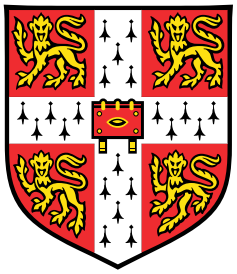
Automatic syntactic analysis of learner English



Yan Huang

Theoretical and Applied Linguistics
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Clare Hall November 2018

I would like to dedicate this thesis to my loving parents and brother.

Acknowledgements

I would like to express my gratitude to my supervisor, Anna Korhonen, for her valuable guidance throughout this project. I am indebted to her for suggesting this avenue of research and providing helpful feedback. I thank her for introducing me to collaborators from different fields and institutions, and I have benefited immensely from her comments on writing. Above all, I am grateful for her faith in me, and for being positive and encouraging during my research exploration.

I also thank Dora Alexopoulou for her advice and practical support. Dora suggested reading materials which inspired my work. She also provided valuable feedback with regard to second language acquisition in the thesis. Furthermore, Dora invited me to participate in the EF project, which enabled me to gain a deep understanding of learner language. It has been a great pleasure to work with Dora and I am grateful for her kindness.

Special thanks are given to my close collaborators Yevgeni Berzak, Akira Murakami and Simon Baker, for their inspirational ideas, constructive comments, and technical suggestions. I am grateful to Roi Reichart and Nigel Collier for their insightful comments on my work. I would also like to thank my colleagues and friends in the Language Technology Lab and the Computer Lab for their technical support, interesting discussion and exchange of ideas.

My study has been funded by the Grace and Thomas C.H. Chan Cambridge International Scholarship from the Cambridge Trust. I am grateful to Mr. and Mrs. Chan and the Cambridge Trust for the generous financial support.

On a personal note, I wish to thank my many friends in Cambridge, who have brought me laughter and made me feel at home in the foreign land. Specifically, I would like to thank Meyya Nagappan, Wesam Asali, Carla Pastorino Campos, Caterina Pello, Milan Gritta and Samantha Hajna for their great inspiration and personal support. Also, I would like to thank Cherry Lam, Daniela Gerz, Gamal Crichton, Qianchu Liu, Meng Zhang, Menglin Xia, Yimai Fang, Laura Wittemans, Xingyi Cheong, Yaqiao Deng and Shi Pu for their good company. I am grateful to Sheng Liu, my friend back in China, for always seeing the best in me. The love and support of my family are constants that I could not have done without. I thank my parents and brother for their continuous encouragement. My final but greatest thanks go to Duncan Grant, my partner, who has brought beauty, humor and peace into my PhD life.

Abstract

Automatic syntactic analysis is essential for extracting useful information from large-scale learner data for linguistic research and natural language processing (NLP). Currently, researchers use standard POS taggers and parsers developed on native language to analyze learner language. Investigation of how such systems perform on learner data is needed to develop strategies for minimizing the cross-domain effects. Furthermore, POS taggers and parsers are developed for generic NLP purposes and may not be useful for identifying specific syntactic constructs such as subcategorization frames (SCFs). SCFs have attracted much research attention as they provide unique insight into the interplay between lexical and structural information. An automatic SCF identification system adapted for learner language is needed to facilitate research on L2 SCFs.

In this thesis, we first provide a comprehensive evaluation of standard POS taggers and parsers on learner and native English. We show that the common practice of constructing a gold standard by manually correcting the output of a system can introduce bias to the evaluation, and we suggest a method to control for the bias. We also quantitatively evaluate the impact of fine-grained learner errors on POS tagging and parsing, identifying the most influential learner errors. Furthermore, we show that the performance of probabilistic POS taggers and parsers on native English can predict their performance on learner English.

Secondly, we develop an SCF identification system for learner English. We train a machine learning model on both native and learner English data. The system can label individual verb occurrences in learner data for a set of 49 distinct SCFs. Our evaluation shows that the system reaches an accuracy of 84% F1 score. We then demonstrate that the level of accuracy is adequate for linguistic research. We design the first multidimensional SCF diversity metrics and investigate how SCF diversity changes with L2 proficiency on a large learner corpus. Our results show that as L2 proficiency develops, learners tend to use more diverse SCF types with greater taxonomic distance; more advanced learners also use different SCF types more evenly and locate the verb tokens of the same SCF type further away from each other. Furthermore, we demonstrate that the proposed SCF diversity metrics contribute a unique perspective to the prediction of L2 proficiency beyond existing syntactic complexity metrics.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

ANLT Alvey Natural Language Tools

BNC British National Corpus

CAF Complexity, Accuracy, and Fluency

CEFR Common European Framework of Reference for Languages

CG Construction Grammar

CLC-FCE Cambridge Learner Corpus - First Certificate in English

COCA Corpus of Contemporary American English

EFCAMDAT EF-Cambridge Open Language Database

IL Interlanguage

L1 First Language

L2 Second Language

LAS labeled Attachment Score

LE Learner Error

MATTR Moving-Average Type-Token Ratio

MaxEnt Maximum Entropy

MLC Mean Length of Clauses

MPB Multiple-Parser-Based

NLP Natural Language Processing

PCFG Probabilistic Context-Free Grammar

PE Parsing Error

POS Parts of Speech

PTB Penn Treebank

SCA Syntactic Complexity Analyzer

SCF Subcategorization Frame

SD Stanford Typed Dependencies

SPB Single-Parser-Based

TAASSC Tool for the Automatic Analysis of Syntactic Sophistication and Complexity

TL Target Language

UAS Unlabeled Attachment Score

UD Universal Dependencies

VAC Verb Argument Construction

WSJ Wall Street Journal

Chapter 1

Introduction

Automatic syntactic analysis is essential for extracting useful information from large-scale learner corpora for linguistic research and natural language processing (NLP). While researchers mainly use standard syntactic analysis systems developed on native language data to analyze learner data, research that investigates the cross-domain effect on the performance of the systems has been limited. Furthermore, there is a lack of systems for automatic identification of subcategorization frames, syntactic constructs important to second language (L2) research. In this study, we provide a comprehensive evaluation of standard POS taggers and dependency parsers on learner data. We also develop an SCF identification system for learner data and propose novel SCF-diversity metrics, which prove to be useful for profiling L2 development.

This introductory chapter first identifies the need for automatic syntactic analysis of large-scale corpora. We then discuss automatic syntactic analysis on the specific domain of learner language, identifying two major problems in the area (section 1.2). Section 1.3 summarizes our contribution to research in this area. The list of external resources used in our research is given in section 1.4. Section 1.5 includes an overview of the organization of this thesis.

1.1 Automatic syntactic analysis

Recent decades have seen the emergence of increasingly large collections of machine-readable texts of natural language, i.e., corpora. Such corpora provide exciting opportunities for both linguistic research and applications of natural language processing (NLP). For linguistic research, large datasets can help to improve the empirical basis of conclusions, and can support the discovery of linguistic phenomena that have escaped human intuition

before (Cai and Liu, 2017). For NLP, a large amount of data can facilitate the use of machine learning models to build powerful and accurate applications.

To fully exploit the power of corpora, it is essential to analyze corpora linguistically. Particularly, syntactic analysis is important. Syntax describes how words are combined into sentences. Such structural information has close interrelation with the other aspects of language including morphology, lexicon, semantics, and pragmatics. Syntax has been an important subject for investigation in corpus linguistics and theoretical linguistics (Gilquin and Gries, 2009). Syntactic analysis provides not only indices to syntactic phenomena in corpora, but also clues for retrieving lexical, semantic and pragmatic information (Meurers, 2015). Syntactic analysis has also been widely used to develop NLP techniques such as lemmatization (Bird and Loper, 2004), semantic analysis (Roth and Lapata, 2016) and discourse analysis (Wang et al., 2017), as well as NLP applications such as machine translation (Meng et al., 2015), automatic summarization (Cheung and Penn, 2014), sentiment analysis (Choi and Cardie, 2008), and question answering (Andreas et al., 2016). Even though recent development of deep learning models has enabled researchers to build some state-of-the-art NLP applications without using explicit syntactic information, such NLP applications are restricted in the area where large-scale annotations of target information are available. Syntactic information remains vital to NLP applications where annotated data is limited.

As corpora become larger, manual marking of syntactic information is infeasible. As a result, automatic techniques are required. In the present thesis, we use "annotation" to refer to the manual marking of linguistic information, and we use "analysis" to refer to the automatic marking of linguistic information. Existing syntactic analysis systems mainly focus on parts-of-speech (POS) tagging and syntactic parsing. Early POS taggers and syntactic parsers were developed by hand-crafted rules, while recent systems resort to increasingly powerful probabilistic models trained on syntactic annotation, which has led to increased accuracy.

However, two problems exist in the automatic syntactic analysis. First, state-of-the-art POS taggers and parsers are mostly trained on a particular domain of native language, e.g., the newswire domain for English. Such POS taggers and parsers might not generalize well to other domains, where their accuracy can drop extensively. Second, POS taggers and parsers are developed for generic NLP purposes. They may not be useful for extracting some specific syntactic phenomena of interest to linguistic research.

1.2 Syntactic analysis of learner language

Learner language represents a domain which is quite different from native language. Learner language includes un-canonical syntactic structures and shows more variation than native language due to factors such as first language (L1) background and L2 proficiency.

The syntax of learner language has long been an important factor in L2 research, education, and NLP applications. L2 researchers have used syntactic information in learner corpora to investigate the acquisition of syntactic features and structures (Crosthwaite, 2016; Kyle, 2016; Lu, 2010; Murakami and Alexopoulou, 2015; Römer et al., 2014; Vyatkina, 2013). Learner corpora also allow L2 educators to extract syntactic observations for identifying typical stages and common problems in L2 teaching (Díaz-Negrillo et al., 2010). In NLP, the syntactic analysis of learner language has been used to support L2 educational applications such as automatic error correction (Ng et al., 2014), automatic essay scoring (Tetreault et al., 2010) and intelligent language tutoring systems (Meurers, 2012). The syntactic analysis of learner corpora has also been used to support the development of NLP applications for non-educational purposes such as automatic native language identification techniques (Jiang et al., 2018; Tetreault et al., 2013), which can be useful in author profiling and forensic analysis (Perkins, 2015).

The most commonly used syntactic analysis systems in L2 research are POS taggers (Gries and Berez, 2017), followed by syntactic parsers for dependency structure or constituency structure (e.g., Kyle, 2016; Lu, 2010). Based on the results of POS taggers or syntactic parsers, researchers have developed techniques for analyzing abstract syntactic features, such as syntactic structural similarity (Graesser et al., 2011) and syntactic complexity (Biber, 1988; Kyle, 2016; Lu, 2010). These techniques are also used to analyze learner language.

Automatic syntactic analysis of learner language is also challenged by the cross-domain problem. The systems that have been used to analyze learner data were developed based on native language data (hereafter referred to as standard systems). However, learner language is significantly different from native language. Standard systems may not perform well on learner data due to learner errors and un-canonical structures. A small number of studies have evaluated the accuracy of POS taggers (Geertzen et al., 2013; Rehbein et al., 2012; Van Rooy and Schäfer, 2002) and parsers (Geertzen et al., 2013; Krivanek and Meurers, 2011; Ott and Ziai, 2010) on learner data. However, most evaluations employed gold-standard annotations which were obtained by manually correcting the results of a POS tagger or parser. Such annotations might have biased towards the reference POS taggers or parsers, making the evaluation results inaccurate (See section 2.3.2). Furthermore, there has been no systematic investigation of how fine-grained learner errors influence standard syntactic analysis systems. Such information would be important for L2 researchers to get an in-depth understanding of

how standard systems perform on learner data, and what preprocessing techniques should be used to minimize the unwanted impact of learner errors on syntactic analysis. Also, there has been no systematic comparison of the performance of multiple syntactic analysis systems on learner English. Neither has there been any comparison of how standard systems perform on native data as opposed to learner data. Such information would be important for L2 researchers to know which syntactic analysis systems, among the many that are available, should be chosen for analyzing learner data. For example, if the accuracy of parsers on native data and learner data is correlated, L2 researchers can predict the accuracy of a parser on learner data based on its accuracy on native data.

Another problem has been the lack of systems for identifying more specific syntactic constructs necessary for L2 research. One such syntactic construct is subcategorization frame (SCF). SCF denotes the number and types of syntactic complements required by a predicate (Chomsky, 1965). SCF requires distinction between complements and adjuncts. For example, the prepositional object *on the chair* in sentence (a) is a complement of *put*, whereas the other prepositional object *in a hurry* is an adjunct. Recent decades have seen an increasing interest in investigating SCF phenomena in L2 research (Bley-Vroman and Joo, 2001; Ellis and Ferreira–Junior, 2009; Gries and Wulff, 2005; Juffs, 1998; Kim et al., 2017; McDonough and Trofimovich, 2016; Römer et al., 2015, 2014; Tono, 2004; White, 1987). SCFs link lexis and morphosyntax, providing unique insight to the interplay between lexical and structural information in L2 acquisition, the ways in which learners might generalize syntactic patterns from individual lexical realizations, and how the properties of individual verbs might constrain the acquisition of morphosyntax.

(a) Sam **put** the pen [on the chair] (in a hurry).

Some L2 studies on SCFs have used corpora to extract empirical evidence. However, they relied on manual post-edition of the output of a POS tagger (Ellis and Ferreira–Junior, 2009; Römer et al., 2014) or syntactic parser (Meurers et al., 2013; Römer et al., 2015; Tono, 2004). Manual post-edition is costly and time-consuming (Römer et al., 2015), and has restricted the amount of SCF data available for research. As a result, previous studies investigated only a limited number of SCF types. The number of learners, as well as their range of L2 proficiency levels and L1 background, is also limited. Developing a system for automatic identification of SCFs can help researchers to save effort in post-edition and achieve SCF information at an unprecedented scale.

Automatic SCF identification is also important for developing linguistic complexity metrics. Measures that gauge linguistic complexity, accuracy, and fluency (CAF) in learner production are fundamental to L2 research (Norris and Ortega, 2009). Research within CAF

has identified a variety of measures, such as lexical diversity, the mean length of clause, the frequency of subordination and coordination (Housen et al., 2012). No research to date has investigated SCFs from the perspective of linguistic complexity. Kyle (2016) points out that independent indices are used to measure lexical and syntactic complexity even though they are actually interrelated, and SCF-based complexity metrics may be useful in measuring this interrelation. However, a large amount of SCF annotation is a pre-condition for calculating SCF-based syntactic complexity measures. Kyle (2016) developed a tool which processes the output of a dependency parser and extracts a syntactic construct called verb argument construction (VAC) to calculate frequency-based linguistic complexity. However, while VAC was intended to be SCF in theory, it was operationally defined as including all direct dependents of a verb. In other words, VAC does not distinguish between complements and adjuncts.

1.3 Our contribution

The aim of the present thesis is to provide some solutions to the cross-domain problem in automatic syntactic analysis and the lack of an SCF identification system for L2 research. We focus on learner English, as English is the most widely learned and used L2 in the world (Lewis et al., 2009). Our study consists of three parts: the evaluation of standard POS taggers and dependency parsers on learner English, the development of an SCF identification system for learner English, and the investigation of SCF diversity during L2 development[1]

1.3.1 Evaluating standard POS taggers and parsers on learner English

The first part of our study evaluates the performance of multiple probabilistic POS taggers and parsers on learner English. We first evaluate the accuracy of the POS taggers and parsers. During this evaluation, we control for annotation bias in the gold standard, and investigate the extent to which annotation bias may affect the evaluation. We also evaluate the impact of fine-grained learner errors on the performance of a POS tagger and a parser, identifying the most influential learner errors. Furthermore, we compare the performance of the POS taggers and parsers on learner English with their performance on native English, and investigate whether the performance of the POS taggers and parsers on native English can predict their performance on learner English.

