Haiku, or for must *use words* in other games) are generated with the help of the Princeton

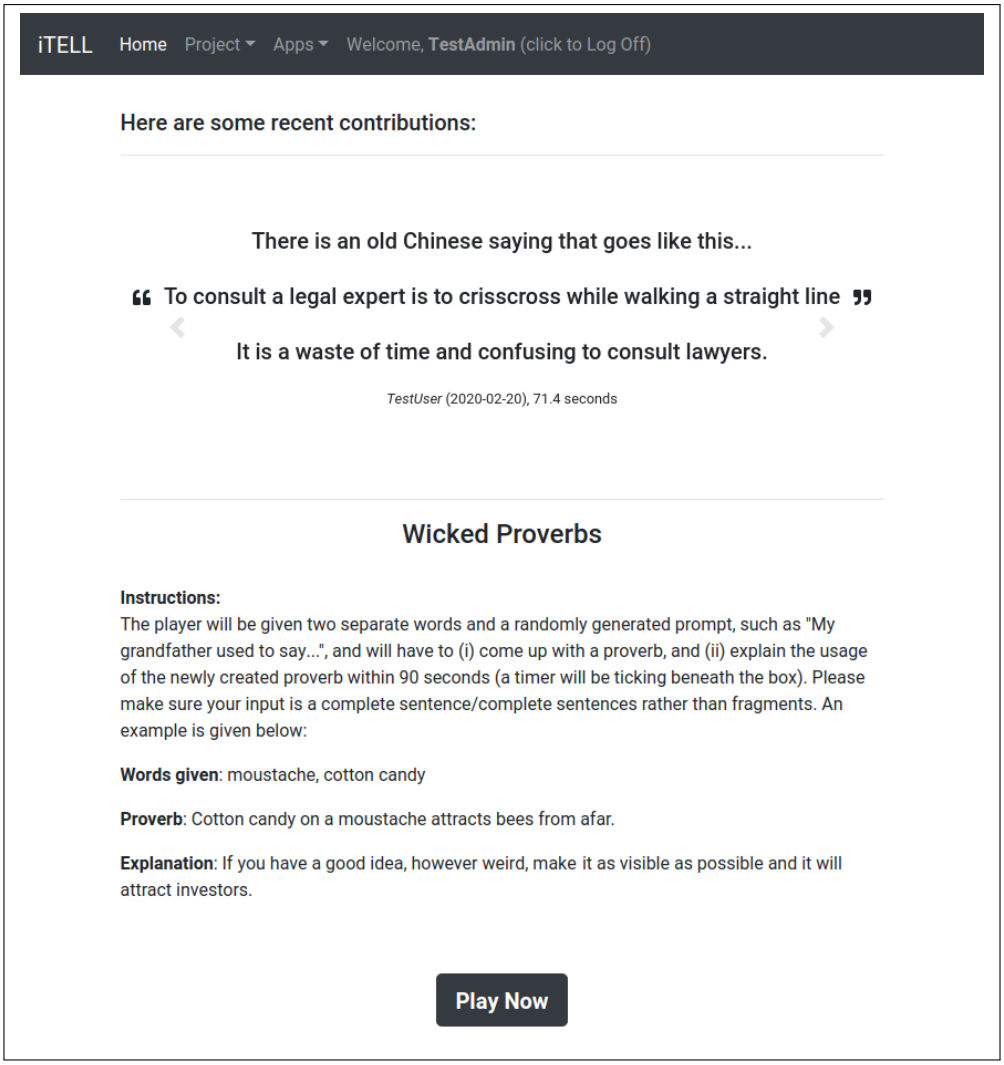


Figure 8.10: Introduction page for **Wicked Proverbs** game

WordNet (Fellbaum, 1998), which is accessed using the API provided by the Natural Language Toolkit (Bird, 2006).

Wordnets are often large lexical databases, where open class words (i.e., nouns, verbs, adjectives and adverbs) are grouped by sets of synonyms into semantic concepts. These rich semantic graph also allow the encoding of some measure of semantic distance, which can useful to expand certain games (i.e., Forced Links).

The Princeton WordNet is used in tandem with curated word-lists designed specifically for each game. While the wordnet is able to provide a level of true randomness, curated lists of words and expressions are used to maintain a level of familiarity and humor that would be

expected from a real-life audience. It should be noted that the system can function perfectly without such curated lists. However, true randomness sometimes generates concepts that are infrequent, and possibly unknown to the user. Concepts like this come with a definition, provided by the wordnet. This can also be used as a way to introduce new vocabulary to second language learners. The mixture of randomized items from wordnet and curated word-lists ensure that users will not be given too many unfamiliar words consecutively, which might lead to frustration.

Despite attempts to control these simulated suggestions the best way possible, there is no guarantee that all suggestions are meaningful or sensible. For examples, in **Haiku on Demand**, the system has generated titles like "the weak pisha paysha" and "the hand-sewn welterweight". The generation of this kind of nonsensical titles often has to do with semantic mismatch that is too far apart for the user's interpretative accommodation. The current way to address this is to allow users to refresh the game and get a new prompt if they do not like the one they are given. These infelicitous suggestions are kept by the system, and can be used to prevent similar suggestions in the future.

Linguistic Adequacy and Feedback

Whenever appropriate, CALLIG tries to enforce certain degrees of linguistic adequacy. This is the pedagogical dimension of the system. It tries to use each game to enable "learnable moments" throughout the user experience. The system tries to be as precise as possible, ignoring problems when it is not prepared to provide useful feedback.

This linguistic adequacy takes different forms in different games. In the **Haiku on Demand** game, for example, only answers that respect the syllable count for each line are accepted as a valid. If the user fails to follow the 5-7-5 syllable constraint, then they will be notified and prompted to try again. In principle, this should raise the user's awareness of how to count syllables, a skill that can help with pronunciation and fluency in a foreign language.

Given **Haiku on Demand**'s poetic nature, there would not be much sense to perform strict grammatical checks in this game. For other games, however, such as **Sex with Me** and **Wicked Proverbs**, grammatical checks are appropriate.

Following the discussion presented in Section 8.1, CALLIG is also using the ERG (Flickinger, 2000; Flickinger et al., 2000; Copestake and Flickinger, 2000) to identify many different classes of grammatical errors. CALLIG uses only a selection of the error checks developed for the LLC-APP, and is focused on strictly grammatical problems (e.g., problems with subject-verb agreement; the omission of articles for singular count nouns; etc.). Other problems checked by the LCC-APP, such as problems of style, are not relevant for an informal setting like this. Similar to what happens with the LCC-APP, more than one error can exist in each sentence. And for each error identified in a sentence, the system will generate a constructive feedback message that aims to explain the error and help the user to avoid it in the future. The feedback messages generated are based on the corrective feedback designed for the LCC-APP, but have been slightly simplified. Since this application is not being used by the NTU team, I had the liberty to test different styles of feedback.

Also contrasting with the behaviors of the LCC-APP, when the system is unsure what is wrong with a sentence (i.e., if the ERG cannot provide a parse for a given sentence), then the error is completely ignored. This is done with the user's experience in mind, as flagging too many ungrammatical sentences might be demotivating for the user. This decision was also based on the fact that players do not have a further network of support (e.g., lecturers) to discuss the problems of each sentence. In addition, it was important to ensure that grammatical checks only happen for games where full grammatical sentences are expected – currently only **Sex with Me** or **Wicked Proverbs**. Figure 8.11 shows an example of how CALLIG reports a grammatical problem. In this case, the system is able to correctly identify the lack of a determiner before the noun "jungle". In order to reward the grammatical sentences, only grammatical answers are displayed as good answers by the system. Ungrammatical answers are, however, kept in the system's log, so they can be used for future research.

Discussion and Future Plans

CALLIG is still under development. As a part of this thesis, CALLIG shows that the technology developed for the LCC-APP is flexible, and that it can be easily incorporated in other applications. CALLIG shows that the same technology used in LCC-APP can be used in less formal

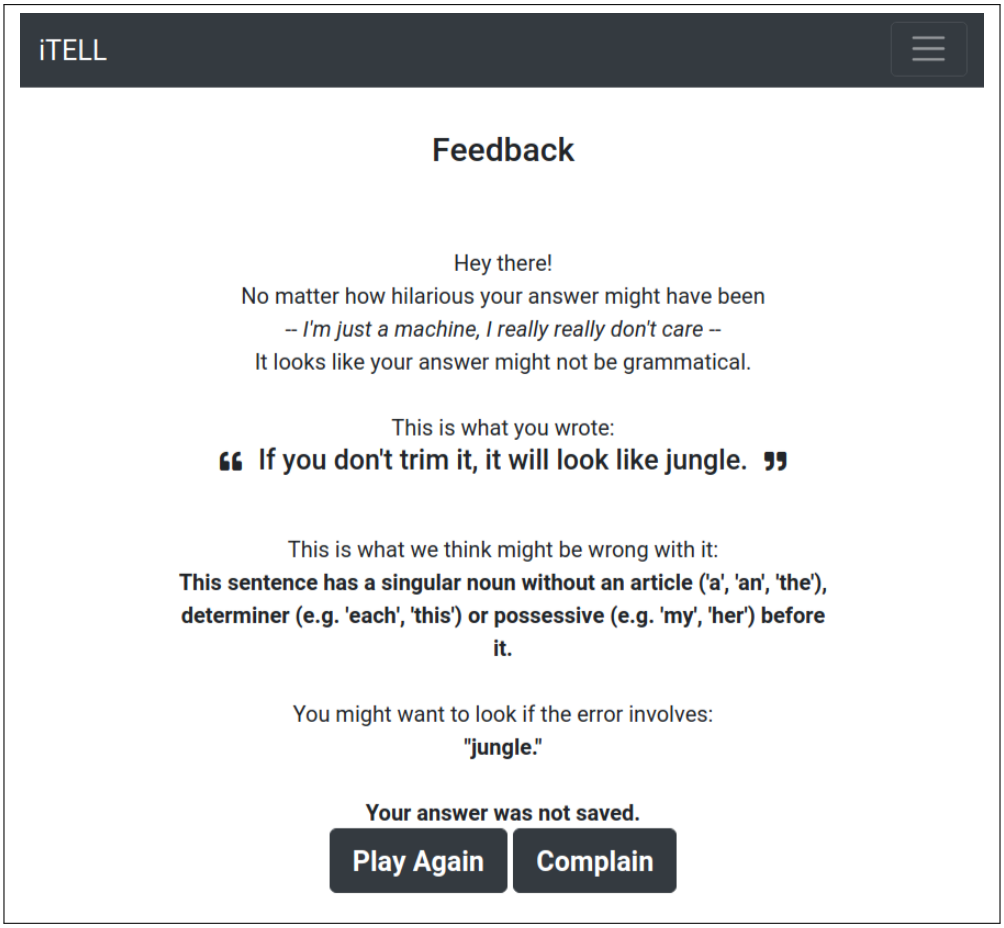


Figure 8.11: Example of constructive feedback provided for an ungrammatical answer in **Sex with Me**

settings of language learning – e.g., advanced classes of English as a second language –, where issues of style and semantic awkwardness can be put aside to the benefit of language games that reward creativity.

The improvisation elements present in CALLIG's games allow the co-existence of *structure* and *freedom to explore* for both teachers and students, which aims to be an enjoyable tool for language training. Improvisation exercises in language classrooms, as of now, require the physical presence of a group of people. And despite its possible benefits, there are people who do not feel comfortable physically performing these games (e.g., in public or in a classroom). CALLIG provides a platform for playing language games in a more private and less labor-intensive setting. It can be useful to build confidence before leaping to physical performances, or as training ground for important skills such as spontaneity, collaboration, and risk-taking, or as a safe way

to participate during a pandemic.

There are currently four areas targeted for CALLIG's further development.

Expanding the Game Selection

CALLIG's current four games serve as a proof-of-concept of what is possible. However, there are many other similar games that could be made available through this system. Many of these new games could borrow the structure/design of existing games, making it quite simple to add new games. For example, the are many one-liner games which are structurally similar to **Sex With Me**. Examples of one-liner games include **Famous Last Words** and **Pick up Lines**. In **Famous Last Words**, the prompt would be the name of a famous figure (dead or alive; real or fictional), and the input would be the last line that the figure utters before dying, making use of common knowledge of such figures. In **Pick up Lines**, the prompt would be an occupation or a famous character, and the input would be a pick-up line uttered by someone with that occupation, playing with stereotypes of different occupations or personalities.

In principle, games must satisfy at least one of two requirements: a) they must be able to focus on some aspect of language learning that CALLIG is able to control and diagnose; or b) they must produce relevant linguistic data related to creativity or humor that can be used in further research on creativity, humor or language learning.

Advanced Linguistic Constraints

With the use of the ERG, CALLIG is also able to impose and check for certain classes of advanced linguistic constraints. For example, similar to what happens with **Haiku on Demand**, where the system is checking the number of syllables per line of input, using the ERG could allow CALLIG to check if specific syntactic phenomena had been used. For example, the system could request and check if a passive construction or a definite noun phrase was used in a specific input.

These specific linguistic requirements can also be incorporated in game design, e.g., in the game **Reverse Trivial Pursuit** – an improvisational language game that focuses on question formation.

In this game, the system prompt should be interpreted as an answer. The goal of the game is for the player to provide as many possible questions as possible. For example, if the generated prompt is *"my intellect"*, some potential answers/questions would be *"What is your most valuable possession?"*, *"What is the sharpest thing in the world?"*, *"What is the thing that makes you unattractive?"*, etc. This game would be especially interesting to test and help learners of English with the formation of questions, as all of the user's answers must be in the form of a question – and this is something that can be checked using the ERG.

In other words, CALLIG can be used to explore the full range of syntactic and semantic contexts that can be enforced by using the ERG. This can be used directly in game design (e.g., for the game **Reverse Trivial Pursuit**) but can also be used as further constraints in other existing games.

Social and Collaborative Gaming

Despite being hosted online, where users can see other people's answers, the current games focus on only on a single player environment.

Currently, the system takes the role of both the host and the audience, in the implemented games, providing instructions as well as suggestions for these games. Nevertheless, adding social features to it would enable CALLIG to dwell deeper into performance style improvisation games, as well as allow meaningful interaction between users. There are plans to extend the social and collaborative setting of CALLIG in two ways:

1. Better integration with social platforms, allowing players to share the results of their interaction with the system. This would allow users to publish the writings they feel the most proud of, and share them on social media. The published works will be accessible to all, and registered users will be able to upvote or downvote other users' writings (possibly even on a scale).

2. An improved platform supporting multiplayer games. Improvisation, as a performance, is generally collaborative in nature, resorting to the use of a "group mind" to create something unpredictable. To explore this, CALLIG would benefit from having a platform capable of hosting collaborative gaming (i.e., chat-room style gaming). Different users would be able to play

the same game either competitively or by requiring them to work together to create a coherent whole (e.g., each active user takes turns in guiding the development of a narrative). Having this feature would allow CALLIG to host games that have been left out because they only make sense in a collaborative setting, as well as different variations of already implemented games.

Multilingual Support

Given the mostly language agnostic design of improvisation games, most of the games currently implemented in CALLIG could easily be supported in other languages. With some exceptions (e.g., the syllable counter for *Haiku on Demand*), most of the language technology CALLIG uses revolves around the semantic hierarchy provided by a WordNet. Fortunately, resources such as the Open Multilingual Wordnet (OMW, Bond and Foster, 2013) include parallel semantic data for hundreds of languages, facilitating this process.

In addition, the grammatical error detection currently provided for English by the ERG, while not an essential to play the games, now has a parallel in the mal-rule enhanced ZHONG, introduced in Chapter 6. As such, the first language planned for CALLIG's multilingual expansion is Mandarin Chinese. Mandarin Chinese is a fairly well-resourced language including, for example, the Open Chinese Wordnet (Wang and Bond, 2013) – also integrated in the OMW. And while it is true that the ERG coverage far surpasses ZHONG's, most games can be adapted to use a smaller pool of vocabulary, effectively lowering the proficiency level required to play the games.

Explore Applied Usages

Finally, the creative outcomes (i.e., the multiple formats of spontaneous writing) produced by the players' interaction with CALLIG will generate a lot of written data with potential applications (e.g., data including semantic association, humor ranking of different forms of creative writings, etc.). This data can then serve as a rich resource for both creativity studies and linguistic studies. For instance, games that require complete sentences as input (and that were checked for grammatical errors) can generate data valuable data for learner corpora. Another example is the game **Forced Links**, which can provide association data between words useful to enrich se-

mantic hierarchies of resources like wordnets. As new games are added to CALLIG, so does the type of data it can generate. The design and development of future games will consider possible research usages of the data it collects, ensuring CALLIG will remain useful to an increasing number of research topics.

Improvisation often generates humor. However, improvisation performances are not generally transcribed, humor studies based on improvisation data are rare, if they exist at all. CALLIG can change this by generating language data readily available for research on humor, creativity, linguistics and other related fields.

8.3 Summary

This chapter introduced two applications developed under iTELL – a suite of web-based applications exploiting deep computational parsers in intelligent Technology-Enhanced Language Learning environments. These applications share a common back-end where mal-rule enhanced grammars, such as the ERG, can be exploited to aid in different aspects of language learning.