[1] We release the SCF identification system and a gold standard annotation of POS tags, dependencies, SCFs and learner errors for learner English at https://github.com/cambridgeltl/subcategorization-frames-and-learner-English-data. Some native English SCF recourses are also available in the repository.

Our results have rich implications with regard to syntactic annotation and analysis of learner data. Firstly, we demonstrate that the common practice of constructing a gold standard – by manually correcting the output of a single parser – can introduce bias to the evaluation of the POS taggers and parsers. We propose an alternative annotation method which can control for the annotation bias. Secondly, We find that learner errors do have an impact on POS tagging and parsing output, and that learner errors on punctuation, spelling, capitalization, argument structures, determiners and prepositions cause most POS tagging and parsing errors. Correcting these learner errors will be an effective pre-processing technique to reduce POS tagging and parsing errors for downstream linguistic research and NLP applications on learner English. Thirdly, we demonstrate that the performance of probabilistic parsers on learner English can be predicted by their performance on native English. This implies that when we want to choose a probabilistic parser for learner English, the most accurate parser on native English can be a good candidate.

1.3.2 Automatic SCF identification for learner English

As a second part of the study, we develop the first SCF identification system for learner English. We adopt a supervised classification approach, training a machine learning model to classify the SCFs of individual verb tokens according to lexical, syntactic and semantic information of the verbs and the context. We solve the cross-domain problem by training the classifier on learner data as well as native language data, both of which are annotated with SCFs. The resulting system can label individual verb occurrences in learner corpora for a set of 49 distinct SCFs ranging from basic transitive and intransitive frames to complicated frames that involve prepositional, verbal or clausal complements. We evaluate the accuracy of the SCF identification system on learner data, and conduct an error analysis. Results show that the system reaches an accuracy of 84.2% in general.

1.3.3 Application of the SCF identification system to linguistic research

To illustrate the usefulness of the SCF identification system, we design the first metrics for SCF diversity. Our metrics are multidimensional, following the suggestion of Jarvis (2013) for a more varied approach to complexity metrics. We then investigate how SCF diversity changes along the full span of L2 proficiency on a large learner corpus. We also compare the prediction power of L2 proficiency between SCF diversity and existing syntactic complexity metrics. Our results shed light on the L2 development of SCF diversity. We quantitatively demonstrated that as L2 learners become more proficient, they tend to use more diverse

SCF types with larger taxonomic distance. Also, more advanced learners use different SCF types more evenly, and locate the verb tokens of the same SCF type further away from each other. Meanwhile, Our results show that the proposed SCF diversity metrics can be a useful measure of linguistic complexity, contributing to the prediction of L2 proficiency beyond existing syntactic complexity metrics.

1.4 External resources

Corpora For learner English data, we employ EF-Cambridge Open Language Database (EFCAMDAT) (Geertzen et al., 2013). For native English data, we employ Penn Treebank (Marcus et al., 1993), and an SCF dataset (Quochi et al., 2014) developed by annotating native language data sampled from British National Corpus (Aston and Burnard, 1998).

Software For parser evaluation, we employ Stanford unlexicalized (Klein and Manning, 2003a) and lexicalized (Klein and Manning, 2003b) parsers, BLLIP parser (Charniak and Johnson, 2005), Berkeley parser (Petrov and Klein, 2007), Turbo parser (Martins et al., 2013) and MaltParser (Nivre et al., 2007). We also employ the Stanford tool for converting constituency structure to dependency structure (De Marneffe and Manning, 2008a). We develop the SCF identification system with Python and SyntaxNet parser (Andor et al., 2016).

1.5 Overview of subsequent chapters

The remaining chapters of this thesis are organized as follows:

Chapter 2 (*Background to syntactic analysis of learner language*) introduces the background and motivation for our work. We review the theoretical concepts and existing taxonomies of POS, syntactic structure, and subcategorization. We then review L2 research on SCFs and syntactic complexity to establish why evaluation of standard POS taggers and parsers on learner language and automatic identification of SCFs are needed. We then survey previous research on evaluating POS taggers and dependency parsers. We also review existing NLP systems related to SCFs. Finally, we summarize the issues that this study intends to solve.

Chapter 3 (*Data*) introduces the datasets used in the experiments of this study. These datasets include learner English data from the EFCAMDAT, native English data from Penn Treebank, and a subset of BNC annotated with SCFs.

Chapter 4 (*Evaluation of POS taggers and parsers*) presents our evaluation of the performance of standard POS taggers and dependency parsers on learner English. We introduce the design of three experiments and then report the result of each experiment. The first experiment involves two rounds of annotation for POS tags and dependency structures on learner data. We compare the accuracy of multiple parsers on the two annotations, confirming the existence of annotation bias in the first annotation. We then analyze and discuss various reasons for the annotation mistakes. The second experiment involves annotation of the impact of learner errors on the output of a parser. We find that learner errors have a substantial impact on parsing output, and we summarize the most influential learner errors. The third experiment compares the accuracy between standard parsers on native language data as opposed to learner data. We find that the performance of a parser on native language data is predictive of its performance on learner data.

Chapter 5 (*Automatic SCF identification*) presents an automatic SCF identification system for learner language. We describe the annotation of SCFs on the learner data, and introduce the machine learning model that we use to develop the system. We train the system in two settings, one using learner SCF data only and the other with additional native SCF data. The second setting proves to be more accurate. We then report the fine-grained accuracy of each SCF type of the system. We also present an error analysis of the system.

Chapter 6 (*Application of automatic SCF identification: investigating L2 SCF diversity*) demonstrates the usefulness of our SCF identification system. First, we design multidimensional SCF diversity metrics. We then use the SCF identification system to analyze learner English data, based on which we extract the SCF diversity measures of the data. We analyze the relation between SCF diversity and L2 proficiency, and find that more advanced learners tend to use more diverse SCFs. Furthermore, we compare the SCF diversity metrics with existing syntactic complexity metrics, and find that the SCF diversity metrics can contribute a unique perspective to the prediction of L2 proficiency.

Chapter 7 (*conclusions*) summarizes the contributions of this study and points out future directions for research on syntactic analysis of learner data.

Chapter 2

Background to syntactic analysis of learner language

In this chapter, we first review the theoretical definitions of relevant syntactic constructs and introduce the existing taxonomies of these constructs. The purpose is to establish the basic syntactic concepts that are used throughout the thesis. During the theoretical review, we discuss the difficulty of defining the syntactic constructs, which reveal inherent ambiguity in the boundary areas of these concepts. This theoretical discussion provides an explanation for the ambiguous cases in the annotation practice in Chapter 4. We then review studies on the L2 acquisition of SCFs and syntactic complexity. The purpose is to establish the importance of automatic syntactic analysis of learner language in L2 research, and to point out the need for automatic SCF identification techniques in the research of L2 SCFs and syntactic complexity. Finally, we review syntactic analysis of learner language. We first discuss the major challenges in analyzing the syntax of learner language. We then survey existing POS taggers and parsers, and review previous evaluations of how standard POS taggers and parsers developed on native language perform on learner language. We then review existing NLP systems regarding SCFs.

2.1 Theoretical review of syntax

Syntax describes the way in which words of a language may be strung together to form sentences (Culicover, 1982). The basic building blocks of all syntactic theories are the syntactic categories of words, or parts of speech (POS). POS makes it possible to describe syntax at an abstract level without enumerating all the word combinations for the same syntactic pattern. The syntactic structure of a sentence is described with structural relations

between the POS of the words in the sentence. Subcategorization frame (SCF) describes the syntactic structure conventionally used with a lexical item, connecting syntax with the lexicon. This paper focuses on the analysis of the three syntactic constructs mentioned above, i.e., POS, syntactic structure, and SCF. The following sections review the theoretical concepts of these constructs and their taxonomies in computational linguistics.

2.1.1 Parts of speech

A syntactic category is a group of words that have similar grammatical properties in a given language (Schachter and Shopen, 1985). Traditional grammar uses POS to refer to the syntactic categories of words, and the POS system developed by traditional grammar has largely fed into modern linguistic theories. For example, the first extant grammar of Greek classified POS into eight categories, i.e., nouns, verbs, participles, article, pronouns, prepositions, adverbs, and conjunctions (Davidson, 1874), which are more or less followed by modern linguistic theories.

In fact, POS is so fundamental to syntax that many modern linguistic theories directly build on commonly-used POS without rigorously defining them at the beginning (Langacker, 1987). Traditionally, POS is defined in terms of the semantic classes of words. For example, a noun denotes "the name of a person, place or thing"; an adjective denotes "a modifying property"; and a verb denotes "an event, action, process or state" (Tallerman, 2013). Such definitions can help to identify the central members of a syntactic category, but fail to provide an adequate basis for classifying many boundary cases. For example, *running* expresses an action, but is used as a noun in *during the running* and an adjective in *the running man*.

Contrastingly, Schachter and Shopen (1985) propose that the primary criteria for POS classification are grammatical rather than semantic. More specifically, the grammatical criteria can be derived from three aspects: the distribution of the word in sentence structure, the inflectional features of the word, and the grammatical relations that the word can perform in a sentence (e.g., subject, object, and indirect object). Such grammatical criteria provide a more workable basis for classifying POS. For example, *running* in *during the running* can be classified as a noun according to the distributional criterion that the determiner *the* usually co-occurs with a noun to express the reference of the noun in context. However, such criteria cannot identify whether a category is a POS – which is normally reserved for "major classes" – or a subclass within a POS class. The grammatical criteria alone can only serve the purpose of distinguishing words apart and provide no principles as to what criteria are central to a category or what categories are actually substantive rather than "a matter of terminology" (Schachter and Shopen, 1985).

Table 2.1 Prototypical syntactic categories

	Noun	Adjective	Verb
Semantic class	Object	Property	Action
Pragmatic function	Reference	Modification	Predication

Croft (1991) argues that grammatical criteria are internal criteria for classifying POS, and more substantive POS categories can be established by considering external criteria in lexical semantics and pragmatics. He proposes that syntactic categories can be considered as typological variation centered around different prototypes, and that three prototypes – nouns, verbs, and adjectives – can be established by pairing typical lexical semantic class and pragmatic function for each category as in Table 2.1, for which the definitions of pragmatic functions are as follows:

Reference to get the hearer to identify an entity as what the speaker is talking about;

Predication to say something about the referent;

Modification to fix the identity of the referent or provide a secondary comment on the predication.

Moreover, the existence of these prototypical syntactic categories can be proved by a typological pattern of markedness (Greenberg, 2005) in terms of lexical roots (e.g., an unmarked lexical root has no more morphemes than a marked one), inflection (e.g., an unmarked lexical item has no less inflectional options than a marked one) and textual features (e.g., an unmarked lexical root appear in higher frequency than a marked one). The prototypes are unmarked, whereas intermediate categories are marked. The prototypical syntactic categories may be universal, as Croft (1991) finds consistent typological markedness patterns for the three syntactic categories across twelve languages which are diverse in terms of language families and regions. Table 2.2 illustrates the typological pattern of markedness in terms of morphosyntax for English. When a lexical root is a member of prototypical syntactic categories, the surface realization of the lexical root has no more monosynaptic features than the marked ones. For example, *run*, a prototypical verb that semantically denotes an action and is pragmatically used to talk about how a referent moves, is unmarked; contrastingly, *running*, which still semantically denotes an action but pragmatically refers to that action, requires an additional gerundial morpheme.

The consistency in the typological markedness patterns across different languages supports the substantive status of the POS categories of nouns, verbs, and adjectives. In other words, these major syntactic categories exist not just because they are grammatically different

Table 2.2 Typological markedness pattern in morphosyntax for English

	Reference	Modification	Predication
Objects	unmarked nouns	genitive, adjectivalizations, PPś on nouns	predicate nominals
Properties	deadjectival nouns	unmarked adjectives	predicate adjectives
Actions	action nominals, complements, infinitives, gerunds	participles, relative clauses	unmarked verbs

from each other within a language, but also because they have a distinct basis in human cognition and communication.

Further analysis of fine-grained semantic features and diachronic factors can distinguish other commonly-used POS, which are either the subclasses of or intermediate classes between the major syntactic categories. For example, one notable semantic feature that distinguishes adjectives from nouns is gradability: prototypical adjectives are gradable as they can be manifested in degrees (e.g., *very happy* vs. *mildly happy*), while prototypical nouns are not gradable (Croft, 1991). Cognitively, there is also a spectrum of conceptualizations ranging from describing (denoting a property, which is strictly adjectival), classifying (denoting an object that has the properties of a stereotypical kind such as *thieves* in *they are thieves*, which is nominal but also has some characteristics of adjectives) to individualizing (denoting an individual object, which is strictly nominal)(Bolinger, 1980). Croft (1991) finds markedness pattern showing that pronouns are a subclass of nouns which are more prototypical nouns than common nouns: more varied inflectional options (e.g., gender and case difference in English)and higher textual frequency. He offers an explanation from the semantic and cognitive aspects: pronouns are always cognitively used to individualize, i.e., refer to an object, rather than to classify, whereas common nouns are often used to classify.

Meanwhile, Croft (1991) shows that numerals and quantifiers have markedness pattern intermediate between those of prototypical nouns and prototypical adjectives. The can be explained from the semantic and cognitive perspectives: quantities fall on a gradable scale like adjectives, but are more discrete so that they can also refer to units as an object. Whether quantities are more like nouns or adjectives depends on the scale of the quantities: when the units are smaller, the individual unit is cognitively more salient and the quantities are more likely to be conceived of as describing the property of the individual(s) (e.g., *three people*), otherwise the aggregate of the units is cognitively more salient and can be considered as a whole (e.g., thousands of people). Similarly, mass nouns, which are sometimes used as modifiers, are intermediate between prototypical nouns and prototypical adjectives. This is

because mass nouns are similar to properties in terms of being unbounded, homogeneous and not individuatable.

Diachronic changes have also contributed to the intermediate syntactic categories. For example, auxiliaries are changing from full verbs to verb affixes. Moreover, adpositions are changing from verbs and relational nouns to case affixes (Heine and Reh, 1984). These theories on the causes of intermediate syntactic categories can help to explain why some syntactic categories are more prone to annotation bias, as we will see in Chapter 4.

While Croft (1991) provides a promising framework for determining syntactic categories, there is on-going discussion and disagreement on this issues among theoretical linguists and cognition scientists (Evans and Levinson, 2009). Corpus linguists and computational linguists, nevertheless, have come up with extensive POS tagsets for practical purposes. The first POS tagset, developed for the Brown corpus of English, contains 87 POS tags (Francis, 1964). Subsequent English POS tagsets expand further, e.g., to 197 tags for the London-Lund corpus of spoken English (Svartvik, 1990). The goal of the extensiveness was to provide distinct coding for all classes of words that have distinct grammatical behavior (Garside et al., 1988). The English POS tagset developed for the Penn Treebank (PTB) (Marcus et al., 1993), however, reduces to 36 tags for the purpose of avoiding data sparsity issue – some infrequent POS categories may involve very few cases in the corpus, which can cause problems to statistical machine learning models for automatic POS tagging. The POS tags eliminated during the reduction are mostly lexically or syntactically recoverable. For example, the Brown corpus tagset assigns distinct POS tags to auxiliaries like *have* and *do*, whereas the PTB tagset merges the auxiliaries with verbs; these auxiliaries can be easily recovered by their surface forms. Ever since its release, PTB has become an influential benchmark dataset for automatic POS tagging and syntactic parsing of English. As a result, the POS tagset of PTB is widely used in both computational linguistics and corpus linguistics.

There are also POS tagsets for languages other than English (e.g., Ejerhed et al., 1992; Hardie, 2003; Khoja, 2001; Przepiórkowski and Woliński, 2003; Xia, 2000). However, designing POS tagsets and annotating POS resources are costly. The lack of POS annotation resources for some minor languages has driven some NLP researchers to investigate cross-lingual POS projection from resource-rich languages like English to resource-poor languages (Das and Petrov, 2011; Xi and Hwa, 2005; Yarowsky and Ngai, 2001). Alternatively, some researchers developed unsupervised techniques for inducing POS tags from multiple languages (Naseem et al., 2009; Snyder et al., 2009). To tackle the increasing need of comparative evaluation during cross-lingual and multilingual natural language processing, Petrov et al. (2012) proposes a universal POS tagset of twelve POS tags, providing a mapping between the universal tagset and language-specific POS tagsets for 25 languages. The

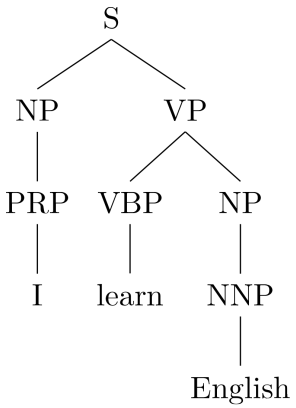


Fig. 2.1 Constituency structure

Universal Dependencies (UD) project (Nivre et al., 2016) further expands the tagset to 17 tags. While these tags are intended to describe the similarity between languages, UD also provides a tagset of morphological features (e.g., Number, which can take values like singular and plural, etc.) to capture detailed and possibly language-specific lexical information.

2.1.2 Syntactic structure

There are two major approaches to describing syntactic structure: the constituency-based approach and the dependency-based approach. While the basic ideas of both approaches can be found in early linguistic theories (e.g., Jespersen, 1924), the concrete ideas of the constituency-based approach were first developed by Chomsky (1957). Subsequently, the constituency approach has underlined many modern syntactic theories including Transformational Grammar (Chomsky, 1965), Lexical Functional Grammar (Kaplan et al., 1982), Generalized Phrase Structure Grammar (Gazdar, 1985) and Head-driven Phrase Structure Grammar (Pollard and Sag, 1994). The constituency approach basically combines words into constituents and recursively combine smaller constituents into larger ones, forming a hierarchical sentential structure. For example, a clausal constituent consists of a subject (noun phrase) and a predicate (verb phrase). Figure 2.1 illustrates the constituency structure.

The dependency approach, on the other hand, was developed by Tesnière (1965) who insisted that verbs should be the root of clauses. The dependency approach establishes pairwise functional relations between words: each relation defines the dependence of a word over the other, i.e., the head. Figure 2.2 illustrates the dependency structure.

Since the dependency-based approach does not involve non-terminal nodes, i.e., nodes beyond the word level, in the tree structure (compare Figure 2.1 and 2.2), dependency representation is minimal compared to constituency representation. Nevertheless, dependency

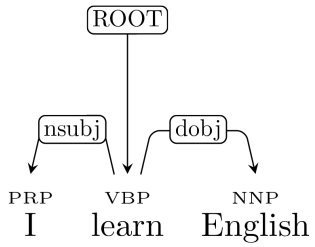


Fig. 2.2 Dependency structure

structure also allows for analysis of sentences into constituents e.g., a noun phrase is a constituent headed by a noun. Hudson (1980b) argues that dependency is necessary for syntax while constituency is unnecessary, because dependency can represent any information presented by constituency, and can easily represent a number of cross-category linguistic constraints which are difficult to describe in terms of constituency. For example, many languages tend to position dependents on one side of their heads (Heine, 1975): head-initial languages like English tend to put heads before their complement dependents (e.g., verbs precede objects and nouns precede relative clauses), whereas head-final languages like Japanese presents the opposite order. This cross-categorical syntactic phenomenon can be succinctly described in terms of the relative positioning of heads and dependents, whereas many more rules are needed to describe in terms of consistency relations (e.g., separate rules for verb phrases and noun phrases). Similarly, there is a general tendency that heads determine the position of their dependents. In English, for example, any word except a finite verb can be positioned at the beginning of a sentence as long as all the modifiers of the word move with it (e.g., we can say *In that car you can go faster* but not **In you can go faster that car*). Furthermore, dependency also provides a succinct cross-categorical explanation of SCFs: the dependents provide fillers to the syntactic frame of the heads.

Furthermore, Evans and Levinson (2009) argue that constituency is not universal because many languages (e.g., Latin and Czech) exhibit free word order in which functionally-related words (e.g., a noun and its modifier) may be non-contiguous and thus unsuitable to be combined into constituents. For these languages, dependency can better describe the syntax.

Nevertheless, for languages (e.g., English) which exhibit relatively fixed word order and naturally group words into functional constituents, constituency is a useful notion. In fact, some syntacticians have used constituency to characterize the head of a dependency relation. For example, Robinson (1970) identifies the head as the word which characterizes the constituent involving the two words. Similarly, Schubert (1987) describes the head as the word whose syntactic properties determines the behavior of the combination of the two words.

Meanwhile, it is sometimes difficult to decide whether a dependency relation exists between two words, or if a relation exists which word should be the head. In this case, grouping the words together as a constituent can avoid forced assignment of a head.

While the basic notions of constituency and dependency seem straightforward, determination of specific syntactic relations can be complicated and debatable. In the constituency framework, there is a tension between the introducing concept of constituent and the definition of constituent based on syntactic trees: the former usually equates constituents with phrases (e.g., *speak English* in *I can speak English*) whereas the latter tends to equate constituents with tree nodes which include sub-phrasal elements (e.g., *can* in the aforementioned example). A number of tests involving omission, insertion, substitution or permutation on sentence structure have been proposed to determine whether part of a sentence is a constituent. Commonly used constituent tests include topicalization (e.g., *Speak English, I can*), proform substitution (e.g., *I can do so* where *do so* substitutes for *speak English*) and answer fragments (*e.g., What can you do? Speak English.*) etc. (see Osborne, 2015, for a full survey of constituent tests). These tests sometimes yield inconsistent results and tend to recognize phrasal constituents more than sub-phrasal constituents.

Similarly, determination of dependency relations can be difficult. Tesnière (1965) describes the dependency relations between four word categories – nouns are the dependents of verbs, adjectives the dependents of nouns, and adverbs the dependents of adjectives – but provides no justification for the existence and directionality of these relations. Schubert (1987) defines that a dependency relation exists between two words co-occurring in the same sentence where one word makes the occurrence of the other syntactically possible; the category of the dependency relation is decided distributionally by grouping interchangeable dependents (e.g., *many people* and *they*) into a class and observing what definitively characterizes the relation between the class and the head. However, in some cases, it can be difficult to decide which word makes the occurrence of the other word syntactically possible. Hudson (1980a) proposes a rule to distinguish the dependent from the head: the dependent is optional in the presence of the head. For example, adjectives are optional in the presence of nouns; for the total class of verbs, the object is also optional. The rule, nevertheless, cannot distinguish the relation between demonstratives, numerals, and nouns, each of which can be a subject alone without the others in many languages including English (e.g., *Those/Three/Men came.* Owens, 1984). This problem may be partly solved by introducing another rule, which is to maintain the cross-categorical consistency in putting dependents on one side of heads. According to this rule, demonstratives are the head of nouns, and auxiliaries the head of main verbs (see Hudson, 1984, for analysis). However, such relation between auxiliaries and main verbs seem to contradict the first rule about optionality. Alternatively, Owens

(1984) proposes a semantic rule: the head is the item referentially central in a construction. Accordingly, demonstrative and numerals are the dependents of nouns, whereas auxiliaries are the dependents of main verbs. However, the relation between demonstrative and numerals remains unsolved. To date, linguists have different stances on the directionality of the relation between content words on one hand, and function words including auxiliaries, complementizer, coordinators, and prepositions on the other hand. Dependency theories including Meaning-Text Theory (Mel'čuk, 1988) and Word Grammar (Hudson, 1984) insist that function words should be the head, whereas the Functional Generative Description of the Prague School (Sgall et al., 1986) subordinate auxiliaries to content verbs. In face of the difficulties in defining directed dependency relation, Owens (1984) proposes that a dependency relation is primitively bilateral, and the head is a derivational concept for the word which has more bilateral dependency relation in general.

In fact, there is a general agreement on the syntactic relations between prototypical syntactic categories of words, e.g, nouns, verbs, and adjectives. Contrastingly, the syntactic relations involving non-prototypical syntactic categories of words are more debatable, which may be partly attributable to the intermediacy in the grammatical behavior and semantic content of the non-prototypical syntactic categories of words, as has been reviewed in Section 2.1.1.

Despite the theoretical disagreement mentioned above, computational linguists have come up with extensive marking schemes for syntactic structure. The English PTB, the first large-scale corpora annotated with syntactic structure (Taylor et al., 2003), follows a constituency scheme which includes 14 non-terminal constituency types (e.g., "ADJP" for adjective phrases, and "SBAR" for clauses introduced by subordinating conjunctions) (Marcus et al., 1993). A number of algorithms have been developed to convert constituency structure to dependency structure (e.g., De Marneffe and Manning, 2008b; Johansson and Nugues, 2007; Yamada and Matsumoto, 2003). The resulting dependency schemes vary in the inventory size and the directionality of the dependency relations between function words and content words. For example, the Stanford typed dependencies (SD) scheme (De Marneffe and Manning, 2008b) has the largest inventory of 49 dependency labels, whereas the LTH scheme (Johansson and Nugues, 2007) has only 22; the SD scheme subordinates function words to content words (e.g., auxiliaries are the dependents of verbs, and coordinators the dependents of conjuncts), whereas the LTH scheme assumes the opposite. The SD scheme has also been used to annotate the English Web Treebank corpus (Silveira et al., 2014). Partly due to the popularity of PTB and the Stanford parser which provides a convenient converter from constituency to the SD dependency scheme, SD has become a de facto standard for annotating English dependencies.

There are also many treebanks for languages other than English. To facilitate cross-lingual comparison and multilingual processing, NLP researchers have been trying to come up with universal syntactic schemes. Due to the descriptive power and flexibility of dependency structure, much effort has been focused on developing universal schemes for dependency structure. Based on the SD scheme for English, researchers developed the Universal Dependency scheme (Nivre et al., 2006). UD includes 37 universal dependency relations and a number of language-specific relations. The scheme has been used to annotate 102 treebanks for 60 languages so far (https://universaldependencies.org).

2.1.3 Subcategorization

Subcategorization specifies the syntactic contexts in which a word of a particular category may appear (Culicover, 1982). More specifically, the constituency framework defines a subcategorization frame (SCF) as the number and types of syntactic complements required by a predicate (Chomsky, 1965). This definition does not consider subjects as part of an SCF. Contrastingly, the dependency framework defines an SCF as valency which includes subjects as complements (Tesnière, 1965). The distinction between the frameworks, nevertheless, is unimportant for languages where the presence of subjects is unrelated to SCFs. In English, for example, the presence of subjects is determined by whether the clauses are imperative or not. As a result, the rest of the thesis does not include subjects in SCFs. To illustrate, in (b), the SCF of *put* consists of a direct object *the pen* and a prepositional complement *on the chair*.

(b) She **put** [the pen] [on the chair].

(c) She **walked** (in a hurry).

(d) She can **sing** [the song].

(e) She **fought** [her corner].

(f) She **opened** the door (with the key).

(g) She **likes** writing, (as you know).

SCF requires distinction between complements and adjuncts. Complements are expected to complete the meaning of the predicate, while adjuncts are more peripheral and complete the meaning of the whole predication. The most common test for distinguishing complements and adjuncts is the elimination test (Helbig and Schenkel, 1991): an element is eliminated from the sentence; if the remaining sentence is ungrammatical, the element is a complement;

otherwise, the element is an adjunct. For example, the prepositional object *on the chair* in (b) is a complement of *put* because it cannot be eliminated; contrastingly, the prepositional object *in a hurry* in (c) is an adjunct because it can be eliminated. In the examples, we boldface the predicates, indicating the complements with square brackets and the adjuncts with round brackets.

The elimination test, however, cannot distinguish complements and adjuncts in all cases. For example, in (d) the nominal object *the song* can be eliminated, but is not an adjunct because *the song* essentially completes the meaning of the predicate, and there is a slight difference in the meaning of the predicate *sing* between the sentences with and without the object: the former restricts the referred ability of singing with regard to a particular song, whereas the latter refers to the general ability of "singing well". This illustrates that a complement can be optional, and its presence or absence can affect the meaning of the predicate.

To better distinguish complements and adjuncts, a number of other tests that involve insertion, substitution or permutation on sentence structure have been proposed (Brinker, 1972; Emons, 1974; Engel and Schumacher, 1978; Herbst, 1984; Somers, 1984; Steinitz and Lang, 1973). These tests may be applicable to different types of elements (e.g., distinguishing complements and adjuncts for prepositional phrases only), and sometimes yield inconsistent results. This is related to the complicated nature of the relation among a predicate and the co-occurred elements, which is affected by semantic relation as well as structural restriction imposed during the conceptualization of real-world events (Croft, 1991). In fact, complements and adjuncts are more like prototypes on a spectrum, where intermediate cases, as well as more extreme cases on both ends, can be found. For example, Somers (1984) further distinguishes six categories along the complement-adjunct spectrum. The first category is integral complements, as illustrated in (e) where the complement *corner* is lexically determined and cannot be replaced by even its synonym such as *nook*. The second category is obligatory complements such as (b). The third category is optional complements such as (d). The fourth category is middles, as illustrated in (f). The prepositional phrase *with the key* can be eliminated without affecting the meaning of the predicate *open*; however, *with the key* is compatible with one semantic aspect of *open* – to move an object so as to change access to a space, and can be used with other verbs that have the same semantic aspect (e.g., *close* and *unlock*), but not with the verbs without this semantic aspect (e,g., *kick*). The fifth category is adjuncts such as (c). The final category is extraperipherals, which are usually discourse markers such as *as you know* in (g). The first three categories can be considered as complements in the traditional sense. The fine-grained distinction helps to explain why some tests fail to deliver consistent results, especially for middles which are neither complements

nor adjuncts. Nevertheless, Somers (1984) cautioned that there are still grey areas between the six fine categories.

Meyers et al. (1996) summarize previous studies and come up with a set of sufficient conditions and rules of thumb for distinguishing English complements and adjuncts. These criteria involve some of the previous tests (e.g., elimination test), semantic consideration (e.g., thematic roles and selectional restriction), syntactic features (e.g., "subordinate clauses headed by *before* are usually adjuncts") as well as co-occurrence frequency of the predicate and the elements, i.e., complements tend to occur with the particular predicate more frequently, whereas adjuncts tend to occur with many predicates at a similar frequency. Among these criteria, the rules on selectional restriction are debatable. Meyers et al. (1996) defines selectional restriction as presupposition associated with one constituent of a phrase about the nature of the other constituents. He then proposes that the selectional restriction can be observed by assuming semantically incompatible sentences to be correct, e.g., in a metaphorical way, and see how the meaning of the predicate and the target element changes. If the meaning of the element changes, the predicate imposes selectional restriction on the element and the element is a complement; otherwise, the element is an adjunct. Meyers et al. (1996) illustrates the selectional restriction of predicates on complement with (h), assuming that only the meaning of *idea* is changed when understanding the sentence in a semantically acceptable way. However, it can also be the case that the meaning of the predicate *tickled* changes, e.g., to "found some funny faults of the idea". Similarly, Meyers et al. (1996) illustrates the selectional restriction of adjuncts on predicates with (i), assuming that only the meaning of *learned* is changed to make the sentence acceptable. However, one can also assume that the meaning of *hammer* is changed, e.g., to "a hammer that has some math equations on its surface". Consequently, the operationalization of the rules on selectional restriction does not seem to work. The problem might lie in the fact that the semantic relation between predicates and co-occurred elements is complicated and the presupposition can be bi-directional between them (Owens, 1984).

(h) *John **tickled** [the idea].

(i) *Mary **learned** math (with a hammer).