The first application introduced was the LCC-APP, a web system that integrates English grammatical error detection and course-specific stylistic guidelines to automatically provide feedback on student assignments. This system was developed in collaboration with a team of lecturers at NTU, and its development was informed using the NTU Corpus of Learner English, discussed in Chapter 5. It is now capable of performing 80 different grammatical and stylistic checks, providing carefully designed constructive feedback to help students identify and correct them on their own. A preliminary evaluation of the system's *in-class* performance has shown measurable improvements in the quality of student assignments. This will be discussed in detail in Chaper 9.

The second application introduced was CALLIG, a collection of language games using elements of improvisation comedy. CALLIG is integrated in iTELL in order to access the ERG's grammatical error detection and diagnosis technology, which is being shared with the LCC-APP. CALLIG is still in its early stages of development. It includes four distinct games which showcase how mal-rules can be effective in less formal learning environments of lan-

guage learning. In the future, CALLIG will be exploited to collect data that is useful to many different lines of research, including but not limited to, research on second language learning, lexical semantics, common sense reasoning, humor and creativity.

Part III:

Evaluation and Conclusions

Chapter 9

Evaluation of Results

This chapter provides a detailed account of the main experiments conducted to evaluate multiple components and systems developed during this thesis. It starts with an extrinsic evaluation of the iTELL's LCC-APP through a blended-learning experiment. This is followed by a set of intrinsic experiments, for both English and Mandarin Chinese, to measure the impact of using mal-rule enhanced models for error detection and diagnosis, as well as more generic parse coverage experiments to evaluate ZHONG's latest version (which is released with this thesis).

9.1 iTELL in a Blended Learning Experiment

This section reports on the extrinsic evaluation experiment conducted using the iTELL's LCC-APP, the English writing support system described in detail in Chapter 8.

This experiment answers the question: *Can the automated corrective feedback provided by the iTELL's LCC-APP improve the quality of engineering students' assignments?*

The answer to this question can help determine if there is enough evidence to recommend the integration of this writing support system into other course curricula and, at a fundamental level, whether or not to continue supporting the development of this and other similar systems.

A full discussion of the motivation of this system, as well as a detailed description of many of its components and design choices can be found in Section 8.1.

Methodology

In order to follow the ethical guidelines imposed by NTU, and to cause minimal disruption in the course curriculum through which this system was evaluated, this experiment used a written assignment which was already part of the curriculum. The assignment consisted of a technical proposal that described an engineering solution to a real life problem, using a fixed document structure consisting of: background, problem, solution, benefits, implementation, budget and conclusion. The assignment was written in pairs (with a few exceptions due to class size, students that dropped the course, etc.), and had a limit of 800 words.

All presented procedures were thoroughly validated by IRB standards at NTU. Since there was an anticipated learning benefit from the interaction with the system, some care was necessary to prevent the use of any preferential methods with specific groups of students. Even though participation was voluntary, the experiment's design had to ensure that all students had the same amount of access to the system – and hence could evenly receive any benefits rising from using it.

The population included a full cohort of 1,855 engineering students enrolled in an Engineering Communication course lectured by the LCC, at NTU. Since local universities are sensitive about nationality profiles, full demographic details from the participants were not collected. However, based on historical data and the lecturers' observations, the cohort consists of a largely (more than 70%) Singaporean undergraduate population, with small proportions of Southeast Asian, South and East Asian students, and very few European students. This comprises a largely Asian population, most of them with native or near-native levels of English proficiency, and the rest distributed along lower levels of proficiency.

Students had a previously established deadline to submit a course assignment and, until the date of submission, were unaware of the existence of the system. After submitting their assignments, each student received an email informing them of the existence of a new system designed to give them feedback on probable grammatical and stylistic problems in their writing. Students were given one extra week to use this system and edit their assignments. The decision to only inform students about the existence of the system after their first submission was to guarantee

that students took the first submission deadline seriously and submitted a finished assignment in their first submission. It was assumed that if students knew about the extension of the deadline beforehand, many would not fully finish their assignment in time for the first submission – making the first (i.e., possibly unfinished) and second submissions not truly comparable.

There was no limit to the number of times an assignment could be uploaded onto the system, and students were encouraged to use the system as much as they wanted before the final deadline. A second submission was set up for exactly one week after the first submission. All students had to resubmit their assignments, regardless of having used the system or not. This was done to encourage (though not force) the use of the system. Interaction with the system was completely optional, since both submissions were done using the university's learning management system.

In order to determine if the system had a positive impact on student language use, a paired-blind review of pre- and post-system submissions was set-up.

Since participation was voluntary, and in order to avoid ceiling effects (i.e., evaluating the system with assignments where there were too few errors to be corrected; see Bruton, 2009), a decision was made to sample only assignments where (i) the students had used the system and (ii) the system detected a decrease of at least six errors in the final submission (when compared with the first submission). This meant that this experiment looked only at submissions where the students had something to gain from using the system and, at the same time, had chosen to take advantage of it. Using these criteria, 108 assignments were randomly sampled.

A group of four experienced lecturers evaluated both pre- and post-system submissions of these 108 assignments. Given that the assignments were not very long (i.e., 800 words) and that students were provided with a full week to improve their assignment, it was not possible to prevent the second version from taking a completely different form, including focusing on a completely different topic. When this happened, a comparison between two completely different versions was not deemed of interest for this experiment. In total, three assignments were unanimously removed from the sample because the content between submissions were not deemed comparable. The remaining 105 assignments were deemed comparable, having both versions differing only in style or sentence structure, but not in content.

The number of sampled assignments was defined in negotiation with the four lecturers who

had to put in the extra work to help conduct this experiment. Since the sample was randomized, the lecturers were likely grading assignments from students that did not belong to them, and hence could not really see this task as teaching load. While it would have been ideal to evaluate a larger sample (or even the full cohort), man-power constraints meant that this was not possible. In the end, the four lecturers agreed to spend the time equivalent to grade roughly 100 documents, which determined the size of this sample.

The 105 selected assignments were separated into two groups (of 52 and 53 assignments each) and each group was given to two different lecturers for comparison. Each assignment had two unidentified versions – an original version (before using the system) and a post-system version. As such, each lecturer graded between 104 and 106 documents. As grading was known to be too subjective (see Section 5.2 for a fuller discussion), lecturers were asked, whenever possible, to choose the better document, without knowing which document corresponded to which version. Lecturers used their standard rubric to evaluate the assignments (including the quality of content and its organization, referencing of sources and issues concerning language and style), but were asked to provide special attention to issues concerning language and style when choosing the better assignment.

In addition to the double-blind review of student assignments, a few questions concerning this experiment were included as as part of the habitual course evaluation survey. Participation in this survey was voluntary. Students were asked if they found the system useful, and whether they were interested in using the system outside the classroom, and were also invited, in an open ended question, to share any suggestions or comments about the system.

Experiment Results

At the end of the experiment, the system had received 2,581 document submissions from 798 pairs/groups of students (i.e., roughly 86% participation). About 55% of the students agreed to release their assignments for future research. The system found 34,141 problems (this includes sentences with multiple problems) across all submissions.

Table 9.1 shows the top 20 classes of errors detected by the system, with their respective frequencies. The distributions of the number of submissions per pair/group of students and of

the number of errors per submission were quite asymmetrical. Some students submitted their assignment as many as 36 times, while the average number of submissions was between three and four. Also, even though the average number of problems found in each assignment was just over 13, some submissions had over 100 identified problems. This asymmetry reflects, in part, the fairly mixed population (in terms of English proficiency) discussed above.

Rank	Label	Freq.
1	No Parse (unspecified problem)	14,834
2	Use of instructions/commands	4,032
3	Overly long sentences	3,727
4	Singular nouns without specifiers (e.g., article or determiner)	3,443
5	Use of first or second person singular pronouns	2,485
6	Repeated words	1,238
7	Use of informal words or expressions	942
8	Use of verbs that do not agree with their subjects	909
9	Use of questions	894
10	Use of contractions	372
11	Omission of the definite article ‘the’	169
12	Use of indefinite articles with uncountable nouns	150
13	Use of comma splicing	126
14	Incorrect use of the verb form ‘are’	126
15	Missing, inappropriate or unnecessary modal	125
16	Use of singular nouns with plural determiners	118
17	Incorrect use of the verb form ‘is’	71
18	Use of plural with mass nouns	61
19	Use of formal or archaic words or expressions	39
20	Use of the wrong form of the indefinite article ‘a/an’	29

Table 9.1: Frequency of the top 20 classes of errors detected by the system

Concerning the blind grading experiment, even though choosing the best version was sometimes reportedly difficult, all lecturers were able to make a decision for each assignment. Table 9.2 provides a summary of these results.

Because lecturers chose the better assignment without knowing which were pre-system submissions and which were post-system submissions, it was assumed that choosing a post-system submission would be an indicator of the system’s potential positive impact. On the other hand, choosing a pre-system submission could be interpreted as a potential negative impact on the quality of writing.

It is important to note that the team was aware that the extra week given to the students

	Pre-System	Post-System	N	Agreement
Lecturer A	19%	81%		
Lecturer B	12%	88%	52	73%
Lecturers A∩B	2%	71%		
Lecturer C	4%	96%		
Lecturer D	30%	70%	53	70%
Lecturers C∩D	2%	68%		
Avg. A, B, C, D	16%	84%		
Avg. A∩B, C∩D	2%	70%		

Table 9.2: Blind Grading of Pre-System and Post-System Assigments (n=105)

might have also contributed to the improvement of the assignments. The ability to interpret the results as suggested above comes from the way assignments were selected. This experiment only considered assignments that actively engaged the system during this extra week more than once (so the decrease in errors was quantifiable), and that did not greatly change the overall structure of the assignment. Because of this, it can be stated that any measurable improvement in the quality of writing was at least in part due to the use of the system.

The results of the lecturers' decision (Table 9.2) show promising trends, especially considering that the system is still in its early proof-of-concept stage, and that much can still be done to improve its performance. Even though there were some differences in the lecturers' preferences, all lecturers gave clear preference to post-system submissions. The agreement between lecturers A and B was 73%, and 70% between lecturers C and D.

The average of all four lecturers' individual decisions (i.e., *Avg. A, B, C, D*), shows that the system had a potential positive impact in up to 84% of the submissions. In the remaining 16% of the cases, at least one lecturer chose the pre-submission version of the document as better. When looking only at those assignments where the lecturers agreed on their decision (i.e., *Avg. A∩B, C∩D*), the numbers are even better: there is a clear potential positive impact in up to 70% of submissions and a potential negative impact in only 2%. For the remaining 28%, there was neither a decisively positive nor decisively negative impact.

The results of this experiment seem to answer positively to the research question posed above: There is enough evidence to support the hypothesis that automated feedback provided

by the iTELL's LCC-APP is able to improve the quality of engineering students' writing.

Two points in the results presented above deserve further discussion. The first point concerns the slight asymmetry in the lecturers' choice of the better assignment. This is best illustrated by observing the individual decisions for lecturers C and D, shown in Table 9.2. Even though the agreements between lecturers A and B, and lecturers C and D are comparable (i.e., around 70%), lecturer C chose the post-system assignment as better 96% of the time (the highest of all four lecturers), while lecturer D preferred the same assignment only 70% of the time (the lowest of all four lecturers). It is relevant to point out that both lecturers, C and D, looked at exactly the same assignments. This was not entirely surprising, as different lecturers are known to have different levels of strictness and attention concerning different classes of problems. But having awareness of this is extremely relevant when conducting experiments like the one being reported here. It is important to stress that systems like the iTELL's LCC-APP should not only cater to a specific population of students but, to some extent, also needs to cater to a particular population of lecturers. Disagreements among lecturers will always be unavoidable.

The second point concerns the data presented in Table 9.1. As can be seen from the data, the most frequent class of error that the system was able to detect was the unspecified error (which is the error message displayed when the system was not able to obtain a parse for that particular sentence). A large majority of these errors came from the system's inability to correctly identify which sections of a document to check (i.e., find contentful sentences). In the early stages of development, when this evaluation took place, the system was yet unable to correctly detect section headers, tables, references or mid-sentence citations – which were the source of most of these errors. At the time of this experiment, students were alerted to this, and told to ignore such errors if they contained any of the elements discussed above. Fortunately, as discussed in Chapter 8 in reference to Figure 8.2, this module of the system has improved immensely on this front since this experiment took place.

Student Survey Results

A total of 236 students answered the course evaluation survey. The two main questions concerning the interaction with the system revealed that the majority of students did not only agree

that system was beneficial (see Fig. 9.1), but that students were also interested in using it for other courses and assignments (see Fig. 9.2). Succinctly, this shows that the system was well received by the students, in addition to improving their assignments.

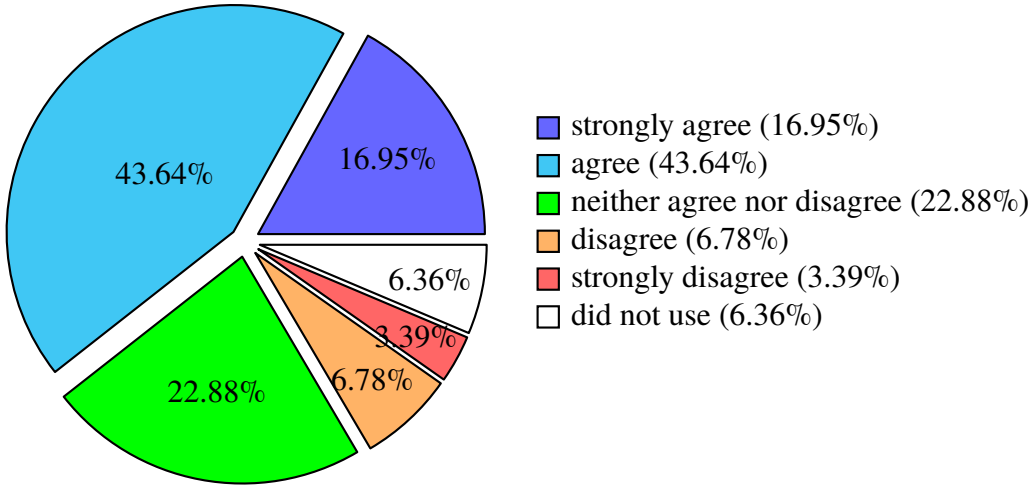


Figure 9.1: Student answers to the statement: 'I found the online error detection tool useful.' (n=236)

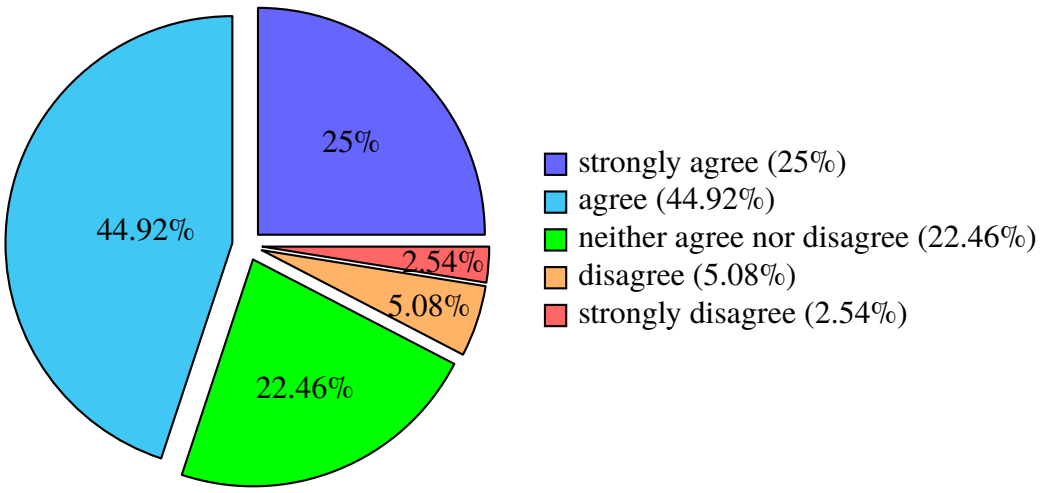


Figure 9.2: Student answers to the statement: 'I would like to use the online error detection tool for other courses and assignments.' (n=236)

The survey also contained an open-ended question inviting students to share any suggestions or comments about the system. An informal thematic analysis identified four main themes: non-expansive comments on the overall quality of their experience; concerns about the experiment's design; new feature requests; and concerns about the quality of the feedback.

Roughly one-quarter of the comments fell under the first category, where students limited themselves to provide non-expansive comments concerning their experience with the tool. Examples of these included: '*its good (sic)*'; '*Not very accurate*'; and '*just quite funny*';

Given the shallowness of these comments, and the fact that they are mostly represented by the other two survey questions presented above, these will not be discussed further.