Nevertheless, the other criteria proposed by Meyers et al. (1996) are operationally feasible and comprehensive. Meyers et al. (1996) conducted an annotation experiment about these criteria on four annotators. The result was promising: on average, 91% of the complements classified by an annotator was classified the same by the other annotators. This means that the criteria can help to distinguish complements and adjuncts consistently.

Table 2.3 Examples of English diathesis alternations

Alternation	Sentence	SCF
Transitivity alternation	She can **sing** [the song].	V-O
	She can **sing**.	V
dative alternation	She **gave** [a book] [to me].	V-O-P
	She **gave** [me] [a book].	V-O-O
causative/inchoative alternation	She **broke** [the vase].	V-O
	The vase **broke**.	V
locative alternation	She **sprayed** [paint] [on the wall].	V-O-P(on)
	She **sprayed** [the wall] [with paint].	V-O-P(with)

Note that the tests for distinguishing complements and adjuncts are conducted for specific predicates. In other words, an element can be a complement to one predicate, and an adjunct to another predicate. For example, *in Cambridge* is a complement for *lives* in (j) but an adjunct for *plays* in (k). This reveals a close connection between complements and the semantics of the predicate.

(j) She **lives** [in Cambridge].

(k) She **plays** badminton (in Cambridge).

In fact, many modern linguistic theories have analyzed SCFs as a phenomenon of linking between syntax and the semantic representation of the predicate (Croft, 1991; Jackendoff, 1992; Levin and Hovav, 1995; Pinker, 1989; Pustejovsky, 1991). From this projectionist perspective, the predicate is responsible for the form and meaning of SCFs, and the semantics of the predicate can be decomposed into a structure of elementary features, e.g., an argument structure involving AGENT, THEME, and RECIPIENT etc., or a semantic structure involving ACT, CAUSE, HAVE, MOVE, etc. Moreover, predicates that share similar semantic structure have similar SCFs. For example, verbs that share the semantic structure of "X acts on Z for Z to have Y" such as *send* and *pass* can be used with the same double-object SCF ("V-O-O", e.g., *She sent me the card* and *she passed me the ball*). Furthermore, Levin (1993) argues that diathesis alternation, the appearance of the same verb in different SCFs, reflects the semantics of the verbs and that verbs can be grouped by diathesis alternation behavior. Table 2.3 shows some typical diathesis alternations in English.

Contrastingly, Goldberg (1995, 2006) argues that SCFs are constructions that have meaning in themselves. A construction is a surface form that has a direct mapping to semantic and pragmatic content which cannot be derived from the components of the surface form. Table 2.4 illustrates the SCF constructions which encode the meaning of basic event types in human experience (Goldberg, 1999). Each SCF construction is illustrated with an

Table 2.4 English SCF construction (Based on Goldberg, 1999)

Construction	SCF	Meaning	Example
Intransitive motion	V-P	X moves to Y	They **buzzed** [into the room].
Transitive	V-O	X acts on Y	She **eyebrowed** [her surprise].
Resultative	V-O-X	X causes Y to become Z	She **kissed** [him] [unconscious].
Double object	V-O-O	X causes Y to receive Z	She **overnighted** [him] [the book].
Caused-motion	V-O-P	X causes Y to move Z	Pat **sneezed** [the foam] [off the coffee].

example of which the predicate verb is used in an unusual way. For example, *eyebrow* is usually a noun but can be understood as a verb which means "using eyebrow to express" in the transitive construction. In this case, the SCF is unlikely to derive from the semantics of the predicate. Such examples demonstrate that SCF constructions have independent meaning and can cause the meaning of the predicate to change.

Apart from the semantic source of SCF, modern syntactic theories also have varied ideas about the taxonomy of SCFs. For example, lexicalist syntactic theories (e.g., HPSG and LSG) encode more syntactic information about SCFs in the lexicon than syntactocentric theories (e.g., Transformational Grammar) (Korhonen, 2002). Furthermore, SCFs can be classified with varied granularity, e.g., whether the SCFs are lexically parameterized for prepositions and particles or not.

Computational linguists have developed varied SCF schemes when constructing large-scale lexicons for real-world NLP applications. Representational works include the manually constructed computational lexicon Comlex (Grishman et al., 1994), which has 92 SCF categories for verbs, and the Alvey Natural Language Tools dictionary (ANLT) (Boguraev and Briscoe, 1987), which is manually adapted from the electronic version of Longman Dictionary of Contemporary English (Procter, 1978). By merging and supplementing the SCF schemes of Comlex and ANLT, Briscoe and Carroll (1997) developed a detailed scheme of 163 SCF categories for verbs (See Appendix A of Korhonen, 2002, for details); this scheme is subsequently extended to 168 SCF categories by Preiss et al. (2007).

2.2 Second language acquisition of syntax

Second language (L2) researchers are interested in investigating how humans acquire the syntax of language, and how this L2 acquisition process is affected when the learner has the pre-knowledge of a different language. Analysis of L2 syntactic acquisition provides a gateway to testify and develop theories in linguistics and psychology, and has important implication for L2 education. This section reviews two important aspects of L2 syntactic acquisition: SCFs and syntactic complexity.

2.2.1 L2 acquisition of subcategorization

The syntactic construct of SCF has attracted much attention in L2 research. This is due to the central role of SCF in many syntactic theories and the close connection between SCF and semantics. Studying L2 acquisition of SCFs can not only reveal how L2 learners develop SCFs, but also provide empirical insights into human linguistic capacity and cognitive mechanism. For example, research on L2 SCF acquisition has helped to testify claims about innate knowledge of argument structure (Pinker, 1989) in the framework of Universal Grammar (UG, Chomsky, 1965), and to explore what specific rules might be included in the innate knowledge and whether they can be reset in L2 acquisition (Bley-Vroman and Yoshinaga, 1992; Joo, 2003; Juffs, 1998; Montrul, 1998; White, 1987, 1991). Research on L2 SCF acquisition has also helped to testify the psychological reality of construction in the framework of constructional grammar (Gries and Wulff, 2005), and to examine usage-based theory about constructional learning (Ellis and Ferreira–Junior, 2009; McDonough and Nekrasova-Becker, 2014). Furthermore, researchers have examined whether cognitive constructs such as working memory and statistical learning can account for L2 SCF acquisition (McDonough and Trofimovich, 2016).

Meanwhile, studies of L2 SCF acquisition can have practical implications for L2 education. Researchers have investigated how learner SCFs develop over time (Ellis and Ferreira–Junior, 2009; Tono, 2004), whether there is difference between L1 and L2 acquisition of SCFs (Bley-Vroman and Yoshinaga, 1992), and how various factors including L1 background (Bley-Vroman and Yoshinaga, 1992; Joo, 2003; Juffs, 1998; Montrul, 1998; White, 1987, 1991), L2 input (Ellis and Ferreira–Junior, 2009; McDonough, 2006; McDonough and Nekrasova-Becker, 2014), and verb semantics (Bley-Vroman and Yoshinaga, 1992; Ellis and Ferreira–Junior, 2009; Gries and Wulff, 2005; Römer et al., 2015, 2014) affect L2 acquisition of SCF (Tono, 2004). These results can inform the development of educational materials, curricula and pedagogical strategies for L2 education.

Most research on L2 acquisition of SCFs rely on experiments with humans. These experiments include grammaticality judgment (Bley-Vroman and Yoshinaga, 1992; Inagaki, 1997; Juffs, 1996, 1998; Souza, 2011; White, 1987, 1991) multiple-choice interpretation (Joo, 2003; Montrul, 1998), self-paced reading (Juffs, 1998), elicited production (Juffs, 1996; McDonough, 2006), sentence completion (Gries and Wulff, 2005) and sentence sorting (Gries and Wulff, 2005; Kim et al., 2017; Kim and Rah, 2016; White, 1991). While these experiments allow for control over various factors, they are highly restricted in scope. First of all, the number and L1 background of the learners are limited. Most experiments involve only dozens of L2 learners and investigate only one or two L1s; the few exceptions that involve learners across several L1s (Juffs, 1998; White, 1987; Zobl, 1989) group the limited number

of learners together during analyses rather than investigate the distinct effect of the different L1s on the L2 acquisition of SCFs. Second, the target SCFs are limited. Some studies investigate SCFs in only one or two types of diathesis alternations – the most researched one being dative diathesis (Bley-Vroman and Yoshinaga, 1992; Inagaki, 1997; McDonough, 2006; McDonough and Nekrasova-Becker, 2014; McDonough and Trofimovich, 2016; Montrul, 1998; White, 1987, 1991), followed by locative alternation (Bley-Vroman and Joo, 2001; Joo, 2003; Juffs, 1996), transitivity diathesis and causitive/inchoative diathesis (Juffs, 1998). Other studies investigate some of the constructions listed in Table 2.4 (Gries and Wulff, 2005; Kim et al., 2017; Kim and Rah, 2016; Souza, 2011; Zobl, 1989). In these experiments, the SCFs are instantiated with only a few verbs and prepositions – if the SCFs involve prepositions.

Recent years have seen an increasing use of the corpus-based approach to investigating the L2 acquisition of SCFs, which has enlarged the research scope. The development of such approach is partly driven by the increasing popularity of Construction Grammar (CG) – the usage-based theory from CG posits a close relation between the frequency of an SCF construction and its acquisition, and corpora provide a good way to examine this relationship. Nevertheless, it should be noted that the corpus-based approach is theory-neutral (Tono and Díez-Bedmar, 2014), and can be used to testify theoretical claims in other linguistic frameworks. To date, the corpus-based approach to the L2 acquisition of SCFs mainly obtains SCF information from L2 texts by manual or semi-manual annotation. Semi-manual annotation is conducted by tagging or parsing corpora with a POS tagger or parser, searching for potential patterns with the POS tags (Ellis and Ferreira–Junior, 2009) or syntactic labels (Meurers et al., 2013; Tono, 2004), and manually editing the results. Such approach allows for the utilization of more naturalistic data (Gilquin and Gries, 2009) at a larger scale than human experiments. As a result, corpus-based studies have enlarged the research scope in terms of the variety of SCFs and the instantiations of the SCFs than experiment-based research: for example, Tono (2004) investigated various SCFs used with the top ten most frequent verbs by Japanese learners of English; Meurers et al. (2013) investigated SCFs involved in 21 binary diathesis; Ellis et al. (2014) and Römer et al. (2015, 2014) investigated the instantiation of the SCF of "V-P" lexicalized by around 20 different prepositions and associated with a large number of verbs.

However, the current methods of SCF annotation are time-consuming, which have restricted the power of the corpus-based approach. Even with the help of general NLP tools, annotators still need to spend a long time in searching and annotating SCFs. Meurers et al. (2013) point out that only 22 out of the many diathesis pairs defined by Levin (1993) can be readily searched by parse information. Furthermore, human effort is required to

distinguish arguments and adjuncts. The time-consuming process of semi-manual annotation is illustrated by the study of Römer et al. (2015): Römer et al. (2015) parsed BNC with the RASP dependency parser, and defined search rules for the "V-P" type of SCFs based on POS tags and the dependency labels. The search rules were initially defined according to COBUILD (Francis et al., 1996) descriptions of target SCFs, but these rules were not enough for distinguishing between adjuncts and complements; furthermore, the search accuracy suffered due to the parsing errors. To improve the search accuracy of SCFs, Römer et al. (2015) conducted three rounds of search refinement, each round involves the definition of search rules, the post-edition of the search results (1,500 sentences for each SCF pattern) and the evaluation of the precision and recall of the search. After the painstaking effort, Römer et al. (2015) achieved an average of 78% precision, 53% recall and 61.2% F1 score across all SCF types. The time-consuming effort has restricted the amount of SCF data available for research. As a result, the number of learners and their L1s, as well as the types of the SCFs that have been investigated by the corpus-based studies, are still limited. Much power of corpora stays untapped, which has prevented a data-driven approach to discovering laws that might underlie L2 SCF acquisition at a macro level (Cai and Liu, 2017).

The problem can be alleviated by developing an automatic analysis tool for SCFs. Currently, the only attempt to exploit SCF information in a completely automatic manner was made by Kyle (2016), who developed Tool for the Automatic Analysis of Syntactic Sophistication and Complexity (TAASSC) to calculate the syntactic sophistication of English L2 texts based on the reference frequency of verb argument constructions (VACs), i.e., SCFs, in the Corpus of Contemporary American English (COCA) (Davies, 2008). However, TAASSC uses a dependency parser, and operationalizes VAC as all the direct dependents of a main verb. This means VAC is operationalized in a way that does not distinguish between arguments and adjuncts. For example, the prepositional phrase *in a hurry* in sentence (c) would be taken as part of the VAC of the predicate *walked*. Note that the complement-adjunct distinction is not only important for the theoretical definition of SCF, but also vital to any L2 study of SCF that concerns the relation between predicates and verbs, or concerns the meaning of a SCF construction. This is because complements have a close relation to the predicate, and strongly indicate the meaning of the predicate and the SCF; contrastingly, adjuncts can be used with many predicates freely, and have no such indication. Meanwhile, the operationalized VAC of TAASSC captures the surface structure rather than the deep structure. For example, a passive structure is considered as a different VAC from its active equivalent.

2.2.2 L2 syntactic complexity

Syntactic complexity is another important construct in L2 research. Complexity, along with accuracy and fluency (CAF), has emerged as a principal angle for defining or characterizing language proficiency (Norris and Ortega, 2009). Due to the central role of language proficiency in L2 research and education, numerous studies have been conducted on what cognitive processes or mechanism may underlie the development of L2 syntactic complexity (Robinson, 2005; Skehan, 1998; Wolfe-Quintero et al., 1998), and how various learner and educational factors may affect the development of L2 syntactic complexity (Bulté and Housen, 2012).

A fundamental issue of research on L2 syntactic complexity concerns how to operationalize and measure syntactic complexity (Bulté and Housen, 2012). Syntactic complexity is multi-dimensional, which follows naturally from the fact that syntax can be analyzed into different layers and involves different components. Syntactic complexity is commonly defined as the ability to use a wide range of sophisticated structures in L2 (Bulté and Housen, 2012; Wolfe-Quintero et al., 1998). Accordingly, there are two basic and complementary aspects of syntactic complexity: the breadth (or diversity) of linguistic structures, and the depth (or sophistication) of linguistic structures. The breadth aspect of syntactic complexity is usually measured by the number of the discrete components of a linguistic unit. According to the size of linguistic units for consideration, the syntactic diversity measures that have been used in L2 research can be roughly classified into sentential, clausal, and phrasal complexity (Bulté and Housen, 2012; Kyle, 2016). Examples of such measures include clauses per sentences, i.e., the average number of clauses in a sentence, and the mean length of clauses (MLC), i.e., the average number of words in a clause. The exact definitions of different levels of linguistic units may differ. For example, the sentence may be replaced by the T-unit, which is defined as a main clause plus any subordinate clause or nonclausal structure attached to or embedded in the main clause (Hunt, 1970). The sentence may also be replaced by the utterance, which is defined as a continuous piece of speech beginning and ending with a clear pause (Brown, 1973). The linguistic units vary according to the purposes and context of syntactic complexity analysis – as Norris and Ortega (2009) point out, the T-unit may be more suitable for intermediate or advanced written data, whereas the utterance is more appropriate for speech data. Furthermore, the linguistic unit and its components may be specified, e.g., the number of adjectival modifiers in a noun phrase (Kyle, 2016).

The depth aspect of syntactic complexity is measured by the difficulty of syntactic structures. The difficulty is defined in relation to language users, e.g., in terms of how they perceive a structure or when they acquire the structure. For example, American learners of German L2 acquire relative clauses later than coordinate clauses (Sinicrope and Byrnes, 2009), which indicate that relative clauses are more difficult than coordinate clauses. However, since

the definition of syntactic sophistication depends on language users, the sophistication of a syntactic structure may vary across different language users, subject to influence from L1 background, language aptitude, memory capacity, and motivation, etc. (Housen and Simoens, 2016). It is therefore important to clarify what group of language users are involved in investigating syntactic sophistication. Meanwhile, syntactic sophistication involves some factors which are independent of language users, such as the perceptual saliency, the frequency of the structure in the input, and the communicative load etc. (Bulté and Housen, 2012). These factors can be taken as partial indicators of syntactic sophistication. For example, Kyle (2016) approximates the input frequency of syntactic structures with their frequency in COCA, using the frequency as an indicator of syntactic sophistication: the higher frequency the frequency, the lower the sophistication.

The information provided by a syntactic complexity measure depends on the size, specificity, and nature of the linguistic unit and its components. Large-grained measures concerning the generic word components of the sentence level linguistic unit (e.g., the mean length of sentence, T-unit or utterance, i.e., MLS, MLT and MLU), provide a holistic view of syntactic complexity – longer sentences generally correlate with higher syntactic complexity. However, what exactly contributes to the length of sentences remain concealed. The contributing factors might be the use of more elaborate noun phrases, and/or extensive use of subordination, etc. To understand the specific dimensions of syntactic complexity, more specific measures are needed. This is especially important because different aspects of syntactic complexity do not necessarily develop linearly during L2 development. For example, Norris and Ortega (2009) mention that the amount of coordination increases at the beginner level of L2 development and subsides at the intermediate level; this makes coordination a powerful indicator of the syntactic complexity at the early stage of L2 development. Similarly, subordination and phrasal elaboration are most powerful for characterizing the intermediate level and the advanced level respectively. Furthermore, the most indicative syntactic complexity measures also differ across various types and genres of production. For example, clausal indices (e.g., frequency of *that* clauses) are more discriminative for informal speech, while phrasal elaboration indices (e.g., frequency of prepositional phrases as nominal postmodifiers) are more discriminative for academic writing (Biber et al., 2011). As a result, developing syntactic complexity measures that tap into different syntactic aspects are important for gaining a comprehensive view of L2 syntactic complexity and achieving a better discrimination for L2 proficiency in different situations.

Meanwhile, whether a syntactic complexity measure is widely used in research and educational practices depends on the availability of automatic tools for calculating the measures. Syntactic complexity measures are essentially statistical and require annotation of

the target syntactic constructs. Manual annotation of the syntactic constructs is costly and becomes almost infeasible for large data. Partly due to the ease of automation, word-based large-grained measures such as MLS and MLT have been most widely used. Recent development in computational tools has helped to popularize other syntactic measures. Lu (2010) developed Syntactic Complexity Analyzer (SCA) which can calculate 14 traditional sentential and clausal measures; some of these measures are related to subordination or coordination (e.g., dependent clauses per T-unit and coordinate phrases per clause). Kyle (2016) developed TAASSC which can calculate 372 syntactic complexity measures. The expanded categories mainly include the measures of fine-grained clausal and phrasal complexity, as well as the measures of syntactic sophistication based on the frequency of verbs and VACs (operationalized as not distinguishing between adjuncts and complements), and verb-VAC contingency (i.e., the probability that a verb and a VAC co-occur). Both tools use standard parsers developed for native English to extract information of syntactic structure.

At a first glance, there are numerous complexity measures available for researchers at the moment, covering various aspects of syntactic structure. A closer examination would reveal that some important syntactic phenomena are not captured yet. SCFs, which have important theoretical and practical values in L2 research (as reviewed in section 2.2.1), have not been examined. This is partly due to the lack of analysis tools for SCFs. Kyle (2016) tried to approximate SCFs with VACs that do not distinguish between adjuncts and arguments, and calculated VAC-based syntactic sophistication measures. While his research attempt has contributed to the investigation of syntactic complexity from the perspective of SCFs, more accurate tools for analyzing SCFs are needed to explore SCF-based syntactic complexity measures.

2.3 Syntactic analysis of learner language

As exemplified in research on the L2 acquisition of SCFs and syntactic complexity, the syntactic analysis of learner language is important for L2 research. Two problems are prominent in the syntactic analysis practices. First, L2 researchers resort to generic syntactic analysis systems such as POS taggers and parsers to annotate SCFs, which require extensive human post-edition. The costly post-edition has restricted the amount of SCF annotation available, leaving much power of the corpus-based approach untapped. Second, the POS taggers and parsers used by L2 researchers to analyze learner language were developed on native language data. Since learner language is different from native language, the performance of standard systems on learner language might suffer, which can influence the

accuracy of the empirical evidence for L2 research. It is therefore important to evaluate how standard systems perform on learner language.

This section first reviews important issues to consider for the syntactic analysis of learner data. To narrow the scope of this thesis, we point out the analysis approach and the type of syntactic structure (i.e., dependency structure) focused in this thesis. We then survey existing standard POS taggers and parsers, and review previous evaluation of standard POS taggers and dependency parsers on learner data. Finally, we survey existing NLP systems with regard to SCFs.

2.3.1 Challenges in analyzing L2 syntax

Syntactic analysis of learner language is challenging. As mentioned earlier, learner language is more variable than native language due to differences in the L2 proficiency of learners, their L1 background and learner errors. As a result, it is difficult to develop a comprehensive syntactic framework to describe learner language (Meurers and Dickinson, 2017). Crucially, learner errors may give rise to contradictions between morphosyntactic and semantic cues for syntactic analysis, causing syntactic ambiguity. Consider the learner sentence (l) as an example. This sentence can be regarded as a well-formed sentence, where *live* is an adjective referring to the state of being broadcasting. This interpretation relies on the morphosyntactic clue that *live* should be an adjective before a copula. However, if the context of the sentence indicates that the learner was introducing his or her hometown, then the intended meaning is closer to *I live in Manaus*, in which case *live* is a verb.

(l) I am live in Manaus.

To address such challenges posed by learner errors, it is necessary to analyze the ambiguous syntactic structures consistently and expressively: contradictory cues should be handled by consistent principles, whilst the syntactic scheme should be expressive in reflecting non-canonical syntactic structures. Researchers have tried to achieve consistency in two ways: the first one is to design some rules to weigh different cues and disambiguate the syntactic structures (Berzak et al., 2016b; Nagata and Sakaguchi, 2016). For example, if we decide that contextual clues should be prioritized over morphosyntactic clues, *live* in sentence (l) is disambiguated as a verb. The second way is to develop separate analysis layers for different cues (Ragheb and Dickinson, 2012, 2014). For example, we can define two analysis layers for contextual and morphosyntactic clues respectively, and analyze *live* as an adjective and a verb in each layer. The multi-layer method can provide richer information about learner language. Nevertheless, how to design the layers remains a challenge. In their attempt to

develop multi-layer parsing schemes for learner language, Ragheb and Dickinson (2012, 2014) found that some ambiguous structures may allow for many possible analyses, and it is infeasible to set a layer for each analysis; furthermore, some layers might be redundant while there can be conflicting clues to consider within a layer.

Another important issue concerns whether native language terminology can be used to analyze learner language. Some researchers argue that learner language (or interlanguage, IL) is a system different from the native language (or target language, TL), and learner language should be analyzed with a scheme developed for itself (Bley-Vroman, 1989; Ragheb and Dickinson, 2011). However, it seems impossible to abandon the syntactic terminology of native language completely when analyzing the syntax of learner language. Since learner language develops with native language as a goal and has many characteristics similar to native language (Ellis, 1994), even if one developed a syntactic scheme for learner language from scratch, the resulting scheme may turn out similar to the native language syntactic scheme. In fact, most research on developing unique syntactic schemes for IL follow native language terminology; the uniqueness of syntactic schemes mostly lie in the design of multiple analysis layers (Dickinson and Ragheb, 2015; Ragheb and Dickinson, 2012; Rosen et al., 2014).

Meanwhile, existing native language syntactic schemes can be flexible and expressive in reflecting many non-canonical structures. To illustrate, the Stanford typed dependencies (SD) (De Marneffe and Manning, 2008a) scheme have dependency labels for describing loose structure, e.g., "parataxis", which can be used to analyze the common un-canonical learner structure of comma splice. For the description of learner SCFs, our annotation practice (Section 5.1) also reveals that most learner SCFs, including those with learner errors, can be described by the native language syntactic scheme. In fact, even though native language syntactic schemes may not explicitly present all levels of information about non-canonical structures in learner data, by defining the analysis rules clearly (e.g., what clues should be prioritized when conflicting clues are present), analysis in native language schemes can still be useful for L2 researchers who are aware of the analysis rules and tailor the analysis results to their own research goals accordingly (Meurers and Dickinson, 2017). This is proved by the fact that some L2 researchers have successfully extracted information from learner data using syntactic analysis in native language schemes (Crosthwaite, 2016; Murakami and Alexopoulou, 2015; Römer et al., 2014; Vyatkina, 2013).

To summarize, researchers need to decide the analysis approach and scheme for syntactic analysis of learner language. To narrow the subsequent discussion, we describe our choice here. We adopt native language syntactic schemes and specify rules to disambiguate structures on one analysis layer. Specifically, we prioritize semantic cues, using the semantic context

Id	**Word**	**POS**	**Head Id**	**Dependency**
1	I	PRP	2	nsubj
2	learn	VBP	0	root
3	English	NNP	2	dobj

Fig. 2.3 Tabular text format of dependency structure

of learner errors to hypothesize the intended meaning for structural disambiguation. Paying attention to the intended meaning helps alleviate the problem of forcing native syntax on learner language (Bley-Vroman, 1989; Ragheb and Dickinson, 2012), achieving expressive representation of non-canonical structure. Furthermore, this thesis follows the principle of not over-interpreting learner errors (Nicholls, 2003; Ragheb and Dickinson, 2014). In other words, we localize learner errors and choose the analyses closest to hypothetically correct structures according to the principle of minimal edit distance (Nagata and Sakaguchi, 2016).

Researchers also need to decide the framework of syntactic structure, i.e., constituency or dependency. Both the constituency (Nagata and Sakaguchi, 2016) and dependency (Berzak et al., 2016b; Dickinson and Lee, 2013; Dickinson and Ragheb, 2009; Geertzen et al., 2013; Krivanek and Meurers, 2011; Ott and Ziai, 2010; Ragheb and Dickinson, 2011) frameworks have been used to annotate learner data. As reviewed in Section 2.1.2, the two frameworks provide different syntactic information and are useful for different purposes. Nevertheless, they are correlated and can transform into one another by rules.

This thesis focuses on dependency structure, because it allows for a succinct way to achieve consistent and expressive analysis of learner structure. Firstly, the dependency structures of learner errors and their correct hypotheses tend to involve fewer differences in syntactic relations, which reduces the possibility of analysis errors. For example, when a learner erroneously adds a progressive auxiliary to the sentence in Figure 2.1 (*I am learn English*), the dependency analysis requires the addition of only one relation (a dependence relation labelled as "aux" pointing from *learn* to *am*). By contrast, the constituency analysis requires one relation deleted (VP -> VBP) and two relations (VP -> VP, VP -> AUX VB) added. Secondly, the word-level dependency relations allow for a tabular text representation format (Figure 2.3), which does not need special software to display, and is easy and fast to index and search by regular expressions. This makes navigating and editing documents simple for annotators. Note that dependency relations can also be transformed to other formats (e.g., XML), and searched by more sophisticated search engines, such as the graph-based engine ANNIS3 (Krause and Zeldes, 2016).

2.3.2 POS tagging and dependency parsing

In this section, we first review existing POS taggers and parsers. Due to the vast amount of NLP research on POS taggers and parsers, and the focus of the present thesis on evaluating rather than developing POS taggers and parsers for learner language, we provide a very general overview of these syntactic analysis systems. Readers who are interested in technical details can see the reference in this section for more information. We then review previous studies on evaluating automatic POS taggers and dependency parsers on learner language.

Overview of POS taggers and parsers

Early POS taggers were developed with hand-crafted rules. The first POS tagger, TAGGIT (Greene and Rubin, 1971), was developed to analyze the Brown corpus. TAGGIT uses co-occurrence rules to disambiguate POS tags. For example, an article can be followed by a noun but not a verb. The most well-known and still widely used rule-based tagger is the Brill Tagger (Brill, 1992). The Brill Tagger operates in two steps. It first tags all words with their most frequent POS tag according to a dictionary, and then uses context rules to change the tags iteratively. The Brill Tagger achieves an accuracy of 95% when tested on 5% of the Brown corpus. While a rule-based system can achieve an accurate result, it requires extensive human effort in defining the rules.

Subsequent research on POS taggers has shifted to the statistical-based approach. This change is partly driven by the release of the large-scale corpus of English PTB (7 million words annotated with POS tags, Taylor et al., 2003), which allows for the more efficient development of POS taggers based on machine learning models. Most statistical POS taggers use variants of hidden Markov model (HMM), which treats POS tags as the hidden states that generate the word sequences. HMM calculates the probability of a POS tag based on the transition probability between adjacent POS tags and the emission probability of the word given the POS tag. The state-of-the-art POS tagger combines dynamic feature induction with HMM, achieving an accuracy of 97.64% on the English PTB test set (Choi, 2016). Other POS taggers have also used machine learning models such as Maximum Entropy model (Tsuruoka et al., 2005), Support Vector Machine (Giménez and Marquez, 2004), Conditional Random Field model (Sun, 2014) and Long Short-term Memory neural network model (Huang et al., 2015).

Similar to POS taggers, research on the development of syntactic parsers has gone through a transition from the rule-based approach to the statistical-based approach. Early research focused on the development of constituency parsing algorithms, applying hand-crafted rules of context-free grammar (CFG) in a bottom-up or top-down manner (Allen,

1995). Later on, statistical parsers were developed to derive CFG rules and their probability from constituency corpora. The integration of lexical information into the probabilistic CFG grammar tremendously boosted the accuracy of constituency parsers (e.g., 88% F1 score on the English PTB test set by Collins, 1997); the accuracy has continued to increase by utilization of other fine-grained information and complicated mechanism such as re-ranking preliminary parses (e.g.,91% F1 score by Charniak and Johnson, 2005).

Recent years have seen increasing interest in multilingual processing and universal dependency, which has made dependency parsing a more active research topic than constituency parsing. Current dependency parsers mainly fall into two categories: graph-based parsers which build all possible parse trees for a sentence and find the parse tree with the highest scoring (e.g., Martins et al., 2010; McDonald et al., 2005; McDonald and Pereira, 2006), and the transition-based parsers which decompose the tree building process into elementary actions and find the sequence of actions with the highest scoring (e.g., Andor et al., 2016; Nivre et al., 2006). There are also hybrid parsers which combine graph-based and transition-based methods (e.g., Zhang and Clark, 2008). Apart from using dependency parsers directly, researchers have also tried to obtain dependency structure automatically by converting the results of constituency parsers using definitive rules (De Marneffe and Manning, 2008a; Johansson and Nugues, 2007; Yamada and Matsumoto, 2003). Kong and Smith (2014) term the method that uses dependency parsers directly as 'd-parsing', and the method that extracts dependency structure by converting constituency parsing as 'c-parsing'. Since the English PTB uses constituency structure, both c-parsing and d-parsing have been used to obtain English dependency structure.

While we review POS taggers and parsers separately, POS tags are important for parsing and many publicly available parsers actually include a POS tagger as a preprocessing component (Charniak and Johnson, 2005; Klein and Manning, 2003a,b; Petrov and Klein, 2007). Furthermore, there is much research effort to jointly train and implement POS tagging and syntactic parsing (Bohnet and Nivre, 2012; Hatori et al., 2011; Li et al., 2011).