The second category contained comments mostly concerned with the way the experiment was conducted, and the constraints imposed by its design. Examples of this category included comments such as: '*Inform students about it in class during the first lesson*'; '*Good that I am able to submit the assignment multiple times*'; and '*Releasing it earlier and allowing access to it out of campus way better (sic)*';

Most comments in this category expressed the students' dislike of being surprised with the opportunity to use the tool after their assignments were first submitted. On the other hand, students seem to have enjoyed the unrestricted use of this tool, and would have preferred earlier access to it. The note about off-campus access in one of the comments shown above is a reference to the fact that students had to either be on campus or use a VPN connection to access the system during the experiment. This restriction was a limitation of the university server in which the system was being hosted, and has since been removed.

The third category contained comments that were seen as a request for new features (i.e., features that fall outside of what the system was originally designed to achieve). These included comments such as: '*Could be more useful if it could correct the mistake instead of just highlight*'; '*Additional feature: reference checking*'; and '*It did not show the similarity report which we could not know (sic) which part has the potential for plagiarism*';

Some comments in this category confirmed the team's intuitions concerning the students' preference for a system that would correct their errors (instead of just detecting them). And although technically possible, this goes against the system's design goals – to engage and involve students in the correction process. Some students also requested features that would make the tool more holistic (e.g., dealing with citations, references, and plagiarism). However, as there is still a good amount of work to be done on other essential features, these features are not likely to be considered in the very near future.

Finally, and most importantly, about half of the comments alluded to some dimension of the quality of the feedback. Most of these comments showed that some students were overwhelmed with the task of correcting their sentences with the given feedback. Some examples of comments in this category include: '*Provide better error messages*'; '*Not bad, but when some mistakes arise, it could not properly explain where the mistakes are*'; '*At times, there were errors identified which essentially said "There seems to be an error here but we are not sure what it exactly is". Could improve on clarifying such errors*'; '*The errors are not explained clearly, thus unsure on how to make changes*; and '*Some of the suggestions are very ambiguous or general, does not identify what is wrong but instead suggest that there MAY be error*;

These comments need to be taken very seriously but, to some extent, fall outside the scope of this thesis. Following the discussion provided in Section 8.1, the main goal of this system was to provide meaningful corrective feedback – i.e., feedback capable of guiding students through the successful correction of a problematic sentence. However, the design of the corrective feedback messages were not a focus of research during this thesis. The system relied heavily on the LCC team's knowledge to compose feedback messages that could be understood by their students.

As was discussed in the previous chapter, the team of LCC lecturers strongly supported the use of indirect corrective feedback, as it engages and guides students through the writing process. However, the concept of indirect corrective feedback is generally not strictly defined. Different kinds of indirect corrective feedback differ in the level of explicitness they provide. At the non-explicit end of this scale, for example, indirect corrective feedback can simply comprise how many errors a sentence has, or both the number and the location of the errors (through circling, underlining, etc.). In more explicit forms, indirect corrective feedback can provide coded errors (e.g., using 'T' to mark a tense problem, etc.), or provide full meta-linguistic explanations (as is the case for this system).

To make matters more complicated, even though there are plenty of studies that provide evidence of the potential of using indirect corrective feedback in learning, it is a fact that applying indirect methods often assumes students have enough linguistic knowledge to understand and correct these errors (Eslami, 2014). However, as it has been pointed out by some students during the survey, this was not always the case.

One possibility for why the corrective feedback was not always clearly understood is the presence of linguistic jargon. The expectation that all students understand concepts such as 'agreement', 'indefinite' or 'countable' might be overly optimistic. Expanding this line of thought, it may also be due to the fact that not all students are native English speakers, and that this linguistic jargon poses a problem but only because it is provided in English (i.e., and possibly would be less of a problem if it were offered in the student's native language). One could also entertain pessimistic possibilities, such as the case where the students have the necessary knowledge, but the way in which the corrective feedback was phrased might not have been the most optimal or efficient. An important note to be made is that the problem with corrective feedback quality is not necessarily bound to the fact that it was delivered through a computer system. Corrective feedback was provided by lecturers before this system existed. And it is very likely that these issues may well have existed before the incorporation of this system in the course curriculum, but that students either ignored this feedback or did not have a channel to voice their concerns.

Unfortunately, despite being an excellent path for future work, answering these questions was not the focus of this thesis.

9.2 Evaluating the new ERG Parse Ranking Model

This section reports on the results of the evaluation experiments conducted using the new mal-rule enhanced parse ranking model for the English Resource Grammar, described in Section 7.2.

The main goal of this evaluation is to determine if the parse ranking model produced with the newly created mal-rule enhanced treebank has a measurable an impact on the performance of the error detection and diagnosis capabilities of the mal-rule enhanced ERG (edERG). To a large extent, this can and should be measured independently from the detection of other classes of errors the LCC-APP is able to diagnose (most of them stylistic in nature). Similarly, it can and should also be measured independently from many other (semantic and pragmatic) problems that can be encountered in student assignments. These would be errors that lecturers would very easily diagnose, but that were never within the scope of the grammar to diagnose (e.g., 'convoluted sentence' or 'unclear reference for pronoun'). In addition to this scope problem,

different lecturers have different sensitivities to different classes of errors – and some mal-rules available in the ERG may not be equally appreciated by every lecturer (e.g., the grammatical number and countability of the noun 'elderly'). These reasons, coupled with the scarcity of data tagged directly by NTU lecturers, led to the decision that the experiments reported here should measure only the ERG's ability to better detect and diagnose the classes of errors it was designed to capture – leaving other concerns about the integration of this technology in pedagogical software as precursors for further research.

These experiments have, as a basis, a test set of 1,000 sentences collected from an unannotated portion of the NTUCLE, spanning approximately 30 student assignments. Considering that the treebank used to train the model being evaluated here was also part of the NTUCLE, using an unannotated portion of the same corpus was the best way to provide an accurate measurement of the improvements that would be expected from using this new model to detect and diagnose errors within NTU's engineering student population – which is LCC-APP's current goal.

As described in Section 8.1, the current version of the LCC-APP uses a two-step approach to error diagnosis: i) the standard release of the ERG is used to provide a filter to likely ungrammatical sentences; ii) the filtered document is then further processed by the mal-rule enhanced version of the same grammar (edERG) to perform error diagnosis. The main reason for this, as discussed earlier, was the high rate of misdiagnoses generated by the edERG alone. This high rate of misdiagnoses happened because there was no available model trained for this grammar using mal-rules – the main motivation to create this new model.

For completeness sake, the experiments provided in this section compare five different systems:

- **edERG (orig.)**: the mal-rule enhanced version of the ERG using its original parse ranking model;
- **edERG (new)**: the mal-rule enhanced version of the ERG using the new, mal-rule enhanced parse ranking model described in Section 7.2;
- **ERG**: the broad-coverage standard release of the ERG grammar (without mal-rules), using its original parse ranking model;

- **ERG → edERG (orig.)**: the same the two step approach currently employed by the LCC-APP. In this version, both the standard release of the ERG and the mal-rule enhanced version of the grammar use their original parse ranking models;
- **ERG → edERG (new)**: a two step approach similar to the one described in the item above, with the main difference that the mal-rule enhanced version of the grammar uses the new, mal-rule enhanced parse ranking model described in Section 7.2;

All systems were created using the 'trunk' branch of ERG's SVN repository[1] (Revision 29199) – the same version used to create the treebank that was used to produce the model being evaluated here. Each of the 1,000 sentences were parsed by each system, using ACE with the same parameters set to create the English portion of the Tembusu Treebank.[2] This evaluation is based on the top/best parse using the models described for each system. Table 9.3 shows a first summary of the results obtained by all five systems.

	Parsed w/o errors	Parsed w/ errors	No parse
edERG (orig.)	0.589	0.315	0.096
edERG (new)	0.703	0.201	0.096
ERG	0.920	0.001	0.079
ERG → edERG (orig.)	0.921	0.037	0.042
ERG → edERG (new)	0.921	0.037	0.042

Table 9.3: Parsing results of top/best parses for the test set (n=1,000)

Table 9.3 shows how the five systems classify the 1,000 sentences in three categories: 'Parsed w/o errors' (i.e., the top parse for a sentence did not include mal-rules), 'Parsed w/ errors' (i.e., the top parse for a sentence includes at least one mal-rule), and 'No parse' (i.e., the system was unable to produce a parse for that sentence). Mal-rules were identified as any lexical type or rule containing either the suffix `_rbst` or the suffix `_mal` in their names. The only exception to this definition was the exclusion of a single mal-rule: a rule that detects when improper capitalization of words happens in the middle of a sentence (i.e., `w_hasnoninitcap_dlr_rbst`). The reason for this exclusion was the fact that the NTUCLE includes assignments that introduce a new engineering solution. This almost always means the creation of a named product (cap-

[1] `http://svn.delph-in.net/erg/trunk`

[2] `--max-chart-megabytes=15000 --max-unpack-megabytes=16000 --timeout=300`

italized) that has never been heard of before, and that appears very frequently throughout that individual assignment. Because this mal-rule exists, the ERG insists on flagging as incorrect capitalization most product names comprised of common nouns. This happens disproportionately more often than any other mal-rule, and counting it as a mal-rule would further increase the numbers being presented here.

Table 9.3 shows that the system 'edERG' with its original model diagnoses 31.5% of the test set as sentences having at least one mal-rule. The system 'edERG' with the newly developed model is somewhat less greedy, diagnosing only 20.1% of these sentences as problematic. This decrease is welcomed, as it was known that the lack of a model developed using mal-rules was leading to many spurious diagnoses. However, both these numbers are still quite large when compared to the output of either of the two-step systems, which identify only 3.7% of this test set as problematic.

A noteworthy point that must be made concerns the 0.1% of sentences predicted to be problematic by the standard release of the ERG (earlier introduced as not being enhanced by mal-rules). However, as a matter of fact, the standard release of the ERG contains a very select number of mal-rules designed to accommodate, for example, some common misspellings. In this particular instance, this 0.1% refers to a single sentence that was correctly identified as problematic due to a mix-up between the possessive *its* and the contraction *it's* (as defined by the rule: `its_poss_2_u_rbst`).

Despite the fact that the trends shown in Table 9.3 point towards the right direction, it was important to confirm if the reduced number of sentences identified as problematic was being achieved by a decrease in misdiagnoses. As such, a second-level of the evaluation looked at the subset of 349 sentences (from the original 1000) that were flagged as problematic by at least one of the systems. Table 9.4 shows the summary of a second-level human analysis of this subset of sentences.

Table 9.4 shows the classification of each of the sentences in this subset into four categories: sentences that were correctly classified as problematic (i.e., that have at least one error); sentences that were incorrectly classified as problematic (i.e., that upon further inspection the sentence was deemed as having no errors); problematic sentences that were ignored by a system

	Correctly Problematic	Incorrectly Problematic	Ignored Problematic	Correctly Ignored
edERG (orig.)	**0.413**	**0.490**	0.020	0.077
edERG (new)	**0.361**	**0.215**	0.072	**0.352**
ERG	0.003	0.000	0.430	0.567
ERG → edERG (orig.)	0.095	0.011	**0.338**	0.556
ERG → edERG (new)	0.095	0.011	**0.338**	0.556

The values presented here make reference to all sentences included in the subset of 349 sentences described above; sentences without a parse were included in either the 'Ignored Problematic' or the 'Correctly Ignored' category, whether they were ungrammatical or grammatical sentences, respectively;

Table 9.4: English grammaticality/ungrammaticality judgments (n=349)

	Precision	Recall	F1
edERG (orig.)	0.457	0.954	0.618
edERG (new)	0.627	0.834	**0.716**
ERG	1.000	0.007	0.013
ERG → edERG (orig.)	0.892	0.219	0.351
ERG → edERG (new)	0.892	0.219	0.351

The values presented here make only reference to sentences that received a parse by that particular system;

Table 9.5: English ungrammaticality judgments: precision, recall and F1 measures

(i.e., a system classifies the sentence as grammatical but the sentence was not); and sentences that were correctly ignored by a system (i.e., a system correctly classifies the sentence as grammatical).

This second-level analysis was only concerned with the broad question of grammaticality. As such, these numbers report only on the grammar's ability to adequately classify a sentence as problematic (i.e., error detection), regardless of its ability to correctly diagnose what is wrong with part particular sentence (which will be discussed below). From the results shown in Table 9.4, it can be confirmed that the system 'edERG (new)', with the newly developed model, performs much better at correctly ignoring sentences without problems. While it is possible to observe a slight decrease in its ability to correctly classify problematic sentences as ungrammatical, it more than halves (from 49% to 21.5%) the number of sentences misclassified as ungrammatical.

The results presented in Table 9.4 also show that both two-step systems have a much lower incidence of misclassified sentences (of just around 1.1%). However, sustaining such a low rate of misdiagnoses does come at a cost – many problematic sentences are ignored by these two-step systems. This happens because the standard release of the ERG is able to parse most sentences without providing any warning (Table 9.3 shows it was able to parse 92% of the test set). As such, it becomes clear that using the standard release of the ERG might not be the most desirable filter, since many problematic sentences are actually being ignored without a warning.

Table 9.5 shows very similar results to those presented in Table 9.4, but from a system-relative perspective of precision, recall and F1 measures. The interpretations are the same, but through a different lens. It should not be a surprise that the system 'edERG (orig.)' offers the highest recall, since it was the system who classified most sentences as problematic. This, however, leads to a much lower precision score (i.e., too many false positives). The system trained with the new model, 'edERG (new)', shows a higher precision but a slightly worse recall measure. Overall, the F1 measure shows that the new system's increase in performance is still clear after balancing the losses of recall and the improvements in precision. But it is important to note that for the specific case of error detection and diagnosis, precision should be rated much higher than recall – as telling a student that a sentence is incorrect when it is not can have a much worse impact than not being able to recognize a sentence as problematic.

The results shown in Tables 9.4 and 9.5 show a very promising trend. This has made clear that using a model trained using a mal-rule enhanced treebank can indeed help reduce the number of sentences incorrectly classified as problematic – which was one of the main goals of its development. It is, however, unfortunate that this new parse-ranking model was not yet able to reduce the number of false positives to a level that might be tolerable in a pedagogical setting. Compounding the values in Table 9.3 with those of Table 9.4, it can be determined that the relative rate of false positives for the system 'edERG (new)' is around 37.3%. In other words, from all the sentences deemed as problematic by the system 'edERG (new)' (n=201), 37.3% of these sentences did not, in fact, have a problem. And while this rate is much better than the 54.3% of false positives generated by the system 'edERG (orig.)', the improvements are not sufficient to use the new model without a previous source of filtering, as it is being done by the iTELL's

LCC-APP (through the two-step approach).

Finally, the model was also evaluated with regard to its ability to properly diagnose the errors in a sentence. To achieve this, a further subset of 151 sentences was selected. This was the subset of sentences that had been confirmed to be problematic from the earlier subset of 349 sentences discussed above. Each system was then evaluated by its ability to select a parse that would diagnose a plausible error in that sentence. These results are shown in Table 9.6.

	Correct Diagnosis	Incorrect Diagnosis	Missed Diagnosis
edERG (orig.)	0.530	**0.424**	0.046
edERG (new)	**0.642**	**0.192**	0.166
ERG	0.007	0.000	0.993
ERG → edERG (orig.)	0.146	0.073	0.781
ERG → edERG (new)	**0.185**	0.033	0.781

The values presented in this table make reference to all sentences included in the subset of 151 sentences described above; sentences without a parse were included 'Missed Diagnosis', since they were all ungrammatical sentences;

Table 9.6: English error diagnosis (n=151)

	Precision	Recall	F1
edERG (orig.)	0.556	0.920	0.693
edERG (new)	**0.770**	0.795	**0.782**
ERG	1.000	0.007	0.013
ERG → edERG (orig.)	0.667	0.157	0.254
ERG → edERG (new)	**0.848**	**0.192**	0.313

The values presented here make only reference to sentences that received a parse by that particular system;

Table 9.7: English error diagnosis: precision, recall and F1 measures

Table 9.6 classifies each of the 151 problematic sentences in one of three categories: correct diagnosis (i.e., every mal-rule provided by the system's top/best parse pointed to a plausible correction for that sentence), incorrect diagnosis (i.e., the top/best parse provided by the system included at least one mal-rule that did not point to a plausible correction for the sentence in question), and missed diagnosis (i.e., the system provided either a parse without mal-rules or did not provide a parse at all).

An effort was made to consider multiple possible corrections for each particular sentence. The main criterion for this evaluation was that all mal-rules included in a parse had to lead to a

plausible correction. Sentences that presented a mix of plausible mal-rules with unlikely mal-rules were classified as 'incorrect diagnosis'. However, if a sentence that had more than one problem included only a single and plausible mal-rule (i.e., a fix to only part of that sentence), that sentence was classified as 'correct diagnosis'. The reason for this was that mal-rules are essentially used to generate corrective feedback for problematic sentences. As such, it was not deemed acceptable to receive a mix of adequate and inadequate diagnoses for the same sentence, since an inadequate diagnosis could lead students to make further mistakes. On the other hand, it was deemed acceptable to receive plausible feedback to only a subset of errors present in a sentence, as it could not only help the student improve the sentence, but it would also be difficult to argue that this would lead students to make further mistakes.