Evaluation on Learner data

Since syntactic annotation is costly and even impractical for large corpora, automatic POS taggers and parsers are increasingly used to analyze learner corpora (Geertzen et al., 2013; Granger et al., 2009; Tono and Díez-Bedmar, 2014). However, due to the absence of POS taggers and parsers specifically developed for learner data, standard taggers and parsers developed for native language data are used to analyze learner data. Understanding how these standard syntactic analysis systems perform on learner data is important for downstream research and application related to learner language.

Nevertheless, few studies have investigated the performance of standard taggers and parsers on learner data in a systematic way. In general, limited research to date shows that standard POS taggers can achieve high accuracy on learner data. Van Rooy and Schäfer (2002) found that the accuracy scores of the TOSCA-ICLE tagger (Aarts et al., 1998), the Brill tagger (Brill, 1992), and the CLAWS tagger (Garside et al., 1997) on a 2,159-word subset of Tswana Learner English Corpus were 87%, 89%, and 96% respectively. Moreover, Geertzen et al. (2013) reported that the Stanford POS Tagger (Klein and Manning, 2003a) obtained 96.1% accuracy on an 11,067-word subset of learner English from EF-Cambridge Open Language Database (EFCAMDAT). Similarly, Rehbein et al. (2012) found that the RFTagger (Schmid and Laws, 2008) was 93.8% accurate on a 124,512-word sample of learner German.

Previous research also found that standard dependency parsers may perform with relatively high accuracy on learner data. The accuracy of a dependency parser is usually measured by the unlabeled attachment score (UAS) and the labeled attachment score (LAS). UAS refers to the percentage of words that have correct head indices, whereas LAS refers to the percentage of words whose head indices and dependency labels are both correct. Geertzen et al. (2013) found that the Stanford Probabilistic Context-Free Grammar (PCFG) parser (Klein and Manning, 2003a) obtained 92.1% UAS and 89.6% LAS on the 11,067-word subset of EFCAMDAT. Moreover, Ott and Ziai (2010) showed that MaltParser (Nivre et al., 2007) scored 79.15% LAS and 84.81% UAS on a 900-word dataset of learner German, in contrast to 83.12% LAS and 86.38% UAS on native German. Krivanek and Meurers (2011) replicated the study of Ott and Ziai (2010) and reported similar results from the WCDG parser (Foth and Menzel, 2006) as well as MaltParser.

However, in most of the studies above, the POS taggers or parsers were evaluated on the gold standards which were obtained by manually correcting the output of exactly the same POS taggers or parsers (Geertzen et al., 2013; Rehbein et al., 2012; Van Rooy and Schäfer, 2002). It is possible that the human annotation may have been biased towards the results of the analysis systems (e.g., accepting some incorrect tags produced by the systems). If present, such bias may have inflated the accuracy scores. Annotation bias may artificially increase inter-annotator consistency (Dandapat et al., 2009; Marcus et al., 1993) as a result of a shared acceptance of incorrect analysis choices (Berzak et al., 2016a; Skjærholt, 2013). Also, it has been shown that unintentional bias towards incorrect analysis choices reduces the quality of the POS annotation on native English (Fort and Sagot, 2010) as well as dependency annotation on native English and upper-intermediate learner English (Berzak et al., 2016a). As a result, it is important to investigate the extent to which human annotation bias is present and its potential impact on parser evaluation on learner data across all L2 proficiency levels.

Meanwhile, few studies have compared the performance of different POS taggers or parsers on learner data. Van Rooy and Schäfer (2002) tested three POS taggers on learner English, but the tagsets of the POS taggers were different, which made the accuracy scores of different POS taggers incomparable. The only valid comparison was made by Krivanek and Meurers (2011) between the WCDG parser and MaltParser on learner German. They found that the WCDG parser slightly outperformed the other by 1.16% UAS and 1.80% LAS. Furthermore, the WCDG parser performed better in identifying core predicate-argument relations, while the MaltParser was more successful in establishing adjunct relations. Nevertheless, no such comparison has been carried out on learner English. Moreover, there has been no attempt to investigate the potential correlation between the performance of a parser on native language and learner language.

There have been some preliminary attempts in investigating the influence of learner errors on POS tagging. Van Rooy and Schäfer (2002) found that when spelling errors were removed from learner English, the absolute accuracy scores of the POS taggers increased by 2-3%. Similarly, Rehbein et al. (2012) found that the accuracy of the RFTagger increased by 4.9% when all learner errors were removed from learner German. Furthermore, Ott and Ziai (2010) qualitatively observed that for learner German, the omission of verbs was detrimental to parsing performance, whereas learner errors on linguistic agreement or word order seldom caused parsing errors. Nevertheless, there has been no systematic investigation into the effect of fine-grained learner errors on the performance of standard parsers on learner English.

2.3.3 SCF systems

We move on to review existing NLP systems regarding SCFs. Brent (1991) proposed the first automatic SCF extraction method which can acquire six SCFs from a corpus. Since then, many NLP systems regarding SCFs have been developed. However, most of these systems are aimed to acquire SCF lexicons for native language. The entry of an SCF lexicon is usually a list or a probability distribution of SCFs for a verb form (e.g., Table 2.5 shows that the probability for *use* to take up the SCF with a direct object, i.e., "V-O", is 68%). However, these systems cannot analyze SCFs for individual verb tokens, such as the SCF of *use* in the particular sentence *I use a carrot for the snowman's nose* ("V-O-P"). The SCFs for individual verb tokens is the type of linguistic information needed in L2 research.

There are two categories of SCF acquisition systems. The first category extracts SCF patterns from either by hand-crafted rules (Briscoe and Carroll, 1997; Han et al., 2004; Korhonen et al., 2000; Manning, 1993; Preiss et al., 2007; Przepiórkowski, 2009) or co-occurrence statistics (Altamirano, 2010; Chesley and Salmon-Alt, 2006; Dębowski, 2009; Ienco et al., 2008; Kawahara and Kurohashi, 2010; Lenci et al., 2008; Messiant et al.,

Table 2.5 An example SCF lexicon entry

use	
V-O	0.68
V-O-P	0.20
V-O-X(to)	0.12

2008; O'Donovan et al., 2005), and often uses rule-based or statistical filters to remove erroneous SCFs from the output. The input to these systems is either raw corpora or corpora preprocessed by POS taggers or parsers. These techniques perform at 60-70% F-score. F-score is the harmonic mean of the percentage of correct SCFs among all the SCFs assigned by the technique and the percentage of correct SCFs that the technique recovers in the gold standard. Note that due to differences in the SCF inventories and experimental settings, the F-scores of different SCF acquisition techniques are usually not directly comparable.

The second category treats SCF acquisition as a clustering problem, using unsupervised methods including, for example, Latent Dirichlet Allocation (Lippincott et al., 2012), multi-way tensor factorization (Van de Cruys et al., 2012), Determinantal Point Processes (Reichart and Korhonen, 2013) and Markov Random Field modeling (Baker et al., 2014). When evaluated so that unlabeled SCF clusters were mapped to traditional SCFs via rules, the best of these systems also achieved F-scores between 60-70%.

There are only two NLP systems which can analyze SCFs for verb tokens. Baker et al. (2014) proposed an unsupervised method that can cluster verb tokens according to information about SCFs. The clusters are regarded as the SCFs of the verbs. However, the labels of the SCFs are unknown. Dušek et al. (2014) developed a system that can assign SCF labels to verb tokens. This system achieves 85% F-score on newswire English. However, the system applies to only a limited number of verb lemmas, because the system uses separate SCF classification models for different verb lemmas.

Meanwhile, all previous SCF systems were developed for native language data. The SCF lexicon extracted from native language data may not describe the distribution of SCFs in learner data properly, and the performance of an SCF identification system trained on native language data may drop on learner data. This is because learner language is quite different from native language. Learner language contains un-canonical structures and possibly more simple SCFs.

In sum, there is a need to develop an SCF identification system which can analyze SCFs for unlimited individual verb tokens in context and is adapted for learner data.

2.4 Summary

In this chapter, we reviewed the theoretical concepts of POS, syntactic structure, and SCF, as well as the major taxonomies of these syntactic constructs used by computational linguists. We then reviewed L2 research on SCF and syntactic complexity to establish the importance of syntactic analysis of learner language in L2 research. We also pointed out that evaluation of standard POS taggers and parsers on learner language as well as an automatic SCF identification system is needed for L2 research. We then reviewed existing POS taggers and parsers, and surveyed previous evaluations of standard POS taggers and dependency parsers on learner language. Finally, we reviewed existing NLP systems related to SCFs.

As we can see from the literature review, there is a need to investigate whether annotation bias exists in the common practice of constructing a gold standard by manually correcting the pre-annotation of a single parser. There is also a need to systematically analyze the impact on fine-grained learner errors on parser performance, as well as a need to compare between multiple parsers on learner data and native data. Furthermore, we pointed out a need to develop an SCF identification system for L2 research, and the need to develop SCF-based complexity measures. The rest of this thesis will report how we address the aforementioned issues.

Chapter 3

Data

This chapter describes the datasets used in the thesis. We employ learner English and native English datasets. We investigate three types of syntactic constructs, i.e., POS, dependency structure and SCFs, using native English datasets that have been annotated with these constructs. We then annotate a subset of the learner English dataset for these constructs, learner errors, and the relation between learner errors and POS tagging or parsing errors. These annotations are used to evaluate the performance of multiple standard POS taggers and dependency parsers on learner data, and develop an SCF identification system for learner language.

Section 3.1 introduces our learner English data, the EF-Cambridge Open Language Database (EFCAMDAT, Geertzen et al., 2013). In section 3.2, we first introduce the native English POS and constituency data, the Penn Treebank of Wall Street Journal (PTB-WSJ, Marcus et al., 1993). We then introduce our native English SCF data (Quochi et al., 2014) which was constructed with sentences sampled from the British National Corpus (BNC, Aston and Burnard, 1998).

3.1 Learner English Data

We used EFCAMDAT [1] (Geertzen et al., 2013) as our learner data. We chose EFCAMDAT because it had data from learners that covered a wide range of proficiency levels and nationalities (see details below). This allowed for the extraction of the learner data samples that were representative of the full proficiency spectrum and diverse L1 background. The sampled learner data, after annotated, provided a comprehensive gold standard for evaluating the performance of standard POS taggers and dependency parsers on learner English (Chapter

[1] http://corpus. mml.cam.ac.uk/efcamdat2

Table 3.1 Alignment between EFCAMDAT proficiency and common standards

Englishtown	1-3[a]	4-6	7-9	10-12	13-15	16
Cambridge Main Suite Exams	-	KET	PET	FCE	CAE	-
IELTS	-	<3	4-5	5-6	6-7	>7
TOEFL iBT	-	-	57-86	87-109	110-120	-
TOEIC Listening & Reading	120-220	225-545	550-780	785-940	945	-
TOEIC Speaking & Writing	40-70	80-110	120-140	150-190	200	-
CEFR	A1	A2	B1	B2	C1	C2

[a] The tasks of each proficiency level were aligned with the can-do statements for the relevant CEFR level, as well as vocabulary and grammatical structures that were appropriate for the level. These alignments were based on the CEFR documentation of the Council of Europe, criterial feature research, and the experience of the content developers.

4), and for training our SCF identification system (Chapter 5). Furthermore, the large size of EFCAMDAT allowed for statistical analysis of learner English across various proficiency levels and writing tasks. This was useful for our quantitative investigation into how SCF diversity changed with L2 proficiency, a linguistic research we conducted to demonstrate the usefulness of the SCF identification system (Chapter 6).

EFCAMDAT was built at Cambridge through a collaboration between Linguistics and EF Education First, an international school of English. EFCAMDAT contained writings submitted to *Englishtown*, the online school of EF Education First. It had more than 47 million words written by nearly 109,000 learners. The writings covered 128 different topics and a variety of writing types such as narrative (e.g., "writing a movie plot") or descriptive (e.g., "describing your house"). The writings spanned across 16 proficiency levels covering the whole spectrum of A1-C2 in the Common European Framework of Reference for Languages (CEFR, Council of Europe, 2001). Table 3.1 shows the alignment between the proficiency levels of EFCAMDAT and common standards, which include Cambridge Main Suite Exams ranging across Key English Test (KET), Preliminary English Test (PET), First certificate in English (FCE) and Certificate of Advanced English (CAE), the International English Language Testing System (IELTS), the Test of English as a Foreign Language (TOEFL), and the Test of English for International Communication (TOEIC). Each writing task had expected length, ranging from 20-40 words for Level 1 to 150-180 words for Level 16. Figure 3.1 illustrates learner texts at the two ends of the proficiency spectrum. Because of the global reach of EF, there was considerable diversity in learner background, with Brazilian being the most dominant group (35% of the writings), followed by Chinese (21%), Mexican (7%), Russian (7%), German (5%), French(4%) and Italian (4%). Table 3.2 shows more details about the distribution of texts and words for the top ten nationalities of EFCAMDAT.

1. LEARNER 40663, LEVEL 1, UNIT 1, MEXICAN

```
Hello Anna, Great!!!  thank you for ask me .  Today I received a
good news in my work.  How are you ?  But first my name's Luis.
I'm 29, I'm from Mexico City and I like go out to dance and also
go to the movie.  What do you like to do ?  See you !!
```

2. LEARNER 39433, LEVEL 16, UNIT 8, GERMAN

```
The creature that is wideley called 'Bigfoot' has been described
by many researchers as a large ape-like creature with much hair
growing from his body.  It seems as if it measures between 2 and 3
metres in height, and weighs at least 200 to 250 kg and is covered
with dark brown, sometimes, dark reddish hair.  Witnesses have
also been describing the creature as having large, deep-sitting
eyes, a pronounced browridge, and a large, low-set forehead.  The
top of its head was pitctured by several witnesses as crested
and rounded.  The creature is often reported to have a stinging,
unfavourable smell by those who claim to have sighted and,
of course in only a few cases, encountered it.  Its enormous
footprint from which it got its name measures as large as 60 cm
long and 20 cm wide.  Altough most stamps have five toes - similar
to all known apes - some have shown footprints with toes ranging
from two to six.  Some included also claws as they are known from
bears.
```

Fig. 3.1 Two texts from EFCAMDAT

Table 3.2 Distribution of words and texts for the top ten nationalities in EFCAMDAT

Nationality	% text	# text	# word
Brazilians	35.3	232,265	15,400,784
Chinese	20.8	136,917	9,931,905
Mexicans	7.1	46,794	3,095,187
Russians	6.9	45,229	3,635,775
Germans	4.8	31,788	2,820,837
French	3.6	23,938	1,901,787
Italians	3.6	23,609	1,996,150
Saudi Arabians	2.9	19,282	1,152,950
Taiwanese	2.2	14,664	1,152,751
Japanese	1.7	11,177	829,566

For the annotation of POS tags, dependency structure, SCFs, learner errors, and the relation between learner errors and POS tagging or parsing errors, we adopted a subset used by Geertzen et al. (2013) which contained 1,000 sentences (11,067-word tokens) from EFCAMDAT. The dataset was extracted by first automatically segmenting EFCAMDAT into sentences, and then pseudo-randomly sampling the sentences with equal representation from all 16 proficiency levels and five of the best-represented nationalities (i.e., Chinese, Russian, Brazilian, German, and Italian). Nevertheless, some sentences in the original dataset contained segmentation errors. To prevent these segmentation errors from introducing artificial learner errors into the sentences, we manually corrected the segmentation of these sentences. Moreover, some sentences suffered from comma splice, i.e., the phenomenon of using a comma to connect two independent clauses such as *The ship was not new, it was not a cruise ship.* In the original dataset, such sentences were truncated at the commas. However, comma splice was a prominent learner error that required attention. To investigate how comma splice affected parsing accuracy, we manually changed the segmentation of these sentences, following the legitimate end of a sentence such as a full stop. In sum, 68 sentences were changed, which led to an increase of the word tokens to 12,003. We hereafter refer to the dataset as EF1000.