As can be seen in Table 9.6, once again, the system using the new model, 'edERG (new)', shows consistent results with those discussed previously. With respect to error diagnosis, the system 'edERG (new)' shows a clear reduction of incorrect diagnoses when compared with the same system using the original model, 'edERG (orig.)'. In addition, despite being able to catch fewer mistakes (as was discussed above), the system using the new model, 'edERG (new)', performs considerably better at correctly diagnosing problems in ungrammatical sentences. This increase in correct diagnoses is also measurable between both two-step systems. The version of the two-step system using the new and improved model was able to correctly diagnose roughly 3% more sentences.

Table 9.7 shows the same trend of results, but from a system-relative perspective of precision, recall and F1 measures, based only on the parses of a given system. The high increase in precision is consistent with both the improvement of correct diagnoses (i.e., true positives) and the decrease of incorrect diagnoses (i.e., false positives). And, once again, despite the earlier discussion that the F1 measure might not be the best measure for the specific case of error detection and diagnosis, it still shows that the new system's increase in error diagnosis performance persists even after balancing the losses of recall and the improvements in precision. And perhaps most relevant for the LLC-APP, Table 9.7 shows that the improvements in the ability to provide correct diagnoses was able to boost both the precision and recall of the two-step system using the new model.

Once again, the trends shown by this last set of results are very promising, as they show that using a model trained with mal-rules improves a grammar's ability to properly diagnose the errors contained in problematic sentences. However, the results achieved by employing the newly developed model still leave room for improvement.

One important note to be made about this room for improvement concerns itself with the size of the treebanks used to build this model. The treebanked portion of the NTUCLE contained only 4,900 sentences. And while this was no small endeavor, the original model that comes packed with the ERG is trained using the Redwoods treebank (Oepen et al., 2002),[3] – a treebank with over 85,000 sentences and in development for almost 20 years. Also related to the size of the treebank, this particular use case includes the added problem that errors are, in principle, less frequent that grammatical sentences. This, while taking into consideration the general Zipfian distribution of language, should be sufficient to understand that many classes of errors were likely missing from the treebanked portion of the NTUCLE, used to build this model. The number of ungrammatical structures effectively caps the learning potential of the model's ability to perform grammaticality judgments and error diagnosis. A good direction for future work would be to focus specifically on acquiring and enriching the NTUCLE treebank with sentences known to be problematic. Another avenue of investigation would be combining the NTUCLE data with the rest of Redwoods to build one large model.

9.3 Evaluating ZHONG's Coverage and Parse Ranking Model

This section reports on multiple evaluation experiments conducted using the new (2.0) version of ZHONG (covered in Chapter 6), and a new mal-rule enhanced parse-ranking model trained on a new Mandarin Chinese treebank (covered in Chapter 7). This section discusses coverage improvements, parse selection and the ability to correctly detect and diagnose common errors made by early learners of Mandarin Chinese.

The main development and evaluation sets in the experiments discussed here come from the NTU Corpus of Learner Mandarin Chinese (NTUCLM). As presented in Section 5.3, this corpus includes both grammatical and ungrammatical sentences naturally produced by early

[3] `https://github.com/delph-in/docs/wiki/RedwoodsTop`

learners of Mandarin Chinese – showcasing many of the problems these learners have during the learning process. Since the primary goal of ZHONG's development, as part of this thesis, was to correctly detect and diagnose grammatical errors present in this corpus, it should be no surprise that the NTUCLM will be taken as the primary evaluation set.

However, in addition to the NTUCLM, multiple subsets of data from the Mandarin Education Corpus (MEC, introduced in Section 5.4) were also used either in the development process or as evaluation data. Development data was used for both vocabulary collection, and to guide the improvement and development of syntactic analyses within ZHONG. Evaluation data, on the other hand, was not used during development, and was kept as strictly evaluation data instead. Table 9.8 shows how the data from these two corpora was split between development data and evaluation data, along with the set size (i.e., number of sentences) and average sentence length of each set.

	Set Name	Set Size	Avg. Sent. Length
Development Data	cmnedu	798	7.74
	tufs	1,531	6.46
	hsksc_01	175	5.71
	hsksc_02	200	7.92
	hsksc_03	81	9.42
	hsksc_04	200	10.51
	hsksc_05	200	11.89
	hsksc_06	157	13.48
	ntuclm_dev	2,013	7.00
Evaluation Data	hsksc_07	200	16.77
	hsksc_08	200	19.84
	hsksc_09	30	22.23
	hsksc_10	200	21.71
	hsksc_11	200	23.12
	hsksc_12	67	22.97
	tatoeba_01	10,000	8.47
	tatoeba_02	10,000	7.95
	tatoeba_03	10,000	7.94
	tatoeba_04	10,000	7.44
	tatoeba_05	7,216	7.23
	ntuclm_test	287	7.28

Average Sentence Length refers to the average number of words, including punctuation, contained by each sentence in that set;

Table 9.8: Split between Development and Evaluation Sets

The development data includes: the 'cmnedu' subcorpus (used in the development of ZHONG's first version, during Fan Zhenzhen's PhD); Mandarin Chinese data collected from the Tokyo University of Foreign Studies ('tufs'); data from the first three textbooks of the HSK Standard Course textbook collection ('hsksc_01' through 'hsksc_06'); and the development portion of the NTUCLM.

The evaluation data includes: data from the fourth and fifth textbooks of the HSK Standard Course collection ('hsksc_07' through 'hsksc_12'); data from the Mandarin Chinese subset of the Tatoeba corpus ('tatoeba_01' through 'tatoeba_05'), and the evaluation portion of the NTUCLM. For a brief reminder of how the NTUCLM corpus was divided between development and evaluation sets see the end of Section 5.3.

	ZHONG (v1.0)			**ZHONG (v2.0)**					
Set Name	% Parsed	Avg. Sent. Amb.	Avg. Sent. Len.	% Parsed	Δ	Avg. Sent. Amb.	Δ	Avg. Sent. Len.	Δ
cmnedu	91.9	261	6.6	94.9	+3.0	411.7	+150.4	6.7	+0.1
tufs	60.6	235	3.6	80.8	+20.2	222.7	-12.4	4.9	+1.3
hsksc_01	82.9	83	4.6	97.1	+14.3	50.7	-32.2	5.5	+0.9
hsksc_02	66.0	344	4.9	86.0	+20.0	319.2	-24.7	6.5	+1.7
hsksc_03	64.2	1,332	5.5	76.5	+12.4	894.6	-437.4	6.6	+1.0
hsksc_04	56.5	843	5.0	70.5	+14.0	808.4	-34.5	6.5	+1.5
hsksc_05	48.0	1,283	4.2	63.0	+15.0	1,318.2	+35.2	6.4	+2.1
hsksc_06	46.5	4,089	4.9	58.6	+12.1	2,516.5	-1572.2	6.2	+1.3
ntuclm_dev	65.4	43	4.4	**92.4**	**+27.0**	244.01	+201.3	6.4	+2.0

Average Sentence Ambiguity refers to the average number of available parses for a given sentence in that set; Average Sentence Length refers to the average number of words, including punctuation, contained by each parsed sentence in that set;

Table 9.9: ZHONG's Parsing Coverage (any parse) over the Development Data

Tables 9.9 and 9.10 show the results of running both ZHONG v1.0 (before any improvements were added by the work in this thesis) and ZHONG v2.0 (its current and most updated version). In this and all further comparisons throughout this section, ZHONG v2.0 makes reference to a version of ZHONG compiled with all available mal-rules.

The numbers in Tables 9.9 and 9.10 show improvements across all sets (both development and evaluation sets), concerning this grammar's ability to provide a parse for a given sentence. These improvements cannot only be measured using the percentage of sentences parsed per

	ZHONG (v1.0)			ZHONG (v2.0)					
Set Name	% Parsed	Avg. Sent. Amb.	Avg. Sent. Len.	% Parsed	Δ	Avg. Sent. Amb.	Δ	Avg. Sent. Len.	Δ
hsksc_07	30.0	2,422	2.6	35.0	+5.0	2,151	-271	3.3	+0.7
hsksc_08	24.0	4,281	2.9	32.0	+8.0	3,273	-1,009	4.1	+1.2
hsksc_09	26.7	1,980	3.4	33.3	+6.7	5,859	+3,878	4.8	+1.4
hsksc_10	31.5	6,070	4.1	33.0	+1.5	4,573	-1,497	4.2	+0.2
hsksc_11	20.0	5,240	2.7	23.5	+3.5	5,308	+68	3.5	+0.8
hsksc_12	16.4	3,528	2.8	19.4	+3.0	2,587	-940	2.9	+0.1
tatoeba_01	31.7	279	2.3	44.6	+12.9	289	+10	3.3	+1.0
tatoeba_02	33.3	246	2.3	47.0	+13.7	220	-26	3.3	+1.0
tatoeba_03	31.8	185	2.3	46.2	+14.4	196	+11	3.3	+1.1
tatoeba_04	31.9	179	2.1	48.5	+16.6	144	-35	3.2	+1.2
tatoeba_05	29.2	145	1.9	50.5	+21.4	110	-34	3.4	+1.5
ntuclm_test	58.9	17	4.1	**91.6**	**+32.8**	211	+193	6.7	+2.5

Average Sentence Ambiguity refers to the average number of available parses for a given sentence in that set; Average Sentence Length refers to the average number of words, including punctuation, contained by each parsed sentence in that set;

Table 9.10: ZHONG's Parsing Coverage (any parse) over the Evaluation Data

set, but also through the increase of average sentence length of parsed sentences. In addition to these desirable improvements, a different kind of improvement is also made evident by the fairly modest increases (and very often decreases) in average sentence ambiguity. Even though the average sentence ambiguity is expected to grow with respect to the length of a sentence, it can be seen that, in many sets, the average sentence ambiguity decreased despite the average length of parsed sentences having increased. This is especially meaningful when considering that ZHONG v2.0 also includes mal-rules, which could also be expected to contribute to spurious parses in some cases. Much of the work done to ZHONG during this thesis comprised tightening loose ends, many of which were generating spurious (or even incorrect) parses for many sentences. Being able to increasing the parse coverage while containing (and sporadically reducing) ambiguity should be seen as a great improvement.

Unsurprisingly, the sets that show greatest improvement are those belonging to the NTUCLM. This is, in part explained by the fact that the NTUCLM was the main development set, and in part explained by the fact that these sets also include ungrammatical sentences, which

the first version of ZHONG should have been unable to parse (even though this did not always happen). Looking specifically at Table 9.10, concerning the evaluation sets, it can be seen that the fairly large Tatoeba corpus shows an improvement between 12.9% and 21.4% – which is fairly consistent with the numbers shown for the development sets. The fairly modest improvements shown for the HSK Standard Course collection ('hsksc_07' through 'hsksc_12') can be explained by the increase in difficulty/complexity that is expected as the course progresses to higher levels of proficiency. This can also be confirmed by the increase in average sentence length across sets. This means that a decrease in parsing coverage was to be expected starting from set 'hsksc_01' (the easiest set) all the way to set 'hsksc_12' (the most difficult set in the corpus) – a fact that is confirmed. Given the graded proficiency levels inherent to these sets, they would be ideal to guide ZHONG's next stages of development.

The second set of experiments concern the use of the new parse-ranking model to provide ZHONG with the ability to adequately select the best parse from the pool of all available parses for any given sentence – both with and without errors. With this in mind, and following what was done for the evaluation of the English model, five systems were compared side-by-side:

- **ZHONG v1.0 (-)**: the first version of ZHONG, as released at the end of Fan Zhenzhen's PhD, without any parse-ranking model;
- **ZHONG v1.0 (orig.)**: the same system as described above, using the original parse-ranking model released with ZHONG's v1.0;
- **ZHONG v2.0 (-)**: the new version of ZHONG (v2.0) without any parse-ranking model;
- **ZHONG v2.0 (orig.)**: the new version of ZHONG (v2.0) using the original parse-ranking model released with ZHONG's v1.0;
- **ZHONG v2.0 (new)**: the new version of ZHONG (v2.0) using the new mal-rule enhanced parse-ranking model, developed as part of this thesis;

The first of the experiments involving the parse-ranking model aimed to verify the ability of the grammar to select the right parse among all available parses for a given sentence. This experiment uses the previously discussed metrics of labeled and unlabeled precision and recall, using the PARSEVAL algorithm (Black et al., 1991). Details about how to calculate these metrics were discussed in Chapter 7, as part of the evaluation of agreement between treebanks.

This experiment uses the entire evaluation set of the NTUCLM (which was treebanked, but not used to train the model being evaluated here), and measures the system's ability to pick the same (or similar) parse as the parse that was hand-picked during the treebanking process. Table 9.11 shows the results of this experiment.

System	Labeled Precision	Unlabeled Precision	Labeled Recall	Unlabeled Recall	Labeled F1	Unlabeled F1
ZHONG v1.0 (-)	0.439	0.591	0.422	0.566	0.431	0.578
ZHONG v1.0 (orig.)	0.470	0.601	0.449	0.573	0.459	0.587
ZHONG v2.0 (-)	0.780	0.934	0.812	0.984	0.796	0.959
ZHONG v2.0 (orig.)	0.920	0.977	0.927	0.986	0.924	0.981
ZHONG v2.0 (new)	**0.972**	**0.991**	**0.971**	**0.990**	**0.972**	**0.990**

The values presented here make only reference to sentences for which there was a gold tree saved in the treebank;

Table 9.11: Measuring ZHONG's best parse against the gold treebank

The numbers shown in Table 9.11 show that the system 'ZHONG v2.0 (new)' performed best in all categories – achieving scores close to perfect. It is worth to note that the gain from using the new parse-ranking model instead of the original model provides modest but still measurable improvements (of around 5% in precision, and about 1.4% for recall).

The low scores shown for ZHONG v1.0 derive from the fact that there were multiple sentences with trees saved in the treebank for which ZHONG v1.0 was unable to produce a parse, greatly penalizing the scores. These included both grammatical sentences (e.g., using syntactic constructions without a previous analysis) and ungrammatical sentences that ZHONG v1.0 was not designed to parse.

The next two sets of experiments follow a format very similar to what was described above, for English. First, all systems were compared by their ability to correctly diagnose a sentence as either problematic or not (i.e., error detection). Table 9.12 shows the results of this inquiry.

As can be derived from Table 9.12, 31.7% of the sentences in this evaluation set were problematic sentences – either due to a grammatical mistake, or due to an open ended range of other problems (e.g., 'semantic awkwardness'). Despite ZHONG's inability to deal with certain classes of problems, the numbers in this table follow the results collected by the NTUCLM tagging process (discussed in greater detail in Section 5.3). Table 9.12 shows trends similar to those shown for English. However, at their current stage of development, the Mandarin Chi-

	Correctly Problematic	Incorrectly Problematic	Ignored Problematic	Correctly Ignored
ZHONG v1.0 (-)	0.000	0.000	0.317	0.683
ZHONG v1.0 (orig.)	0.000	0.000	0.317	0.683
ZHONG v2.0 (-)	0.153	0.220	0.164	0.463
ZHONG v2.0 (orig.)	0.101	0.059	0.216	0.624
ZHONG v2.0 (new)	**0.129**	**0.017**	0.188	0.666

The values presented here make reference to all 287 sentences included in the NTUCLM evaluation set; sentences without a parse were included in either the 'Ignored Problematic' or the 'Correctly Ignored' category, whether they were ungrammatical or grammatical sentences, respectively;

Table 9.12: Mandarin Chinese grammaticality/ungrammaticality judgments (n=287)

	Precision	Recall	F1
ZHONG v1.0 (-)	1.000	0.000	0.000
ZHONG v1.0 (orig.)	1.000	0.000	0.000
ZHONG v2.0 (-)	0.411	0.484	0.444
ZHONG v2.0 (orig.)	0.630	0.319	0.423
ZHONG v2.0 (new)	**0.881**	**0.407**	**0.556**

The values presented here make only reference to sentences that received a parse by that particular system;

Table 9.13: Mandarin Chinese ungrammaticality judgments: precision, recall and F1 measures

nese systems show a much better ability to avoid misclassifying sentences as problematic. The best system was, as expected, 'ZHONG v2.0 (new)', which diagnosed 12.9% of the sentences as problematic (almost half of the total number of problematic sentences, 31.7%). This system only misclassified 1.7% of the sentences (n=5) as problematic – which is quite promising. On the other hand, this system was unable to correctly classify as ungrammatical about 18.8% of the sentences in this test set – which was not completely unexpected. A few ignored grammatical errors do not have associated mal-rules because were too infrequent in the NTUCLM to be deemed worth pursuing. In addition, many sentences tagged as problematic have problems that fall outside the scope of a syntactic check, such as sentences equivalent to '*I am France*' or '*This is the office's professor*'. As discussed in Chapter 3, it is possible to use the semantics produced by the grammar to check for certain classes of semantic errors. And even though detecting semantic errors was not part of the work developed in this thesis, they were still part of the test set used in this evaluation (because the NTUCLM includes all types of errors). In the

context of this evaluation, it is important to understand that many of the ignored problems were, in fact, not really grammatical problems (and hence not able to be detected by mal-rules).

Very similar trends and discussion arise from the results presented in Table 9.13, seen from the perspective of error detection, through precision, recall and F1 measures. The best system was, once again, 'ZHONG v2.0 (new)', with a fairly high precision of 88.1%. The fairly modest level of recall was already explained above.

A further analysis of what was lowering the precision revealed that a single problem was responsible to the large majority (i.e., 80%) of the misdiagnoses. The concerns surrounding this issue were discussed earlier, in Section 5.3, as part of the discussion of bare adjectival predicates in Mandarin Chinese (i.e., those missing a degree specifier). During this earlier discussion, it had been raised that this was not in fact an error when used contrastively. And while contrastive constructions were not expected appear in this corpus, there were a few instances that were correctly classified as grammatical during the tagging process. A further refinement of this mal-rule will be needed to include some awareness of contrastive constructions (e.g., through the presence of certain conjunctions).

Finally, Table 9.14 presents the results concerning each system's ability to provide the correct diagnosis in the presence of an ungrammatical sentence. This experiment used the subset of 91 sentences tagged as problematic in the NTUCLM evaluation set.

	Correct Diagnosis	Incorrect Diagnosis	Missed Diagnosis
ZHONG v1.0 (-)	0	0	1.000
ZHONG v1.0 (orig.)	0	0	1.000
ZHONG v2.0 (-)	0.143	0.341	0.516
ZHONG v2.0 (orig.)	0.275	0.044	0.681
ZHONG v2.0 (new)	**0.363**	**0.044**	0.593

The values presented in this table make reference to all sentences included in the subset of 91 problematic sentences contained in the NTUCLM evaluation set; sentences without a parse were included 'Missed Diagnosis', since they were all ungrammatical sentences;

Table 9.14: Mandarin Chinese error diagnosis (n=91)

The numbers in Table 9.14 show that the best system, 'ZHONG v2.0 (new)', provided the right diagnosis to 36.3% of ungrammatical sentences, while about 59.3% were either not parsed or not diagnosed as problematic. This is consistent with what was discussed above, concerning

	Precision	Recall	F1
ZHONG v1.0 (-)	1.000	0.000	0.000
ZHONG v1.0 (orig.)	1.000	0.000	0.000
ZHONG v2.0 (-)	0.295	0.302	0.299
ZHONG v2.0 (orig.)	0.862	0.357	0.505
ZHONG v2.0 (new)	**0.892**	**0.471**	**0.617**

The values presented here make only reference to sentences that received a parse by that particular system;

Table 9.15: Mandarin Chinese error diagnosis: precision, recall and F1 measures

the reason why a fairly high number of sentences tagged as problematic might have been been ignored. The same trends are also evident when inspecting Table 9.15, with the system 'ZHONG v2.0 (new)' showing the highest precision, recall and F1 measures across the board.

Both tables make evident that using the new mal-rule enhanced parse-ranking model provides a considerable improvement in the system's ability to provide a correct diagnosis. A further inspection of the sentences responsible for limiting the precision in the system 'ZHONG v2.0 (new)' revealed that about half of them had the correct diagnosis available in the parse inventory (i.e., it is likely that there were not enough instances for the model to learn properly), and the other half of these sentences needed a mal-rule that was not available in the grammar (i.e., a new mal-rule would have to be added before the system could attempt to provide an adequate diagnosis).

9.4 Summary

The results reported in this chapter provide a clear statement of the many contributions of this thesis. Not only did the results from the experiment with the LCC-APP corroborate the idea that systems like this can be successfully incorporated in language/writing courses, but further experiments with the novel mal-rule enhanced parse-ranking models also showed promising results in their ability to drastically improve the precision of these systems. An early version of the system was shown to improve students' essays 84% of the time (Table 9.2). I then improved the detection of ungrammatical phenomena in English from a precision of 45.7% to 62.7% with

an improved language model trained on mal-rules, at the cost of a small loss in recall (Table 9.5).

This chapter also reported on the work done to extend ZHONG as a high-quality parser of Mandarin Chinese, showing both considerable coverage improvements for standard language, and the creation of a promising error detection and diagnosing system for early proficiency learners of Mandarin Chinese. By extending the grammar I increased the coverage on held out test sentences from 58.9% to 91.6% (Table 9.10). With an improved language model trained on mal-rules I raised the precision for the detection of ungrammatical phenomena from 63.0% to 88.1% while simultaneously improving the recall (Table 9.15).

Chapter 10

Discussion

This thesis has shown that mal-rules are a viable alternative to design error detection systems in the context of Computer Assisted Language Learning. This was demonstrated through the entire process of creating a successful educational application, from the collection of learner data to the development of an end-to-end application that has benefited thousands of NTU students.

The LCC-APP has, in addition to helping improve students' writing, been extremely well received by students, lecturers and school administrators. Based on the early success of the tested prototype, the team of lecturers at the Language and Communication Centre decided to continue supporting its development by integrating the LCC-APP in the curriculum of the Engineering Communication course at NTU.

School administrators have also shown a lot of interest in this work, funding this project internally through two consecutive EdeX Teaching and Learning Grants, which are designed specifically to encourage research and develop methods or tools to improve student learning. Most recently, the team was also encouraged by the university to apply for a Ministry of Education Tertiary Education Research Fund (MOE-TRF), which supports the same kind of research but at a greater scale.

In the past 20 months, COVID-19's world-wide pandemic has shown that we might be reaching a point where computer mediated education is no longer simply a 'choice for good', but a necessity for the preservation of social development. Face-to-face teaching is highly susceptible to interference from natural and social crises. The future of education relies not only on an

increase of digital literacy among educators, but also on the existing infrastructure to support digital and distance learning. This thesis offers a viable option to start dealing with the absence of robust infrastructure to support central aspects of pedagogy and assessment in online language teaching and learning.

10.1 Limitations and Future Work

In this section I will discuss some aspects that can be seen as limiting the impact of this thesis, as well as possible future work directions that can help solve some of these issues.

Learner Data Profiling

One of the limitations that can be raised about the methodology used in this thesis is the limited ability to profile learners by their backgrounds. As discussed earlier in this thesis, first language transfer is known to play an important role in language learning (Gass, 1988), which includes the errors learners are likely to make.

While it is true that exemplary learner corpora often include socio-demographic information about the students from which the data was collected, Singaporean universities are fairly sensitive about racial and nationality profiles, which are often tied to language backgrounds. For the English data, the team of lecturers I worked with was also quite averse to the idea of collecting any sort of data from the students that could be used to trace the author of collected assignments. For Mandarin, collecting any kind of socio-demographic data was also disallowed under the Institutional Review Board (IRB) covering the project.

In the future, I agree it would be interesting to collect some socio-demographic data that could help better understand certain groups of students. It is fairly likely that students whose mother tongue is Mandarin Chinese make slightly different errors than those whose mother tongue is Malay, or Tamil – all common languages among NTU's student population. This could eventually help build slightly different grammars, with different sets of mal-rules and possibly even different parse-ranking models for different student populations.

The NTUCLE and NTUCLM will continue to be developed even after the conclusion of

this thesis, and collecting some socio-demographic information will definitely be a matter of consideration in future data collection.

Balancing Coverage and Precision in Grammar Engineering

Using symbolic parsers and mal-rules to perform error detection also poses some limitations tied to quality of the parser, which can be discussed in reference to coverage and precision.

Concerning coverage, a straightforward problem is the fact that symbolic parsers are likely to be unable to produce analyses for every sentence. Coverage is obviously tied to how much work has been put into the parser, but it is also very dependent on work in theoretical linguistics. If a phenomena in a certain language is not yet well understood, it is unlikely to have a robust implementation in a symbolic parser.

These problems are also related to the concept of precision. Whereas coverage denotes the inability to provide any parse for a given sentence, precision here relates to the fact that some sentences may receive a parse that is partially or even completely incorrect. In the worst case, this includes providing analyses for ungrammatical sentences without using mal-rules – which should never happen.

Unfortunately, '*all grammars leak*'. This is a piece of wisdom shared by the famous linguist/anthropologist Edward Sapir (1921, p29) that has been reappropriated as a ground truth within the grammar engineering community. When a certain unwanted interaction (i.e., a *syntactic bug*) is fixed, or coverage for a missing syntactic construction is added to the grammar, a number of other unexpected and unwanted interactions (i.e., 'leaks') usually appear. And this means that no grammar is ever perfect.

Large and mature grammars, like the ERG, usually struggle less with problems such as coverage and precision. Relatively young, medium-sized grammars like ZHONG are more flawed and incomplete.

In order to design mal-rules that detect certain classes of error, a grammar must first have a sound implementation of the phenomena. Unfortunately, ZHONG had many phenomena that only had a preliminary analysis that covered the more frequent manifestations of phenomena. This made it necessary for me to improve and extend ZHONG's linguistic repertoire before I

could add mal-rules. This work is non-trivial and takes a lot of time. Because of this, not all errors identified in the NTUCLM could end up detected by the grammar. Some areas that are either underdeveloped or completely missing and that adversely affected this thesis include: classifiers and measure words, comparative constructions, and verb complements (e.g. directional, resultative, quantitative, potential, etc.).

However, it is also important to note that while I would have preferred ZHONG to be more robust, this thesis demonstrates that even immature computational grammars can be used to capture certain classes of errors. This means that, using systems like the LinGO Grammar Matrix project (Bender et al., 2002), it is possible to develop computational grammars robust enough to support the development of CALL systems focused on early proficiency learners of a language at a fairly low cost.

As one of the current maintainers of ZHONG, I feel committed to continue working on improving this grammar's coverage and precision, supporting the design and implementation of additional mal-rules and, eventually, transforming ZHONG into a system similar to the LCC-APP. This commitment has been partially supported by the award of a Marie Skłodowska-Curie Individual Fellowship that will fund ZHONG's development for the next two years. I will be looking specifically at improving the analysis of Mandarin Chinese noun phrases, along with expanding ZHONG's mal-rule repertoire.

Precision of Error Detection and Diagnosis

Despite the large improvements in error detection and diagnosis provided by the new mal-rule enhanced parse-ranking models presented in this thesis, it would be fair to point out that these improvements might not be sufficient to support widespread use of these systems in a classroom setting (especially if unsupervised).

As it has been discussed earlier in Chapter 9, F1 measures are designed to measure intrinsic performance of systems – in this case, the balance between precision and recall. For educational applications, however, precision should be taken as the most important measure.

The best English system (i.e., the two-step approach using the mal-rule enhanced model) reported 89.2% precision in error detection and 84.8% precision in error diagnosis. For Mandarin

Chinese, the best system (i.e., using the mal-rule enhanced model) reported 88.1% precision in error detection and 89.2% in error diagnosis. These values are comparable and fairly competitive with some of the best scores reported in recent shared tasks on Grammatical Error Detection and Correction. For reference, the best ranked system performing token-level error detection in the 2019's shared task on English Grammatical Error Correction (Bryant et al., 2019) reported 91.15% precision.[1]

Interpreting these scores, this means that around one in every ten sentences diagnosed as problematic is actually a good sentence. And for sentences that are actually problematic, both grammars (i.e., the ERG and ZHONG) provide a diagnosis that is also not truly helpful one out of ten times (e.g., a diagnosis that predicts a correction that is too far from the student's intended meaning). This is a challenge when we think of incorporating these systems in a classroom.

For the specific case of the LCC-APP, students were not only aware that the system was still in development (and therefore likely to incorrectly diagnose a few sentences), but were also encouraged to clarify any doubts through a network of trained peer tutors (available to the entire university), that could answer any questions they had concerning the error diagnosis.

As discussed earlier in this thesis, even human lecturers make mistakes when diagnosing errors (Lee, 2004). As such, attaining perfect precision might well be an unattainable goal. However, in order to gain the trust of language educators, it is important to push further, and raise these results as high as possible.

The simplest solution may well lie in the fact that the models I trained for this thesis were fairly small, especially in comparison to the model normally used by the ERG. The mal-rule enhanced English model was trained on 3,737 treebanked sentences. In comparison, the standard ERG model is trained using the Redwoods treebank, which contains more than 85,000 sentences. As such, it is plausible to assume that using more data to train a mal-rule enhanced model would improve the precision of error detection. Fortunately, this data is readily available through the NTUCLE-X, which contains more than 19,000 sentences not yet treebanked.

An alternative solution, which can be seen as complementary, would be to take advantage of the ambiguity generated by the parser to create a complex output – an idea that has been

[1] `https://www.cl.cam.ac.uk/research/nl/bea2019st/results/r/Kakao&Brain.html`

introduced in Chapter 9. Currently, precision is measured from a single (i.e., highest ranked) parse as the source of feedback. This parse not only decides if a sentence is classified as ungrammatical, through the presence of mal-rules, but also defines the error diagnoses through the exact set of mal-rules used to parse the sentence.

However, as discussed in the introductory chapters of this thesis, error detection and correction is an implicitely ambiguous problem. A single ungrammatical sentence can have multiple corrections – some more plausible, and others less so. Currently, if the model offers a parse with mal-rules for a grammatical sentence, or suggests an implausible correction for a problematic sentence, it hurts the system performance. This is true even if the perfect parse is ranked second by the model.

In the future, I would like to see if working with more than the single best parse produced by the models could improve these systems' performance. This could mean, for example, not diagnosing a sentence as ungrammatical if the second or third best parse predict that the sentence would be grammatical. And it could also include providing more than one diagnosis for a sentence deemed problematic.

From an educational perspective, providing a small set of possible diagnoses containing a few viable corrections might be sufficient to gain the trust of language educators, even if, from time to time, this set of corrections also includes a few unlikely suggestions.

Finally, a third method to try to boost the precision of error detection and diagnosis is to include some lexical semantic knowledge in the parse-ranking models.

The necessity of lexical semantic knowledge in error detection is well documented. Flickinger and Yu (2013), for example, note that for certain nouns (e.g., *society*), the use of a determiner is mostly determined by which sense the word has in the sentence. When used with the sense of *an extended social group having a distinctive cultural and economic organization*[2] this noun appears most often without a determiner (while not strictly ungrammatical, a determiner might be considered spurious and should be detected). But when used with the sense *a formal association of people with similar interests* it requires a determiner if used in the singular form. Currently, the ERG doesn't distinguish these two senses, and is hence incapable of diagnosing

[2] `https://en-word.net/lemma/society`

a spurious determiner for the first sense.

A solution for this problem would be to enrich grammars with lexical semantic knowledge, for example, as is provided by wordnets. There seems to be a fair amount of interest in this topic within the DELPH-IN community which has seeded some possible approaches (Le, 2019). Both English and Mandarin Chinese have two very good resources for this – the Princeton WordNet (PWN, Miller, 1995; Fellbaum, 1998) and the Chinese Open Wordnet (Wang and Bond, 2013).

In the future, I would also like to work towards this goal – bringing lexical semantics within computational grammars, to hopefully improve their error detection precision.

Learning and Corrective Feedback

The fourth and final topic I will cover in this section is concerned with the fact that the blended learning experiment described for the LCC-APP in Chapter 9 cannot be used to claim that learning actually took place – which is entirely true.

In the absence of pre- and post-tests focusing on measuring a learning goal, it is impossible to prove that students can actually learn something from the interaction with this system. What is clear and evident, however, is that the system helps students improve their writing, and that there are benefits associated with it.

But despite not proving any learning results, I **do** expect that learning took place. There is plenty of evidence that using corrective feedback promotes learning (Lalande, 1982; Ashwell, 2000; Van Beuningen, 2010; Ferris, 2010). And while I can not claim this thesis provides evidence of learning, I would like to argue it provides something that is very relevant to this discussion: a ready-to-use platform to promote and conduct future research on corrective feedback design and learning.

The LCC-APP is a robust platform capable of error detection at various levels of specificity. Mal-rules can be as specific as one wants to design and constrain them. This provides an ideal setting to test multiple kinds of corrective feedback, from the most indirect forms to the most direct (i.e., offering a correction for the error).

Within the context of this thesis, as discussed in Chapter 8, I allowed the LCC team to drive the design of corrective feedback. They were the practitioners, they knew their students,

and they were best positioned to make an educated guess about what kind of feedback would work best. However, it became clear through the discussion with LCC lecturers, and further confirmed by the results of a student survey discussed in Chapter 9, that there is little consensus of what should be the right form of the corrective feedback.

As previously discussed, one possibility for why the corrective feedback was not always clearly understood could be the use of linguistic jargon. An alternative reason could be the fact that not all students are native English speakers and would have preferred feedback in their mother tongue. It is hard to know for sure without further researching this topic.

In the future, I would like to use the LCC-APP to measure the learning impact of different types of corrective feedback. In addition to be able to try different kinds of text-based feedback, the technology behind the LCC-APP would also enable trying new kinds of feedback.

One example of a unique kind of feedback would be to provide more than one diagnosis for the same sentence, as suggested above. From a pedagogical perspective, being presented with more than one correction might actually be productive. Students would start by becoming aware that a problem exists, but would then have to engage the fact that a sentence can be corrected in multiple ways. Comparing different diagnoses could also lead to some sort form of implicit learning of certain syntactic interactions that may be difficult to explain in words.