We syntactically annotated EF1000 for the evaluation of standard POS taggers and dependency parsers on learner English (Chapter 4), and for the development and evaluation of the SCF identification system (Chapter 5). We use the whole EFCAMDAT for the investigation of the relation between SCF diversity and L2 proficiency (Chapter 6).

3.2 Native English Data

We used two native English datasets: the part of PTB which included the annotation of POS tags and constituency structure for the articles from Wall Street Journal, and a general-domain SCF corpus which included the annotation of SCFs for the sentences sampled from BNC.

3.2.1 Penn Treebank

We used the WSJ part of PTB (Marcus et al., 1993, PTB-WSJ) as our native English dataset in the evaluation of POS taggers and dependency parsers, because PTB-WSJ has been widely used in the field of NLP to train standard English parsers. PTB-WSJ had one million words annotated with POS tags and constituency structure. The constituency structure was presented in a bracketed format as in Fig 3.2. The POS tagset had 36 tags for words, and 12

```
( (S
    (NP-SBJ (NNP Mr.) (NNP Vinken) )
    (VP (VBZ is)
      (NP-PRD
        (NP (NN chairman) )
        (PP (IN of)
          (NP
            (NP (NNP Elsevier) (NNP N.V.) )
            (, ,)
            (NP (DT the) (NNP Dutch) (VBG publishing) (NN group) )))))
    (. .) ))
```

Fig. 3.2 Bracketed format of constituency structure in Penn Treebank

tags for currency symbols and punctuation. The constituency tagset had 14 tags for phrases and clauses.

A number of tools were available to convert the constituency structure into dependency structure (e.g., De Marneffe and Manning, 2008b; Johansson and Nugues, 2007; Yamada and Matsumoto, 2003). We followed the conventional segmentation of PTB-WSJ for training and testing the parsers (Andor et al., 2016): we used Sections 2-21 of PTB-WSJ as the native English training data for some parsers (see Chapter 4 for more details), and Section 23 for parser evaluation.

3.2.2 SCF corpus

We used a general-domain native English SCF corpus for the development of the SCF identification system. The purpose was to examine whether including the native English SCF corpus as part of the training data can help to increase the accuracy of SCF identification (Chapter 5). Even though native English was different from learner English, they still resembled each other in some way. From the machine learning perspective, the increase of training data may increase the coverage of the model features for unseen data.

The SCF corpus contained 6,133 sentences (186,534 word tokens) sampled from BNC for 24 verb lemmas. The verb lemmas were chosen by a linguist to represent a wide range of subcategorization behaviors. In each sentence, only one verb was annotated for SCF. The SCFs were annotated by the linguist following the same guidelines and procedure in producing two published domain-specific SCF corpora (Quochi et al., 2014).

The SCF corpus was originally annotated with the SCF inventory of Preiss et al. (2007) which contained 168 SCF types. We mapped the fine-grained SCFs to a coarse-grained one defined on the popular Stanford typed dependencies (De Marneffe and Manning, 2008a). We conducted the mapping for three reasons: first, the distribution of SCFs is Zipfian (Korhonen, 2002), and many fine-grained SCFs rarely appear in real-world data. Second, learners tend

to use simple SCFs, and a coarse-grained inventory provided appropriate granularity for analyzing the SCFs of learner language. Third, a coarse-grained SCF inventory also provided a suitable level of specificity for many downstream NLP tasks (Van de Cruys et al., 2012).

Our mapping basically abstracted over some lexicalized adverbs and prepositions in the original frames. For example, the fine-grained inventory regarded the SCF of *bought* in *She bought a book for him* as different from the SCF of *sent* in *I sent him as a messenger* ("V-O-for" vs. "V-O-as"), while the course-grained inventory regarded both SCFs as the same ("V-O-P"). The course-grained inventory also merged cases that involved formal subjects or clausal subjects with cases that involved nominal subjects. For example, the SCF of *seems* in *It seems that they left* was considered the same as the SCF of *complained* in *He complained that they were coming*, and the SCF of *annoys* in *That she left annoys them* was considered as the same in *This annoys them*. The course-grained inventory also ignored differences in control. For example, the difference between the object control of *sent* in *I sent him as a messenger* and the subject control of *serves* in *She serves the firm as a researcher* was ignored, and the SCFs in both cases were considered as "V-O-P". Our final SCF inventory contained 70 SCF types (See Appendix B).

In the rest of the present thesis, we use the terminology of our SCF inventory to refer to SCFs. The SCF types were named by the complements involved. For example, "dobj_N" (direct object) was used to refer to the SCF of "V-O" mentioned in the previous chapters. Multiple complements were joined by colons. For example, "dobj_N:pobj" had two complements: "dobj_N" and "pobj" (prepositional object). Meanwhile, "_" denoted the broad POS tag of the head word of the complement or whether the complement was introduced by a wh-word. For example, "ccomp_VTENSED" meant that the head word of the clausal complement was a finite verb. Moreover, "=>" denoted the dependent of a complement, and the dependent may be lexicalized and denoted by "-". For example, "ccomp_VTENSED=>mark-that" (e.g., the SCF of *indicated* in *It indicated that he left*) meant that the clausal complement had a dependent of marker (a word that introduced a subordinate finite clause), and the marker was lexicalized by "that".

The native English SCF corpus had 43 types of SCFs, indicated by "N" in Appendix B. Table 3.3 shows the distribution of the ten most frequent SCFs in the corpus, illustrating each SCF type with an example from the corpus. The first column of Table 3.3 denotes the number of an SCF in Appendix B, where the guideline examples of the SCFs are available.

Table 3.3 Distribution of the ten most frequent SCFs in the native English SCF corpus

#[a]	SCF	%	Example (native English SCF corpus)
23	dobj_N	40.4	Then we can **help** [each other].
65	su	10.0	Aunt Violet tried to **help**.
32	dobj_N:pobj	8.9	You will **help** [me] [with my English].
68	xcomp_VBARE=>aux_TO	6.0	It did **help** [to convince him].
50	pobj	5.9	... to **help** [with energy conservation].
42	dobj_N:xcomp_VBARE=>aux_TO	5.7	**Helping** [the patient] [to move around].
9	ccomp_VTENSED	3.8	["He'll be fine,"] she **said**.
40	dobj_N:xcomp_N	2.5	We **call** [this] [a calendar year].
1	acomp	2.2	To me, she just **seemed** [kind].
24	dobj_N:iobj	2.2	She **gave** [him] [an enquiring glance].

[a] # denotes the number of an SCF in Appendix B.

Chapter 4

Evaluation of POS taggers and parsers

In this chapter, we evaluate how standard POS taggers and parsers perform on learner English. Since some parsers include a POS tagging component and we evaluate only such POS taggers, we regard POS tagging as an integrated part of parsing. In other words, we use parsers to refer to automatic syntactic analysis systems that produce syntactic relations and possibly POS tags. According to the research gaps pointed out in Chapter 2.3.2, we guide our evaluation with the following research questions:

1. What is the accuracy of different standard parsers on learner English?
2. Is there annotation bias in the gold standard created by manually correcting the output of a single parser? If there is, how does the annotation bias influence the accuracy scores?
3. What is the impact of fine-grained learner errors on parsing?
4. What is the relation between the performance of standard parsers on learner English and native English?

We conduct three evaluations. Firstly, we evaluate the accuracy of multiple parsers on learner English. During this evaluation, we also investigate the potential of annotation bias and its impact on the evaluation results. In the second part, we evaluate the effect of learner errors on parsing. Finally, we compare the accuracy scores of the parsers on learner English and native English, and examine the correlation between the two sets of scores. The following section describes our manual annotation of the learner dataset for the evaluations. We then report the evaluation results and summarize our findings.

4.1 Manual annotation of learner English

We manually annotated the learner dataset (EF1000) for POS tags, dependencies, learner errors, as well as the relations between learner errors and parsing errors. We introduce the annotators below, and describe the annotation of each aspect in the following sections.

Two researchers participated in the annotation of POS tags, dependencies and learner errors. They independently annotated 30 sentences for training, and 200 sentences for calculating inter-annotator agreement. It turned out that their inter-annotator agreement on both annotation tasks was sufficiently high (see Chapters 4.1.1 and 4.1.2), which meant that the two annotators were consistent and the annotation was reliable. As a result, only one annotator continued to annotate the rest of the learner dataset, i.e., the remaining 770 learner sentences. This annotator also annotated the relations between learner errors and the parsing errors of a single parser (Chapter 4.1.3).

4.1.1 POS tags and dependencies

For the purpose of investigating annotation bias, we annotated two versions of POS tags and dependencies. Firstly, the dependency structure of EF1000 was annotated by manually correcting the output of a single parser. We refer to this parser as the pre-annotation parser, and the manual annotation as the single-parser-based (SPB) annotation throughout the rest of this paper. Secondly, the SPB annotation was compared to the output of several other parsers and, where differences existed, the SPB annotation was reviewed (see details below). The reviewed annotation is hereafter referred to as the multiple-parser-based (MPB) annotation. We then evaluated the parsers on both annotations. We considered the MPB annotation to represent the accurate annotation of the learner data, whilst the comparison of accuracy scores on the MPB and SPB annotations showed whether annotation bias existed and influenced the parser evaluation.

POS and dependency schemes

We used Penn Treebank POS tagset (Marcus et al., 1993) and Stanford typed dependencies (SD) (De Marneffe and Manning, 2008a), the most widely-used dependency scheme for English in the field of computational linguistics. Having evolved over time, SD was not only mature in describing well-formed language, but also flexible in describing language errors. SD included dependency relations for loose structures (e.g., “parataxis”, “discourse”) and words that were erroneously separated (“goes-with”). These relations were useful for

describing learner errors. For example, when *furthermore* was misspelled as *further more*, *more* can be annotated as being a dependent of *further* in the relation of"goes-with".

Our SD scheme varied slightly from that of Geertzen et al. (2013). Firstly, their scheme was older and did not include the dependency relations of "discourse" and "goes-with". There were also other minor changes which made our version more concise. For example, "vmod" was introduced to generalize over non-finite verbal modifiers that were participial (formerly "partmod") or infinitival (formerly "infmod") (see De Marneffe and Manning, 2008b, for details). Secondly, Geertzen et al. (2013) used the default setting of SD, which treated copulas as the dependents of their complements. This caused inconsistency in representing the dependency relations between verbs and their complements (e.g., for the similar sentences *They look like flowers* and *They are flowers*, *flowers* was regarded as a complement in the first sentence but the root in the second sentence). Contrastingly, we treated copulas as the heads of their complements (i.e., *flowers* was still a complement in *They are flowers*).

Parsers

Since rule-based parsers required extensive human effort to define rules and their parsing schemes were difficult to change, our evaluation focused on probabilistic parsers. A probabilistic parser computed the most likely parse of a sentence according to a statistical syntactic model which associated syntactic rules with probabilities. The statistical model was trained on a corpus of POS tags and syntactic relations. As such, the probabilistic parsers can be tailored to the scheme of the training corpus.

As mentioned in Section 2.3.2, there were two approaches to obtaining SD automatically: the c-parsing approach (Kong and Smith, 2014) converted the output of a constituency parser to dependency relations by definitive rules (De Marneffe et al., 2006), while the d-parsing approach extracted the dependency relations directly. We chose three constituency parsers for c-parsing and two dependency parsers for d-parsing. These parsers were well-known and had been frequently used in NLP (Cer et al., 2010; Kong and Smith, 2014). Moreover, we tested two different settings for each of the two constituency parsers. As a result, seven different parsing settings were tested in total. Meanwhile, each constituency parser included a POS tagger as a preprocessing component, so we tested five different POS tagging settings.

The constituency parsers were as follows:

- **Stanford parser** (Version 3.5.1): We tested two ready-made syntactic models, both of which had been trained on a number of treebanks [1] in addition to PTB-WSJ Sections 2-21. The first syntactic model (hereafter referred to as SU) followed a probabilistic

[1]The training data for the Stanford parsers are listed in http://nlp.stanford.edu/software/parser-faq.shtml.

context-free grammar (PCFG) (Klein and Manning, 2003a), whilst the second model (SL) followed a lexicalized PCFG which integrated head words into the syntactic rules (Klein and Manning, 2003b). Since Geertzen et al. (2013) showed that the SU parser setting achieved high accuracy on learner data, we selected SU as the pre-annotation parser for the construction of the gold standard, and provided the POS tags produced by SU to dependency parsers which required POS tag input;

- **BLLIP parser** (The latest version retrieved from the official repository on March 25, 2015) (Charniak and Johnson, 2005): We tested two ready-made syntactic models trained on different datasets. The first one (hereafter referred to as BS) on OntoNotes-WSJ and the Google Web Treebank; the second one (BW) on PTB-WSJ and about two million sentences from Gigaword.

- **Berkeley parser** (Version 1.7) (Petrov and Klein, 2007) (BK): We used a ready-made syntactic model called "eng_sm6", which had been trained on PTB-WSJ Sections 2-21.

We used the Stanford typed dependency converter (version 3.5.1) (De Marneffe and Manning, 2008a) to convert the constituency structures produced by the aforementioned parsers to collapsed SDs. The converter required the constituency structures in the Penn Treebank (PTB) format. Since the POS tags of auxiliary verbs ("AUX") in the constituency output of BLLIP parser differed from the PTB format, we replaced these POS tags with their counterparts produced by the pre-annotation parser SU.

For d-parsing, we used **Turbo parser** version 2.1.0 (Martins et al., 2013) (TB) and **MaltParser** version 1.8 (Nivre et al., 2007) (MT). We converted Sections 2-21 of PTB-WSJ to the basic SD format, and trained both dependency parsers with default settings on the dataset. When training the MaltParser, we followed the feature template used in the ready-made "engmalt" model. Since these dependency parsers contained no POS taggers, we provided the POS output of the pre-annotation parser SU to these parsers during the evaluation. The original outputs of these dependency parsers followed the basic SD format. We converted them to collapsed SDs using the converter (De Marneffe and Manning, 2008a) again.

Annotation procedure

The training process for POS and dependency annotation was as follows. First, the annotators learned the PTB annotation guideline for POS tagging (Santorini, 1990) and the Stanford typed dependencies manual (De Marneffe and Manning, 2008a) for dependency parsing.

They then independently annotated 30 sentences randomly selected from the learner dataset. During the annotation, the annotators had access to the converted dependency relations from PTB-WSJ. If uncertain about how to annotate a word, they could search the word in the database to find out how it was usually annotated. The two annotators then discussed and resolved their annotation disagreement on the 30 sentences.

Two annotations were produced for each sentence. First, the annotators corrected the output of the pre-annotation parser SU to generate a single-parser-based (SPB) annotation. During this annotation, the annotators had access to the gold standard of Geertzen et al. (2013) for reference. Despite some aforementioned differences in the sentences (Section 3.1) and the annotation schemes (Section 4.1.1), the gold standard of Geertzen et al. (2013) provided additional human annotation information that may help to improve the annotation accuracy. The annotators could also check the context of the sentence, i.e., the learner essay that contained the sentence.

After completing the SPB annotation, the annotators generated a multiple-parser-based (MPB) annotation by reviewing the SPB annotation according to alternative annotations provided by the other parsers. Specifically, we extracted the words where the outputs of at least one of the other six parser settings disagreed with the pre-annotation parser SU (hereafter referred to as annotation mismatches), and displayed all the disagreements as well as the SPB annotation to the annotators. The annotators then re-annotated these cases. When an annotation (i.e., POS tag, head index or dependency label) provided by one of the six parser settings was correct and that of the SPB annotation was incorrect, the correct annotation was marked with "C" (correction). When an annotation of one of the six parser settings was different from that of the SPB annotation but both annotations were acceptable, the annotation provided by the parser setting was marked with "M" (multiple options). Furthermore, if both the annotation of SPB and those of the other parsers were incorrect, the annotation of SPB was corrected and marked with "N" (non-replacement correction). We then generated the MPB annotation by substituting the annotations marked with "C" and "N" for their counterparts in the SPB annotation, and including the alternative annotations marked with "M".