Another unique form of feedback would be to use DELPH-IN's semantic-based machine translation technology (Bond et al., 2005, 2011) to provide a space of interaction to confirm of the intended meaning before diagnosing an error. A fully detailed proposal of this kind of corrective feedback can be found in Morgado da Costa et al. (2016).

Even though this thesis did not set out to investigate the learning impact of the design and quality of corrective feedback, it opens the door to new and exiting research opportunities in this field.

Chapter 11

Conclusions

In a few words, the goal of this thesis was to determine the relevance and viability of grammatical error detection and correction using high-quality, hand-curated, and theoretically sound symbolic parses. And, to this end, I believe this thesis successfully accomplished its goals.

To a certain extent, this thesis is a comment on the so called *deep learning tsunami* in Natural Language Processing – or the Computational Linguistics without the *linguistics*. While I truly believe in the benefits of using stochastic models for certain aspects and layers of Natural Language Processing, it is also important to ask ourselves what are the costs of foregoing hundreds of years of research in linguistics in favor of purely data-driven statistical methodologies.

It is my strong personal conviction that mal-rules are the way forward to build applications for Computer Assisted Language Learning (CALL). The road is long, but it is also extremely appealing. Members of the DELPH-IN Consortium often use the motto *'slow and steady wins the race'*. Personally, I see this both as a reminder that quality takes effort and time to achieve, but also that it is important to enjoy the journey. Within deep linguistic processing, this journey invites us to see language as a puzzle waiting to be solved, and reminds us that we must draw pleasure at attempting to understand its inner workings.

I believe we must start re-engaging CALL as a true interdisciplinary field, listening and exchanging ideals and expectations with all stakeholders – students, teachers, linguists and computer scientists. As I have discussed in the beginning of this thesis, it seems fairly clear that the field of Grammatical Error Detection and Correction has long since lost touch with its roots.

One of the most engaging parts of my dissertation was to work closely with NTU lecturers, understand their expectations and often attempt to explain the difficulty of the task at hand. It was a journey of mutual learning. I found that working with symbolic parsers was productive in managing expectations of quality and performance. Instead of working with a black-box (i.e., how statistical systems are usually described), lecturers could actually understand the amount of work and knowledge required to build a symbolic parser. Any shortcomings in the parser was a knowledge gap, or work that had not yet been done. And it was also fairly easy to understand why seemingly implausible parses or corrections were being offered by the system. Ambiguity is, after all, a very real problem – and not only for computational systems.

I have personally learned a lot during this thesis, and I look forward to continue on this road, attempting to *close the divide*, and enjoying the journey while I attempt to *solve* the puzzle of language.

Major Thesis Contributions

This interdisciplinary thesis has made multiple contributions to the fields of Linguistics, Computational Linguistics, Corpus Linguistics, and Computer Assisted Language Learning. Its main contributions can be listed as follow:

1. I provided a critical analysis of the current state of Grammatical Error Detection and Correction, along with the motivation and theoretical foundation to explore mal-rules as a viable alternative.

2. I built two new learner corpora, one for English (NTUCLE) and one for Mandarin Chinese (NTUCLM). In total, these corpora contain more than 11,800 sentences, hand-tagged for grammatical errors. A large portion of these sentences can and will be released under an open license.

3. I contributed to ZHONG, an open-source computational grammar for Mandarin Chinese. I improved and added a variety of linguistic phenomena. Many of these included theoretical contributions in the analysis of Mandarin Chinese using HPSG. Among these, separable verbs and interactions between aspect and negation deserve a notable mention.

Experiments on held-out learner and educational data showed that my contributions expanded ZHONG's coverage from 58.9% to 91.6%.

4. I implemented more than 60 mal-rules in ZHONG, effectively transforming it into an error detection system. These mal-rules cover a wide number of common mistakes made by early learners of Mandarin Chinese, chosen through the analysis of the NTUCLM.

5. I developed a new multilingual treebank of learner and educational data (the Tembusu Treebank), that includes data for English and Mandarin Chinese. This treebank is unique, as it is the first known treebank to use mal-rules to provide syntactic analyses for ungrammatical sentences. In total, it contains 7,983 hand-tagged sentences (4,246 sentences for Mandarin Chinese, and 3,737 sentences for English). A large portion of this treebank can and will be released under an open license.

6. Using the Tembusu treebank, I created two new mal-rule enhanced parse-ranking models to improve error detection and diagnosis for the ERG and ZHONG. Using its new model, the ERG error detection precision increased from 45.7% to 62.7%. And ZHONG's increased from 63% to 88.1%.

7. I developed an open-source web application that exploits mal-rules in the English Resource Grammar to provide immediate feedback about a wide range of language problems common among NTU students. This system was used by thousands of NTU students, and tested in a blended learning experiment with the help of NTU's Language and Communication Centre. The results of this experiment were very promising, showing that this application helped students improve their assignments 84% of the time.

8. Finally, I developed a second open-source web application that uses the same error detection technology, along with elements of improvisational comedy, to build fun an engaging language games. This application is currently in early stages of beta testing.

Part IV:

Bibliography

Bibliography

Mustafa Al Emran and Khaled Shaalan. 2014. A survey of intelligent language tutoring systems. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 393–399. IEEE.

Tim Ashwell. 2000. Patterns of teacher response to student writing in a multiple-draft composition classroom: Is content feedback followed by form feedback the best method? *Journal of second language writing*, 9(3):227–257.

Timothy Baldwin, Mark Dras, Julia Hockenmaier, Tracy Holloway King, and Gertjan van Noord. 2007. The impact of deep linguistic processing on parsing technology. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 36–38.

Lina María Castro Benavides, Johnny Alexander Tamayo Arias, Martín Darío Arango Serna, John William Branch Bedoya, and Daniel Burgos. 2020. Digital transformation in higher education institutions: a systematic literature review. *Sensors*, 20(11):3291.

Emily M. Bender, Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar customization. *Research on Language and Computation*, 8(1):23–72.

Emily M Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the 2002 workshop on Grammar engineering and evaluation-Volume 15*, pages 1–7. Association for Computational Linguistics.

Emily M Bender, Dan Flickinger, and Stephan Oepen. 2008. Grammar engineering for linguistic hypothesis testing. In *Proceedings of the Texas Linguistics Society X Conference: Computational linguistics for less-studied languages*, pages 16–36.

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2011. Grammar engineering and linguistic hypothesis testing: Computational support for complexity in syntactic analysis. In Emily M. Bender and Jennifer E. Arnold, editors, *Language from a Cognitive Perspective: Grammar, Usage and Processing*, pages 5–29. CSLI Publications, Stanford, CA.

Emily M Bender, Dan Flickinger, Stephan Oepen, Annemarie Walsh, and Timothy Baldwin. 2004. Arboretum: Using a precision grammar for grammar checking in CALL. In *InSTIL/ICALL Symposium 2004*.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. Universal Dependencies for learner English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 737–746, Berlin, Germany. Association for Computational Linguistics.

Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Ezra Black, Steven Abney, Dan Flickinger, Claudia Gdaniec, Ralph Grishman, Phil Harrison, Donald Hindle, Robert Ingria, Frederick Jelinek, Judith L Klavans, et al. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.

Barry Boehm. 1988. A spiral model of software development and enhancement. *IEEE Computer*, 21(5):61–71.

Francis Bond and Ryan Foster. 2013. Linking and extending an Open Multilingual Wordnet. In *51st Annual Meeting of the Association for Computational Linguistics: ACL-2013, Sofia*, pages 1352–1362.

Francis Bond, Sanae Fujita, Chikara Hashimoto, Kaname Kasahara, Shigeko Nariyama, Eric Nichols, Akira Ohtani, Takaaki Tanaka, and Shigeaki Amano. 2004. The Hinoki treebank. a

treebank for text understanding. In *International Conference on Natural Language Processing*, pages 158–167. Springer.

Francis Bond, Sanae Fujita, and Takaaki Tanaka. 2008. The Hinoki syntactic and semantic treebank of Japanese. *Language Resources and Evaluation*, 42(2):243–251.

Francis Bond, Luís Morgado da Costa, and Tuan Anh Le. 2015. IMI – A multilingual semantic annotation environment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, System Demonstrations (ACL 2015)*, pages 7–12, Beijing, China.

Francis Bond, Hiroki Nomoto, Luis Morgado da Costa, and Arthur Bond. 2020. Linking the TUFS basic vocabulary to the Open Multilingual Wordnet. In *Proceedings of the 12th Conference on Language Resources and Evaluation*, Marseille, France. European Language Resources Association (ELRA).

Francis Bond, Stephan Oepen, Eric Nichols, Dan Flickinger, Erik Velldal, and Petter Haugereid. 2011. Deep open-source machine translation. *Machine Translation*, 25(2):87–105.

Francis Bond, Stephan Oepen, Melanie Siegel, Ann Copestake, and Dan Flickinger. 2005. Open source machine translation with DELPH-IN. In *Open-Source Machine Translation: Workshop at MT Summit X*, pages 15–22, Phuket.

Anthony Bruton. 2009. Designing research into the effects of grammar correction in L2 writing: Not so straightforward. *Journal of Second Language Writing*, 18(2):136–140.

Christopher Bryant, Mariano Felice, Øistein E Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? 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), pages 697–707, Beijing, China. Association for Computational Linguistics.

Chao-Huang Chang and Gilbert K Krulee. 1991. Resolution of ambiguity in Chinese and its application to machine translation. *Machine translation*, 6(4):279–315.

Li-ping Chang and Keh-jiann Chen. 1995. The CKIP part-of-speech tagging system for modern Chinese texts. In *Proceedings of 1995 international conference on computer processing of oriental languages*, pages 172–175.

Pi-Chuan Chang, Michel Galley, and Christopher D Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the third workshop on statistical machine translation*, pages 224–232.

Keh-Jiann Chen, Chu-Ren Huang, Li-Ping Chang, and Hui-Li Hsu. 1996. Sinica corpus: Design methodology for balanced corpora. In *Proceedings of the 11th Pacific Asia Conference on Language, Information and Computation*, pages 167–176.

Keh-Jiann Chen and Shing-Huan Liu. 1992. Word identification for Mandarin Chinese sentences. In *COLING 1992 Volume 1: The 15th International Conference on Computational Linguistics*.

Keh-Jiann Chen, Chi-Ching Luo, Ming-Chung Chang, Feng-Yi Chen, Chao-Jan Chen, Chu-Ren Huang, and Zhao-Ming Gao. 2003. Sinica treebank. In *Treebanks*, pages 231–248. Springer.

Nancy F Chen, Rong Tong, Darren Wee, Peixuan Lee, Bin Ma, and Haizhou Li. 2015. iCALL corpus: Mandarin Chinese spoken by non-native speakers of European descent. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of ACL '98/EACL '98*, page 16–23, USA. Association for Computational Linguistics.

Ann Copestake. 2002. *Implementing typed feature structure grammars*, volume 110. CSLI Publications.

Ann Copestake. 2007. Semantic composition with (robust) minimal recursion semantics. In *Proceedings of the Workshop on Deep Linguistic Processing*, pages 73–80. Association for Computational Linguistics.

Ann Copestake. 2009. Slacker semantics: Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 1–9, Athens.

Ann Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second Linguistic Resources and Evaluation Conference*, pages 591–600, Athens, Greece.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332.

Flávio M Costa, José Carlos L Ralha, and Célia G Ralha. 2006. Aprendizagem de língua assistida por computador: Uma abordagem baseada em HPSG. *Revista Brasileira de Informática na Educação*, 14(1).

Berthold Crysmann. 2003. On the efficient implementation of German verb placement in HPSG. In *Proceedings of RANLP 2003*, pages 112–116, Borovets, Bulgaria.

Berthold Crysmann. 2005. Relative clause extraposition in German: An efficient and portable implementation. *Research on Language and Computation*, 3(1):61–82.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013a. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013b. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *BEA@ NAACL-HLT*, pages 22–31.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on*

Building Educational Applications Using NLP, pages 54–62. Association for Computational Linguistics.

Robert Dale and Adam Kilgarriff. 2011. Helping our own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249. Association for Computational Linguistics.

Vidas Daudaravicius, Rafael E Banchs, Elena Volodina, and Courtney Napoles. 2016a. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–62.

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. 2016b. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–62, San Diego, CA. Association for Computational Linguistics.

Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15. ACM.

Rebecca Dridan and Stephan Oepen. 2012. Tokenization: Returning to a long solved problem —a survey, contrastive experiment, recommendations, and toolkit. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 378–382.

Thomas Ernst. 1995. Negation in Mandarin Chinese. *Natural Language & Linguistic Theory*, 13(4):665–707.

Elham Eslami. 2014. The effects of direct and indirect corrective feedback techniques on EFL students' writing. *Procedia-Social and Behavioral Sciences*, 98:445–452.

Zhenzhen Fan. 2019. *Building an HPSG Chinese grammar (Zhong)*. Ph.D. thesis, Nanyang Technological University.

Zhenzhen Fan, Sanghoun Song, and Francis Bond. 2015a. Building ZHONG, a Chinese HPSG Shared-Grammar. In *Proceedings of the 22nd international conference on Head-driven Phrase Structure Grammar*, pages 96–109.

Zhenzhen Fan, Sanghoun Song, and Francis Bond. 2015b. An HPSG-based shared-grammar for the Chinese languages: ZHONG [|]. In *Proceedings of the Grammar Engineering Across Frameworks (GEAF) Workshop*, pages 17–24.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT Press Cambridge.

Dana R. Ferris. 2010. Second language writing research and written corrective feedback in SLA: Intersections and practical applications. *Studies in Second Language Acquisition*, 32(2):181–201.

Dan Flickinger. 2000. On building a more effcient grammar by exploiting types. *Natural Language Engineering (Special Issue on Efficient Processing with HPSG)*, 6(1):15–28.

Dan Flickinger. 2010. Prescription and explanation – using an HPSG implementation to teach writing skills. In *Invited talk, HPSG Conference*.

Dan Flickinger. 2011. Accuracy vs. robustness in grammar engineering. *Language from a cognitive perspective: Grammar, usage, and processing*, 201:31–50.

Dan Flickinger, Ann Copestake, and Ivan A Sag. 2000. HPSG analysis of English. In *Verbmobil: Foundations of speech-to-speech translation*, pages 254–263. Springer, Berlin, Germany.

Dan Flickinger, Michael Goodman, and Woodley Packard. 2016. UW-Stanford System description for AESW 2016 shared task on grammatical error detection. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 105–111.

Dan Flickinger and Justin Chunlei Yang. 2011. ManGO: Mandarin Grammar Online. In *http://www.delph-in.net/2011/mango.pdf*, Seattle, USA. Delph-in Summit.

Dan Flickinger and Jiye Yu. 2013. Toward more precision in correction of grammatical errors. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 68–73.

Johann Gamper and Judith Knapp. 2002. A review of intelligent CALL systems. *Computer Assisted Language Learning*, 15(4):329–342.

Rao Gaoqi, Baolin Zhang, Xun Endong, and Lung-Hao Lee. 2017. IJCNLP-2017 task 1: Chinese grammatical error diagnosis. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 1–8.

Ignacio Garcia. 2013. Learning a language for free while translating the web. Does Duolingo work? *International Journal of English Linguistics*, 3(1):19.

Susan M. Gass. 1988. *Second Language Acquisition and Linguistic Theory: The Role of Language Transfer*, pages 384–403. Springer Netherlands, Dordrecht.

Michael Wayne Goodman. 2019. A Python library for deep linguistic resources. In *2019 Pacific Neighborhood Consortium Annual Conference and Joint Meetings (PNC)*, Singapore.

Sylviane Granger. 2003. The International Corpus of Learner English: a new resource for foreign language learning and teaching and second language acquisition research. *TESOL Quarterly*, 37(3):538–546.

Hans Werner Guesgen and Joachim Hertzberg, editors. 1992. *Constraint relaxation*, pages 41–56. Springer Berlin Heidelberg, Berlin, Heidelberg.

Joy Paul Guilford. 1967. *The nature of human intelligence*. McGraw-Hill.

L Kirk Hagen. 1994. Unification-based parsing applications for intelligent foreign language tutoring systems. *CALICO Journal*, 12(2&3):5–31.

Juho Hamari, Jonna Koivisto, and Harri Sarsa. 2014. Does gamification work? a literature review of empirical studies on gamification. In *2014 47th Hawaii International Conference on System Sciences*, pages 3025–3034. IEEE.

Chikara Hashimoto, Francis Bond, and Dan Flickinger. 2007. The Lextype DB: A web-based framework for supporting collaborative multilingual grammar and treebank development. In *International Workshop on Intercultural Collaboration*, pages 76–90. Springer.

Chikara Hashimoto, Francis Bond, Takaaki Tanaka, and Melanie Siegel. 2008. Semi-automatic documentation of an implemented linguistic grammar augmented with a treebank. *Language resources and evaluation*, 42(2):117–126.

Trude Heift. 1998. An interactive intelligent language tutor over the internet. In *Proceedings of ED-MEDIA, ED-TELECOM 98, World Conference on Education Multimedia and Educational Telecommunications*, volume 2, pages 508–512.

Lars Hellan, Tore Bruland, Elias Aamot, and Mads H Sandoy. 2013. A Grammar Sparrer for Norwegian. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013), Oslo, Norway. NEALT Proceedings Series*, volume 16.

Lars Hellan and Petter Haugereid. 2003. The norsource grammar–an exercise in the matrix grammar building design. In *Proceedings of Workshop on Multilingual Grammar Engineering, ESSLLI 2003*.

Karl M Kapp. 2012. *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons.

Yuji Kawaguchi. 2007. Foundations of center of usage-based linguistic informatics (ubli). *Corpus-Based Perspectives in Linguistics*, pages 3–28.

Yuji Kawaguchi, Toshihiro Takagaki, Nobuo Tomimori, and Yoichiro Tsuruga, editors. 2007. *Corpus-Based Perspectives in Linguistics*, volume 6 of *Usage-Based Linguistic Informatics*. John Benjamins Publishing Company, Amsterdam.

Jong-Bok Kim, Jaehyung Yang, Sanghoun Song, and Francis Bond. 2011. Deep processing of Korean and the development of the Korean Resource Grammar. *Linguistic Research*, 28(3):635–672.

Nate Kornell. 2009. Optimising learning using flashcards: Spacing is more effective than cramming. *Applied Cognitive Psychology*, 23(9):1297–1317.

Hans-Ulrich Krieger and Ulrich Schäfer. 1994. TDL – a type description language for constraint-based grammars. *arXiv preprint cmp-lg/9406018*.

John F. Lalande. 1982. Reducing composition errors: An experiment. *The Modern Language Journal*, 66(2):140–149.

Tuan Anh Le. 2019. *Developing and applying an integrated semantic framework for natural language understanding*. Ph.D. thesis, Nanyang Technological University.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. Automated grammatical error detection for language learners. *Synthesis lectures on human language technologies*, 3(1):1–134.

Icy Lee. 2004. Error correction in L2 secondary writing classrooms: The case of Hong Kong. *Journal of Second Language Writing*, 13(4):285–312.

John SY Lee, Herman Leung, and Keying Li. 2017. Towards universal dependencies for learner Chinese. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 67–71.

Lung-Hao Lee, Li-Ping Chang, and Yuen-Hsien Tseng. 2016a. Developing learner corpus annotation for Chinese grammatical errors. In *2016 International Conference on Asian Language Processing (IALP)*, pages 254–257. IEEE.

Lung-Hao Lee, Gaoqi RAO, Liang-Chih Yu, Endong XUN, Baolin Zhang, and Li-Ping Chang. 2016b. Overview of NLP-TEA 2016 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pages 40–48, Osaka, Japan. The COLING 2016 Organizing Committee.

Lung-Hao Lee, Yuen-Hsien Tseng, and Liping Chang. 2018. Building a TOCFL learner corpus for Chinese grammatical error diagnosis. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Lung-Hao Lee, Liang-Chih Yu, and Li-Ping Chang. 2015. Overview of the NLP-TEA 2015 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pages 1–6, Beijing, China. Association for Computational Linguistics.

Charles N. Li and Sandra A. Thompson. 1981. *Mandarin Chinese: A Functional Reference Grammar*. University of California Press.

Qiang Li. 2013. Coercion of locatives in Mandarin Chinese. In *Workshop on Chinese Lexical Semantics*, pages 76–87. Springer.

Jo-Wang Lin. 2003. Aspectual selection and negation in Mandarin Chinese. *Linguistics*, 41(3):425–459.

Jiang Liping. 2015. The underlying idea and practice of HSK Standard Course. *Journal of International Chinese Teaching*.

Christopher D Manning. 2015. Computational linguistics and deep learning. *Computational Linguistics*, 41(4):701–707.

Montserrat Marimon. 2010. The Spanish Resource Grammar. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, Valletta, Malta. European Language Resources Association (ELRA).

Sarnoff Mednick. 1962. The associative basis of the creative process. *Psychological review*, 69(3):220.

Nurit Melnik. 2007. From 'hand-written' to computationally implemented HPSG theories. *Research on Language and Computation*, 5(2):199–236.

George A Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning sns for automated Japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155.

Luis Morgado da Costa, Francis Bond, and He Xiaoling. 2016. Syntactic well-formedness diagnosis and error-based coaching in computer assisted language learning using machine

translation. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pages 107–116, Osaka, Japan. The COLING 2016 Organizing Committee.

Luis Morgado da Costa and Joanna Ut-Seong Sio. 2020. CALLIG: Computer assisted language learning using improvisation games. In *Proceedings of the Games and Natural Language Processing Workshop at the 12th Edition of the Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association (ELRA).

Luis Morgado da Costa, Roger V P Winder, Shu Yun Li, Benedict Christopher Lin Tzer Liang, Joseph Mackinnon, and Francis Bond. 2020. Automated writing support using deep linguistic parsers. In *Proceedings of the 12th Conference on Language Resources and Evaluation*, Marseille, France. European Language Resources Association (ELRA).

Stefan Müller and Walter Kasper. 2000. HPSG analysis of German. In *Verbmobil: Foundations of speech-to-speech translation*, pages 238–253. Springer.

Noriko Nagata. 1996. Computer vs. workbook instruction in second language acquisition. *CALICO journal*, 14(1):53–75.

Ryo Nagata, Edward Whittaker, and Vera Sheinman. 2011. Creating a manually error-tagged and shallow-parsed learner corpus. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1210–1219, Portland, Oregon, USA. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *CoNLL Shared Task*, pages 1–14.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Diane Nicholls. 2003. The Cambridge learner corpus: Error coding and analysis for lexicography and elt. In *Proceedings of the Corpus Linguistics 2003 conference*, volume 16, pages 572–581.

Hiroki Nomoto, Kenji Okano, David Moeljadi, and Hideo Sawada. 2018. TUFS Asian Language Parallel Corpus (TALPCo). In *Proceedings of the Twenty-Fourth Annual Meeting of the Association for Natural Language Processing*, pages 436–439.

Stephan Oepen. 2001. [incr tsdb()] — competence and performance laboratory. User manual. Technical report, Computational Linguistics, Saarland University, Saarbrücken, Germany. In preparation.

Stephan Oepen, Dan Flickinger, and Francis Bond. 2004. Towards holistic grammar engineering and testing — grafting treebank maintenance into the grammar revision cycle. In *Beyond Shallow Analyses — Formalisms and Statistical Modelling for Deep Analysis (Workshop at IJCNLP-2004)*, Hainan Island.

Stephan Oepen, Kristina Toutanova, Stuart M Shieber, Christopher D Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods treebank: Motivation and preliminary applications. In *COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes*.

Woodley Packard. 2015. Full Forest Treebanking. Master's thesis, University of Washington, USA.

Jeeyoung Peck, Jingxia Lin, and Chaofen Sun. 2013. Aspectual classification of Mandarin Chinese verbs: A perspective of scale structure. *Language and Linguistics*, 14(4):663.

M Perifanou and A Economides. 2014. MOOCs for foreign language learning: an effort to explore and evaluate the first practices. *INTED2014 Proceedings*, pages 3561–3570.

Mateja Petrovčič. 2016. Word sketches of separable words liheci in Chinese. *Acta Linguistica Asiatica*, 6(1):47–57.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press.

Marwa Ragheb and Markus Dickinson. 2012. Defining syntax for learner language annotation. In *Proceedings of COLING 2012: Posters*, pages 965–974, Mumbai, India.

Marwa Ragheb and Markus Dickinson. 2014. Developing a corpus of syntactically-annotated learner language for English. In *Proceedings of the 13th International Workshop on Treebanks and Linguistic Theories (TLT13)*, pages 292–300, Tübingen, Germany.

Gaoqi Rao, Qi Gong, Baolin Zhang, and Endong Xun. 2018. Overview of NLPTEA-2018 share task Chinese grammatical error diagnosis. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–51, Melbourne, Australia. Association for Computational Linguistics.

Gaoqi Rao and Lung-Hao Lee. 2018. NLP for Chinese L2 Writing: Evaluation of Chinese Grammatical Error Diagnosis. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020. Overview of NLPTEA-2020 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 25–35, Suzhou, China. Association for Computational Linguistics.

Claudia Ross and Jing-heng Sheng Ma. 2017. *Modern Mandarin Chinese grammar: A practical guide*. Routledge.

Alla Rozovskaya and Dan Roth. 2010. Annotating ESL errors: Challenges and rewards. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 28–36, Los Angeles, California. Association for Computational Linguistics.

Ivan A Sag, Thomas Wasow, Emily M Bender, and Ivan A Sag. 1999. *Syntactic theory: a formal introduction*, volume 2. CSLI Stanford.

Keisuke Sakaguchi, Courtney Napoles, and Joel Tetreault. 2017. GEC into the future: Where are we going and how do we get there? In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 180–187, Copenhagen, Denmark. Association for Computational Linguistics.

Marwan H Sallam, Elena Martín-Monje, and Yan Li. 2020. Research trends in language MOOC studies: a systematic review of the published literature (2012-2018). *Computer Assisted Language Learning*, pages 1–28.

Edward Sapir. 1921. *Language: An introduction to the study of speech*. New York: Harcourt, Brace & Co.

R Keith Sawyer. 2011a. *Structure and improvisation in creative teaching*. Cambridge University Press.

R Keith Sawyer. 2011b. What makes good teachers great? The artful balance of structure and improvisation. *Structure and improvisation in creative teaching*, pages 1–24.

David Schneider and Kathleen F. McCoy. 1998. Recognizing syntactic errors in the writing of second language learners. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 2*, pages 1198–1204, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mathias Schulze. 2008. AI in CALL – artificially inflated or almost imminent? *Calico Journal*, 25(3):510–527.

Melanie Siegel. 2006. *JACY - A Grammar for Annotating Syntax, Semantics and Pragmatics of Written and Spoken Japanese for NLP Application Purposes*. Habilitation, University of Bielefeld.

Melanie Siegel and Emily M Bender. 2002. Efficient deep processing of Japanese. In *Proceedings of the 3rd workshop on Asian language resources and international standardization-Volume 12*, pages 1–8. Association for Computational Linguistics.

Melanie Siegel, Emily M Bender, and Francis Bond. 2016. *Jacy: An implemented grammar of Japanese*. CSLI Publications.

Ut Seong Joanna Sio and Lian Hee Wee. 2012. Teaching linguistics using improvised comedy. In *Language Arts in Asia: Literature and Drama in English, Putonghua and Cantonese*, pages 283–302. Cambridge Scholars Publishing.

Patrick Suppes, Tie Liang, Elizabeth E Macken, and Daniel P Flickinger. 2014. Positive technological and negative pre-test-score effects in a four-year assessment of low socioeconomic status k-8 student learning in computer-based math and language arts courses. *Computers & Education*, 71:23–32.

Dara Tafazoli, Elena Gómez María, and Cristina A Huertas Abril. 2019. Intelligent language tutoring system: Integrating intelligent computer-assisted language learning into language education. *International Journal of Information and Communication Technology Education (IJICTE)*, 15(3):60–74.

Liling Tan and Francis Bond. 2014. NTU-MC toolkit: Annotating a linguistically diverse corpus. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, page 86−89, Dublin.

Takaaki Tanaka, Francis Bond, Stephan Oepen, and Sanae Fujita. 2005. High precision treebanking —blazing useful trees using POS information. In *Proceedings of the 43rd annual meeting of the Association for Computational Linguistics*, pages 330–337.

Yasuhito Tanaka. 2001. Compilation of a multilingual parallel corpus. In *Proceedings of the 15th Pacific Asia Conference on Language, Information and Computation, PACLIC 2001*, pages 265–268, Hong Kong, China. City University of Hong Kong.

Shou-hsin Teng. 1973. Negation and aspects in Chinese. *Journal of Chinese Linguistics*, pages 14–37.

Kristina Toutanova, Christopher D Manning, Dan Flickinger, and Stephan Oepen. 2005. Stochastic HPSG parse disambiguation using the Redwoods corpus. *Research on Language and Computation*, 3(1):83–105.

Huihsin Tseng, Pi-Chuan Chang, Galen Andrew, Dan Jurafsky, and Christopher D Manning. 2005. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*.

Chih-Hsiung Tu, Laura E Sujo-Montes, and Cherng-Jyh Yen. 2015. Gamification for learning. In *Media Rich Instruction*, pages 203–217. Springer.

Hans Uszkoreit. 2004. New chances for deep linguistic processing. *Computational Linguistics and Beyond. Academia Sinica: Taipei*, pages 111–134.

Catherine Van Beuningen. 2010. Corrective feedback in L2 writing: Theoretical perspectives, empirical insights, and future directions. *International Journal of English Studies*, 10(2):1–27.

Alexandra A Vorobyeva. 2018. Language acquisition through massive open online courses (MOOCs): Opportunities and restrictions in educational university environment. *XLinguae*, 11(2):136–146.

Lulu Wang and Stefan Müller. 2013. Regularity and idiomaticity in Chinese separable verbs. In *Workshop on Chinese Lexical Semantics*, pages 229–240. Springer.

Maolin Wang, Shervin Malmasi, and Mingxuan Huang. 2015a. The Jinan Chinese Learner Corpus. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 118–123.

Shan Wang and Francis Bond. 2013. Building the Chinese Open Wordnet (COW): Starting from core synsets. In *Proceedings of the 11th Workshop on Asian Language Resources, a Workshop at IJCNLP-2013*, pages 10–18, Nagoya.

Wenjie Wang, Sanghoun Song, and Francis Bond. 2015b. A constraint-based analysis of A-NOT-A questions in Mandarin Chinese. In *Proceedings of the 22nd international conference on Head-Driven Phrase Structure Grammar (HPSG 2015)*, pages 196–215.

Roger V. P. Winder, Joe MacKinnon, Shu Yun Li, Benedict Lin, Carmel Heah, Luis Morgado da Costa, Takayuki Kuribayashi, and Francis Bond. 2017. NTUCLE: Developing a corpus of

learner English to provide writing support for engineering students. In *Proceedings of the 4th Workshop on NLP Techniques for Educational Applications (NLPTEA 2017)*, Taipei, Taiwan.

S. Wu, W. Tian, and Y. Zhang. 2010. *Chinese Link: Beginning Chinese. Simplified Character Version, Level 1*. Chinese Link: Zhong Wen Tian Di. Beginning Chinese. Level 1. Prentice Hall.

Richard Z Xiao and Anthony M McEnery. 2008. Negation in Chinese: a corpus-based study. *Journal of Chinese linguistics*, 36(2):274–330.

Renjie Xu, Zhiqiang Gao, Yingji Pan, Yuzhong Qu, and Zhisheng Huang. 2008. An integrated approach for automatic construction of bilingual Chinese-English WordNet. In *Asian Semantic Web Conference*, pages 302–314. Springer.

Nianwen Xue, Zixin Jiang, Xiuhong Zhong, Martha Palmer, Fei Xia, Fu-Dong Chiou, and Meiyu Chang. 2010. Chinese Treebank 7.0. *Linguistic Data Consortium, Philadelphia*.

Liu Xun. 2010. *New Practical Chinese Reader Vol. 1 (2nd.Ed.)*. Beijing Language Culture University Press.

Liang-Chih Yu, Lung-Hao Lee, and Li-Ping Chang. 2014. Overview of grammatical error diagnosis for learning Chinese as a foreign language. In *Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–47.

Li Yuan, Stephen Powell, and JISC CETIS. 2013. MOOCs and open education: Implications for higher education. *Cetis White Paper*.

Yi Zhang. 2008. *Robust deep linguistic processing*. Ph.D. thesis, Saarland University.

Part V:

Appendices

Appendix A

Source Code and Data Repositories

This dissertation produced multiple repositories of source code and language resources that are released to the public, under an open license. Here is a list of all public repositories sharing the work completed during thesis:

1. The source code for all changes committed to ZHONG (including all lexical and syntactic expansions, along with mal-rules and trained parse-ranking model) is released under the MIT License and can be found under: `https://github.com/delph-in/zhong`.

2. The source code for the iTELL suite of applications (including the LCC-APP Assignment Checker and CALLIG) is released under the MIT License and can be found under: `https://github.com/lmorgadodacosta/iTELL`.

3. The source code for the Learner Corpus Tagging Tool (distributed as part of IMI – A Multilingual Semantic Annotation Environment) is released under the MIT License and can be found under: `https://github.com/bond-lab/IMI`

4. The NTU Corpus of Learner English (NTUCLE) is released under a CC-BY 4.0 license and will be made available under: `https://github.com/lmorgadodacosta/NTUCLE`

5. The open sections of the Mandarin Education Corpus (MEC) – i.e., those that are not restricted by copyright – are released under a CC-BY 4.0 license and will be made available under: `https://github.com/lmorgadodacosta/MEC`

Unfortunately, the NTU Corpus of Learner Mandarin (NTUCLM) is not available for release due to licensing restrictions imposed during the collection of the data.

Appendix B

Publications and Presentations

This Appendix includes a list of all peer-reviewed publications, along with conference and invited presentations I have accrued by the end of my PhD degree. I have been very fortunate in my ability to collaborate on many projects during my PhD – many of which were unable to be included in this thesis due to scope.

I am particularly proud of being an active member of the Global Wordnet Association, and of having had the privilege to work on various projects concerned with computational lexical semantics, as well as having helped create three new wordnets during my PhD: The Open Kristang Wordnet, the Coptic Wordnet and the Cantonese Wordnet.

Publications (Peer-Reviewed)

2021 Bond, Francis and Kirkrose Devadason, Andrew and Teo, Rui Lin Melissa and **Morgado Da Costa, Luis**. Teaching Through Tagging —Interactive Lexical Semantics. *Proceedings of the 11th Global WordNet Conference (GWC 2021). Global Wordnet Association*. Pretoria, South Africa.

2021 P. McCrae, John and Wayne Goodman, Michael and Bond, Francis and Rademaker, Alexandre and Rudnicka, Ewa and **Morgado Da Costa, Luis**. The GlobalWordNet Formats: Updates for 2020. *Proceedings of the 11th Global WordNet Conference (GWC 2021). Global Wordnet Association*. Pretoria, South Africa.

2020 **Morgado da Costa, Luis** and Winder, Roger V P and Li, Shu Yun and Liang, Benedict Christopher Lin Tzer and Mackinnon, Joseph and Bond, Francis. Automated Writing Support Using Deep Linguistic Parsers. *Proceedings of the 12th Conference on Language Resources and Evaluation. European Language Resources Association (ELRA)*. Marseille, France.

2020 **Morgado da Costa, Luis** and Sio, Joanna Ut-Seong. CALLIG: Computer Assisted Language Learning using Improvisation Games. *Proceedings of the Games and Natural Language Processing Workshop at the 12th Edition of the Language Resources and Evaluation Conference. European Language Resources Association (ELRA)*. Marseille, France.

2020 **Morgado da Costa, Luis**. Pinchah Kristang: A Dictionary of Kristang. *Proceedings of the Globalex2020 at the 12th Edition of the Language Resources and Evaluation Conference. European Language Resources Association (ELRA)*. Marseille, France.

2020 Bond, Francis and Nomoto, Hiroki and **Morgado da Costa, Luis** and Bond, Arthur. Linking the TUFS Basic Vocabulary to the Open Multilingual Wordnet. *Proceedings of the 12th Conference on Language Resources and Evaluation. European Language Resources Association (ELRA)*. Marseille, France

2020 Bond, Francis and **Morgado da Costa, Luis** and Goodman, Michael Wayne and McCrae, John Philip and Lohk, Ahti. Some Issues with Building a Multilingual Wordnet. *Proceedings of the 12th Conference on Language Resources and Evaluation. European Language Resources Association (ELRA)*. Marseille, France.

2019 Slaughter, Laura and **Morgado Da Costa, Luis** and Miyagawa, So and Büchler, Marco and Zeldes, Amir and Lundhaug, Hugo and Behlmer, Heike. The Making of Coptic Wordnet. Proceedings of the 10th Global WordNet Conference (GWC 2019). Wroclaw, Poland.

2019 Sio, Joanna Ut-Seong and **Morgado Da Costa, Luis**. Building the Cantonese Wordnet. Proceedings of the 10th Global WordNet Conference (GWC 2019). Wroclaw, Poland.

2018 Slaughter, Laura and Wang, Wenjie and **Morgado da Costa, Luis** and Bond, Francis. Enhancing the Collaborative Interlingual Index for Digital Humanities: Cross-linguistic Analysis in the Domain of Theology. Proceedings of the 9th Global WordNet Conference (GWC 2018). Singapore.

2017 Winder, Roger V. P. and MacKinnon, Joe and Li, Shu Yun and Lin, Benedict and Heah, Carmel and **Morgado da Costa, Luis** and Kuribayashi, Takayuki and Bond, Francis. NTU-CLE: Developing a corpus of learner English to provide writing support for engineering students. Proceedings of the 4th Workshop on NLP Techniques for Educational Applications (NLPTEA 2017). Taipei, Taiwan. (IJCNLP 2017 Workshop)

2016 **Morgado da Costa, Luis** and Bond, Francis and Xiaoling, He. Syntactic Well-Formedness Diagnosis and Error-Based Coaching in Computer Assisted Language Learning using Machine Translation. Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016). Osaka, Japan.

2016 **Morgado da Costa, Luis** and Bond, Francis. Wow! What a useful extension! Introducing Non-Referential Concepts to Wordnet. Proceedings of the 10th edition of the International Conference on Language Resources and Evaluation (LREC 2016). Portorož, Slovenia.

2016 **Morgado da Costa, Luis** and Bond, Francis and Kratochvíl, František. Linking and Disambiguating Swadesh Lists: Expanding the Open Multilingual Wordnet Using Open Language Resources. Proceedings of GLOBALEX 2016 Lexicographic Resources for Human Language Technology, 10th edition of the International Conference on Language Resources and Evaluation (LREC 2016). Portorož, Slovenia.

2016 Bond, Francis and Ohkuma, Tomoko and **Morgado Da Costa, Luis** and Miura, Yasuhide and Chen, Rachel and Kuribayashi, Takayuki and Wang, Wenjie. A Multilingual Sentiment Corpus for Chinese, English and Japanese. Proceedings of Emotion and Sentiment Analysis Workshop, 10th edition of the International Conference on Language Resources and Evaluation (LREC 2016). Portorož, Slovenia.

2016 **Morgado da Costa, Luis** and Bond, Francis and Gao, Helena. Mapping and Generating Classifiers using an Open Chinese Ontology. Proceedings of the 8th Global WordNet Conference (GWC 2016). Bucharest, Romania.

2016 Moeljadi, David and Bond, Francis and **Morgado da Costa, Luis**. Basic copula clauses in Indonesian. Proceedings of the Joint 2016 Conference on Head-driven Phrase Structure Grammar and Lexical Functional Grammar, Polish Academy of Sciences, Warsaw, Poland. CSLI Publications. Stanford, CA.

2015 Bond, Francis and **Morgado da Costa, Luis** and Le, Tuan Anh. IMI – A Multilingual Semantic Annotation Environment. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, System Demonstrations (ACL 2015). Beijing, China.

2015 **Morgado da Costa, Luis** and Bond, Francis. OMWEdit - The Integrated Open Multilingual Wordnet Editing System. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, System Demonstrations (ACL 2015). Beijing, China.

Conference or Invited Presentations

2021 Sio, Joanna and **Morgado Da Costa, Luis**. The Cantonese Wordnet. Presented at *The School of Cantonese Studies 2021, The Education University of Hong Kong*. Hong Kong.

2019 **Morgado Da Costa, Luis** and Bond, Francis. Seeking Meaning Beyond Grammaticality: Towards the Future of Intelligent Language Tutoring Systems. Presented at *The 5th CoHASS Graduate Conference, Nanyang Technological University*. Singapore.

2019 Sio, Joanna Ut-Seong and **Morgado Da Costa, Luis**. CALLIG (Computer-Assisted Language Learning using Improvisational Games). *AFLiCo 8 - Language, Cognition, and Creativity. 8th International Biennial Conference of the French Association for Cognitive Linguistics (AFLiCo)*. Mulhouse, France.

2018 Bond, Francis and **Morgado da Costa, Luis**. The Open Multilingual Wordnet and Teaching Through Tagging. Invited Talk at *Workshop on Ontology and Rich Semantics: Frameworks and Application*. Singapore.

2018 **Morgado da Costa, Luis**. Rich Syntactic and Semantic Models in Error Detection and Intelligent Language Tutoring Systems. Invited Talk at *2nd International Conference on Informatics for Development (ICID)*. Yogyakarta, Indonesia.

2018 **Morgado da Costa, Luis**. Multilingual Intelligent Tutoring and Error Detection Systems. Invited Talk at *Chongqing Southwest University*. Chongqing, China.

2018 **Morgado da Costa, Luis**. TELL@NTU Intelligent Tutoring and Error Detection Systems. Presented at the *Technology Enghanced Learning Workshop, 9th Global WordNet Conference (GWC 2018)*. Singapore.

2018 Choi, Hannah and Bond, Francis and **Morgado da Costa, Luis**. Verb Subcategorization in Indonesian. Presented at the *Twenty-Second International Symposium On Malay/Indonesian Linguistics (ISMIL 22)*. Los Angeles, USA.

2018 **Morgado da Costa, Luis**. Basic Digital Humanities: Definitions, Methods and Tools. *Sinophone Borderlands Data Collection and Management Workshop*. Olomouc, Czechia.

2017 **Morgado da Costa, Luis**. Integrating Machine Translation into Intelligent Computer Assisted Language Learning. *FOSSASIA 2017*. Singapore.

2016 **Morgado da Costa, Luis**. Extending Wordnet: the never-ending story. Presented at the *Second Workshop/Hackathon for the Wordnet Bahasa*. Nanyang Technological University, Singapore.

2016 Kratochvíl, František and **Morgado da Costa, Luis**. Extending lexical resources for Abui, a Papuan language. Presented at *The 2nd Workshop/Hackathon for the Wordnet Bahasa*. Nanyang Technological University, Singapore.

2014 **Morgado da Costa, Luis**. Wordnet Extensions. Presented at *The 1st Workshop/Hackathon for the Wordnet Bahasa*. Nanyang Technological University, Singapore.

2014 **Morgado da Costa, Luis**. The OMW Interface. Presented at *The 1st Workshop/Hackathon for the Wordnet Bahasa*. Nanyang Technological University, Singapore.

2014 **Morgado da Costa, Luis**. NTU Multilingual Corpus and Open Multilingual Wordnet (Poster). Presented at the *ADSC /I2R Winter Workshop on Natural Language Processing*. Singapore.

Appendix C

NTUCLE: Error Tag Set

This appendix contains NTUCLE's final error tag set, with an indication of the frequency of each error type contained in the corpus after the adjudication process. A full discussion of how this error tag set was created can be found in Winder et al. (2017). The 'Source' column in the table below indicates how the tags relate in direct comparison with the NUS Corpus of Learner English (NUCLE, Dahlmeier et al., 2013b), which was essential in deriving this tag set:

- 'Sub-divided': broader NUCLE tags that were sub-divided to be more specific
- 'Modified': NUCLE tags that were modified slightly to be more specific
- 'Moved': NUCLE tags that were moved to other categories
- 'NUCLE': NUCLE tags that were not changed
- 'Re-named': NUCLE tags that were re-named to fit the NTUCLE schema
- 'NTUCLE': tags created for NTUCLE

Categories	Tags	Explanation	Freq.	Source
Articles, determiners	ACh	Wrong choice of article/determiner *A development of a new product is required*	69	Expanded
	AMiss	Missing article/determiner *a stall with [a] shorter queue*	449	Expanded
	AUnn	Unnecessary article/determiner *two holes in the two of the sides*	144	Expanded
Citations	CitForm	Incorrect citation form *(Sim, R. 2013)*	100	Expanded
	CitMiss	Missing citation *According to a study [citation], Singaporean students ...*	6	Expanded
Expression	ExpAw	Awkward expression (meaning is clear) *paths are of high human traffic*	366	NTUCLE
	ExpUC	Unclear expression (meaning is unclear) *A rubbish bin to test our idea as well as human resources from the companies*	249	Moved
Mechanics	MCase	Wrong use of upper or lower case *The Rubbish bin is a common object*	98	Expanded
	MPunc	Punctuation error *This[,] in turn[,] would create an orderly environment*	190	Expanded
	MSpace	Missing or unnecessary space *They can not be used in open areas*	27	Expanded
	MSpel	Spelling error *a cold and quite environment*	58	Expanded
Nouns	NCount	Wrong form of countable/uncountable noun *Users can exchange notes and advices*	77	NTUCLE
	NNum	Wrong choice of singular/plural form of the noun *one of his speech*	525	NUCLE
	NPoss	Wrong choice of possessive form *the timers can be adjusted to workers['] feedback*	22	NUCLE
Prepositions	PreCh	Wrong choice of preposition *at the comfort of his home*	227	Expanded
	PreMiss	Missing preposition *EasyGrip will be a great addition [to] every household*	53	Expanded
	PreUnn	Unnecessary preposition *video tutorials can be played to teach users on how to use the mouse*	54	Expanded

Categories	Tags	Explanation	Freq.	Source
Pronouns	ProAgr	Pronoun and reference do not agree in number/person/gender *An electrostatic precipitator works by absorbing dirty air, passing them through ionising electrodes*	88	Re-named
	ProCh	Wrong choice of pronoun *they things tend to slip off their mind easily*	32	Expanded
	ProMiss	Missing pronoun *5 'X' s will identify owners as irresponsible and deny [them] a pet.*	21	Expanded
	ProRef	Unclear reference for pronoun *The components can be mounted onto a circuit board, which is covered with a plastic housing once it is completed.*	92	Modified
	ProUnn	Unnecessary pronoun *Death then follows if the victim he is been left untreated within minutes*	8	Expanded
Sentence structure	SComS	Comma splice *The wobbling table can cause food and drinks to be spilled out of their containers, writing can become messy.*	40	Expanded
	SConv	Convoluted sentence *Rubbish bins are facing one problem in crowded areas where bins fill up quickly that cleaners have hard time discerning as there are too many bins, and only come at fixed timings to clear the rubbish currently.*	-	NTUCLE
	SDMod	Dangling modifier *Looking at the bigger picture, a canteen can efficiently accommodate more diners in a given time.*	16	Expanded
	SFrag	Sentence fragment *Thus, showing that our students have a huge desire to always learn something new.*	58	NUCLE
	SLong	Overly long sentence *However, they would not be able to do the required printing if they possess an EZ-link card that has insufficient stored monetary value and hence may require the assistance of friends by borrowing their EZ-link cards, or make their way back to (...) [+38 words]*	14	NTUCLE
	SMMod	Misplaced modifier *An ideal conducive learning environment is essential as it facilitates effective teaching and learning process coupled with a well-equipped lecture theatre*	11	NTUCLE
	SPar	Parallelism missing *students will find it a hassle to go through emails and calling to find out more*	37	NUCLE

Categories	Tags	Explanation	Freq.	Source
Sentence structure	SRun	Run-on sentence *there is an increase in commuters for public transport[;] this leads to higher congestion in public transport*	26	Expanded
	SSub	Problematic subordinate clause *The immediate benefited [sic] ones would be the needy groups, directly solving their food shortage.*	25	NUCLE
Style	StyContr	Contractions *It's a rectangular device*	25	NTUCLE
	StyF	Overly formal words or expressions *To solve the aforementioned problems*	1	NTUCLE
	StyMood	Inappropriate use of interrogatives and imperatives *Establish a collaboration with an existing music-streaming app.*	13	NTUCLE
	StyPron	Inappropriate use of first and second person pronouns *I could not manage to find the cost of one EZ link top up machine*	9	NTUCLE
	StyWch	Casual or colloquial words or expressions *some find it a hassle to search for an available power socket*	92	NTUCLE
Subject-verb agreement	SubVA	Subject and verb do not agree in number and/or person *The portable charger are basically portable*	148	NUCLE
Transitions	TCh	Wrong choice of link words/phrases *Hence users will also be able to purchase a UV light, where they can use it to identify areas which were not cleaned properly*	50	Expanded
	TMiss	Missing link words/phrases *The food owners select the nearest food centre, [and] fill in their address and contact number.*	26	Expanded
	TUnn	Unnecessary link words/phrases *Skipping lunch can cause students to be distracted by hunger and thus affecting academic performance.*	34	Expanded

Categories	Tags	Explanation	Freq.	Source
Verbs	VForm	Wrong form of the verb *NTU is rank 13th in the world*	231	NUCLE
	VMiss	Missing verb *The files they need [?] directly streamed to their computer.*	23	NUCLE
	VMod	Missing, inappropriate or unnecessary modal *To produce the application, the following steps are taken:*	138	NUCLE
	VTense	Verb tense *Each year Nanyang Technological University (NTU) welcomed approximately 4,500 students into their freshmen year*	121	NUCLE
	VVoice	Wrong choice of active or passive voice *The phenomenon of overcrowding of Canteen B has been existed for a long time.*	27	NTUCLE
Word order	PosAd	Wrong position of adjective/adverb *vacuum cleaners can be used to clean narrow spaces also*	3	Re-named
	PosW	Incorrect word order *the problem of dropping things off the desk*	13	Re-named
Words (lexical)	WCh	Wrong choice of word *The air conditioner is an electric appliance that alternates the surrounding temperature.*	411	NTUCLE
	WColloc	Words do not collocate *Find assistance from Sistic to sell tickets*	73	NTUCLE
	WForm	Wrong form of the word *Rentascoot™ is environmental friendly*	96	NUCLE
	WMiss	Missing words *This system can simplify [?] and reduce the time of packing away [?].*	95	NTUCLE
	WUnn	Unnecessary words *... which poses severe risks to nature as well as human health issues*	195	NTUCLE
Others	Oth	Other errors requiring correction	140	NUCLE

Table C.1: Final list of error tags. Examples for each error are provided below the explanation of each tag, with the words selected for each error underlined. Possible corrections are provided in brackets when deemed necessary.