To illustrate the MPB annotation procedure, consider Figure 4.1: the columns from left to right correspond to word indices, words, POS tags, head indices, dependency labels, marks for the aforementioned three annotation types (# by default), and parser settings. As we can see, in this example SL annotated the head index as 8 and the dependency label as "rcmod" (relative clause), while the SPB annotation annotates the head index as 2 and the dependency label as "advcl" (adverbial clause). The annotator decided that both annotations of the head index were incorrect and that the correct head index was 4. He, therefore, marked the head

Id	Word	POS	Head Id	Dependency	POS	Head Id	Dependency	Annotation
12	was	VBD	2	advcl	#	N(4)	#	SPB
12	was	VBD	8	rcmod	#	#	C	SL

Fig. 4.1 Format of the re-annotation based on annotation mismatches

index of SPB with "N" and provided the correct head index in parentheses "(4)". By contrast, the annotation of SL on the dependency label was correct while that of SPB was not, so the annotator marked the dependency label of SL with "C".

As mentioned earlier, the two annotators annotated another 200 sentences after the training. We measured the inter-annotator agreement on the MPB annotations. According to the kappa metric, the inter-annotator agreement on the annotation of POS tags, head indices, and dependency labels was 0.961, which was similar to the inter-annotator agreement achieved by Geertzen et al. (2013) (0.971). Alternatively, according to the conventional parsing evaluation metrics, our inter-annotator agreements were 97.03% on POS accuracy, 94.46% on UAS, and 91.69% on LAS, which were close to those achieved by Ragheb and Dickinson (2013) (around 99% on POS accuracy, 97% on UAS and 95% on LAS; note that their scores were not directly comparable to ours due to differences in the annotation schemes). These results showed that the inter-annotator agreement between our annotators was sufficiently high. One annotator then completed the remaining 770 sentences of the annotation.

4.1.2 Learner errors

To investigate the impact of learner errors on parsing, we first annotated the learner errors in EF1000. The following section describes the learner error annotation scheme. We then present the annotation procedure.

Learner error scheme

The annotation of any learner error required the assumption of a correct form, i.e., the target hypothesis (Ellis, 1994). Since different target hypotheses may be assumed for the same learner error, it was difficult to achieve consistent annotation (Reznicek et al., 2013). Nevertheless, there were several ways to improve the consistency of learner error annotation: firstly, using a learner error scheme with predefined learner error types (Fitzpatrick and Seegmiller, 2004); secondly, designing the taxonomy of learner errors properly; thirdly, setting rules for typical ambiguous cases and training the annotators with these rules.

We used the learner error scheme of the Cambridge Learner Corpus (CLC-FCE) (Nicholls, 2003). The scheme included over 80 learner error types. The majority of the learner errors

were defined along two dimensions: the deviation of the learner error from the target hypothesis and the syntactic category of the target hypothesis word. For example, the learner error "MV" represented a missing (M) verb (V). These two dimensions were most descriptive for learner errors (James, 2013); combining them helped to achieve a fine-grained annotation scheme that allowed for consistent annotation of learner errors. The scheme had been applied to CLC-FCE, which provided a large number of consistent annotation examples[2]. As a result, CLC-FCE can be used as a reference to resolve ambiguous cases during learner error annotation.

Nevertheless, in addition to the original taxonomy, we added two learner error types: "C" (Capitalization error) for capitalization errors, and "SP" (Space error) for wrongly split or concatenated words. In CLC-FCE, these two types of learner errors were somewhat inappropriately annotated as "RP" (punctuation needs replacement) (see Appendix A for the full taxonomy of learner errors used in this study).

The annotation of learner errors used the format of XML markup illustrated as follows:

I <ns type="TV">graduate<c>graduated</c></ns> in 1983 .

where the erroneous sentence segment *graduate* was marked by , while the target hypothesis *graduated* was marked by <c>; the learner error type was indicated by <ns type="TV">, which means wrong verb tense.

Annotation procedure

The annotation of a learner error involved three steps: identifying a sentence segment that contained a learner error, marking the learner error type, and providing a correction. The annotators went through a training before the annotation. First, the two annotators learned the CLC-FCE learner error taxonomy (Nicholls, 2003). They then independently annotated the 30 training sentences, during which they had access to the learner error annotation of CLC-FCE for reference. The two annotators then discussed and resolved their annotation disagreement on the 30 sentences.

After the training, the two annotators annotated 200 sentences. We calculated two types of inter-annotator agreements which have been used in previous research on learner error annotation (Fitzpatrick and Seegmiller, 2004; Rosen et al., 2014). The first one was a global measure based on the percentage of overlapping learner errors in both annotations (Fitzpatrick and Seegmiller, 2004). The formula was as follows:

[2]While there was no proof of the consistency of the learner error annotation of CLC-FCE in terms of reliability metrics, the experience of our annotators and the fact that CLC-FCE had been widely used in automatic error correction where consistent training examples were required suggested that the annotation of CLC-FCE was consistent.

$$\frac{n_{overlapping}}{\frac{n_1+n_2}{2}} \tag{4.1}$$

where $n_{overlapping}$ referred to the number of overlapping learner errors in the two annotations; n_1 and n_2 denoted the number of learner errors in the two annotations respectively.

As mentioned before, the goal of checking the inter-rater agreement was to find out whether the annotators were reliable in identifying learner errors so that we can entrust the rest of the annotation to only one annotator. Since there might be multiple ways to correct an identified learner error, which was unavoidable and should be allowed, we did not require the corrections to be the same for the overlapping learner errors. We first defined the overlapping learner errors as the learner errors annotated with the same beginning word and the same learner error type. This definition allowed for automatic calculation. We also proposed a less stringent definition of the overlapping learner errors: the learner errors that targeted a similar range of text even though the actual annotation might be different. For example, for the sentence segment *we are in* the necessary step*, one annotator corrected it as *we are* [*at*] *the necessary stage*, while the other one corrected it as *we are* [*taking a*] *necessary step*. Even though the annotations of learner error types were different, they were considered to be overlapping under the less stringent definition because such variances were caused by the inherent ambiguity of the target hypothesis rather than oversight or inability to detect a learner error. This definition of overlapping learner errors required manual calculation.

The inter-annotator agreement was 70.9% under the stringent definition of the overlapping learner errors, and 86.3% under the less stringent one. Fitzpatrick and Seegmiller (2004) achieved a much lower inter-annotator agreement at 60%. This was mainly due to the absence of predefined learner error types in their annotation scheme. In other words, they required the correction to be the same when calculating the overlapping learner errors. When we changed the stringent definition of overlapping learner errors as having the same correction rather than learner error type, the inter-annotator agreement was 63.5%. The higher figure demonstrated the improvement in consistency presumably brought by our annotation scheme and the application of CLC-FCE as a reference during the annotation.

We also analyzed the inter-annotator agreement of specific errors in terms of the kappa metric (Rosen et al., 2014). Table 4.1 shows the kappa inter-annotator agreement of the learner errors that appeared at least three times on average between the annotations of both annotators.

Table 4.1 indicates that most learner errors were annotated consistently, especially the spelling error (“S”), capitalization error (“C”), missing a determiner (“MD”), wrong form of

Table 4.1 Kappa inter-annotator agreement of learner errors

Learner error	Kappa	Avg. # tags
S	0.897	44
C	0.877	21
MD	0.841	19
MT	0.787	17
MP	0.665	15
RP	0.623	15
RT	0.614	12
RD	0.699	10
AGV	1.000	10
UD	0.699	10
FN	0.823	9
MV	0.624	8
SP	0.705	7
FV	0.615	7
AS	-0.003	7
M	0.152	6
RA	0.909	6
MC	0.909	6
AGN	1.000	5
W	0.213	5
UT	0.889	5
RV	0.666	5
RJ	0.666	5
CE	-0.001	4
DJ	0.750	4
DN	0.571	4
RN	0.333	3
R	0.000	3

a noun ("FN"), a pronoun needed replacing ("RA"), missing a conjunction ("MC") and an unnecessary preposition ("UT") ($\kappa > 0.8$).

However, learner errors of incorrect argument structure ("AS"), something missing ("M"), incorrect word order ("W"), something needed replacing ("R"), a noun needed replacing ("RN") and complex error ("CE"), were not consistent between the two annotators ($\kappa < 0.4$). Further analysis shows that these errors were subject to more varied target forms, and were therefore not easy to annotate in the same way among different annotators. The finding was similar to that of Rosen et al. (2014) on the annotation of learner Czech: they found that learner errors like incorrect morphology, whose target forms were easy to establish, can be annotated consistently, whereas learner errors like incorrect complex verb forms or wrong lexis cannot be annotated consistently due to varied target forms.

In general, the inter-annotator agreement was high, which showed that the two annotators were reliable in identifying learner errors. As a result, one annotator continued to annotate the rest of the learner dataset for learner errors. This annotator also annotated the effect of learner errors on POS tagging and parsing errors of all learner sentences (see the following section).

4.1.3 Relations between learner errors and parsing errors

To investigate the impact of learner errors on parsing, we then annotated the relations between learner errors and parsing errors. We operationally defined that a learner error caused a parsing error if the removal of the learner error led to the disappearance of the parsing error. Since learner errors may jointly affect parsing (i.e., some parsing errors may be caused by the co-occurrence of two or more learner errors), it was important to annotate the effect of both individual and the combinations of learner errors. However, the number of learner error combinations increased exponentially with the number of learner errors in a sentence. For example, a sentence that contained five learner errors had $2^5 - 1$ (i.e., 31) combinations of learner errors. Observing whether the correction of these combinations led to the disappearance of a parsing error was time-consuming. To limit the scale of our problem, we evaluated the effect of learner errors only on the pre-annotation parser SU.

The annotation procedure was as follows. Since we needed to annotate the relations between learner errors and parsing errors, we first extracted the learner sentences that contained both learner errors and parsing errors (344 sentences). Secondly, we corrected various combinations of the learner errors to produce partly or totally corrected sentences. Thirdly, we parsed the corrected sentences with SU. Fourthly, we extracted parsing errors by contrasting the parsing output to the MPB annotation. The annotator then annotated the effect of the learner errors in the following way: for any parsing error, if the correction of a learner

error combination resulted in the disappearance of the specific parsing error in the corrected sentence, we annotated the parsing error as related to all the learner errors in that combination. Only the minimum combination of learner errors was annotated; any other learner error combinations which included these learner errors and caused the disappearance of the same parsing error were not annotated. For the sentences that contained less than 6 learner errors (332 sentences), the annotators examined all their corrected sentences, i.e., checked the effect of each learner error combination on parsing. The rest of 12 sentences contained too many learner errors; examining all corrected sentences was impractical. Nevertheless, the annotation on the 332 sentences showed that most learner error combinations that affected parsing errors involved fewer than four learner errors. Therefore, the annotator examined only the correction of fewer than four learner errors for the 12 sentences.

4.2 Evaluation

We evaluated the performance of the parsers on the annotated learner English dataset (E-F1000) and the gold standard dataset of POS tags and dependency structure of native English (PTB-WSJ). We report our investigation in the annotation bias and the accuracy of multiple parsers on learner English in the following section. We then report the impact of learner errors on the performance of the baseline parser. Finally, we report the comparison between the performance of the parsers on learner English and native English.

4.2.1 Annotation bias on learner English

Firstly, we evaluated the parsers against the SPB annotation. Table 4.2 shows the accuracy scores of the parsers. The accuracy was measured by the proportions of the words that received correct POS tags (POS), unlabeled attachments (UAS), labeled attachments (LAS), and the combination of POS tags and labeled attachments (All), as well as the proportions of the sentences that were free of the errors in each of the aforementioned aspects. The parsers in the d-parsing category had no POS accuracy scores because they did not include a POS tagger. It turned out that the pre-annotation parser performed the best on all criteria, even though the word-based accuracy scores, except on POS tags, were slightly lower than the results of Geertzen et al. (2013), in which the corresponding figures of the word-based POS, UAS, LAS and All were 96.1%, 92.1%, 89.6% and 88.6% respectively. The maximum performance gaps between the parsers were smaller on POS tags than on dependency relations (see the "Max. Diff." row of Table 4.2).

Table 4.2 Accuracy of the parsers on the SPB annotation

Parsing approach	Parser[a]	Accuracy by word (%)				Accuracy by sentence (%)			
		POS	UAS	LAS	All	POS	UAS	LAS	All
c-parsing	SU	96.31	91.49	88.03	87.1	72.0	59.70	49.8	46.7
	SL	95.25	89.25	85.06	83.66	62.3	50.3	40.5	36.5
	BS	94.95	90.53	86.88	84.88	59.4	55.0	43.5	35.9
	BW	95.00	90.64	86.96	85.10	59.7	56.3	45.2	37.8
	BK	94.81	90.26	86.36	84.40	61.2	54.9	43.3	36.9
d-parsing	TB	–	89.88	86.32	–	–	54.1	43.0	–
	MT	–	88.38	84.67	–	–	48.4	38.5	–
Max. Diff.		1.50	3.11	3.36	3.46	12.6	11.3	11.3	10.8

[a] SU: Stanford PCFG (unlexicalized) parser; SL: Stanford lexicalized parser; BS: BLLIP parser trained on OntoNotes-WSJ and Google Web Treebank; BW: BLLIP parser trained on PTB-WSJ and Gigaword; BK: Berkeley parser; TB: Turbo parser; MT: MaltParser.

The coincidence that the pre-annotation parser performed the best on the SPB annotation seemed to suggest the presence of an annotation bias in the SPB annotation towards the pre-annotation parser. We then evaluated the parsers against the MPB annotation. The results (Table 4.3) confirmed the hypothesis about annotation bias. In this evaluation, the BLLIP parser turned out to be the best in all aspects except the sentence-based POS accuracy, on which the Berkeley parser performed the best. Specifically, BW, the parsing setting where the BLLIP parser was trained on Gigaword and PTB-WSJ, achieved the best results. On the other hand, the rank of the pre-annotation parser SU dropped to the third on the accuracy of word-based POS, the fifth on word-based UAS and LAS, and even the sixth on sentence-based UAS and LAS. The changes in the accuracy scores of the pre-annotation parser and the word-based accuracy scores of the best-performing parser between the two evaluations were significant according to chi-squared tests. These differences demonstrated that the SPB annotation was indeed biased towards the pre-annotation parser. The bias changed the ranking of the parsers, affecting the accuracy scores of the pre-annotation parser and the best-performing parser most. Furthermore, the maximum performance gaps between the parsers diminished, especially on POS (from 1.50% to 0.40% on the word level, and from 12.6% to 2.1% on the sentence level). This meant that the annotation bias in the SPB annotation also artificially increased the performance gaps between the parsers. In fact, the performance of various parsers on POS tagging was similar.

To better understand the annotation bias, we quantitatively and qualitatively investigated the re-annotations that produced the MPB annotation. First, we identified the annotation mismatches with respect to each parser (i.e., cases where the annotation of the particular parser disagreed with the pre-annotation parser SU) and all parsers (i.e., cases where the

Table 4.3 Accuracy of the parsers on the MPB annotation

Parsing approach	Parser[a]	Accuracy by word (%)				Accuracy by sentence (%)			
		POS	UAS	LAS	All	POS	UAS	LAS	All
c-parsing	SU	95.41***[b]	89.77***	86.05***	84.67***	64.7***	52.6**	42.5**	37.9***
	SL	95.38	89.70	85.46	84.06	62.6	53.7	42.9	36.9
	BS	95.63*	91.43*	87.77*	86.09**	63.7*	59.6*	47.7*	39.5
	BW	95.64*	91.53*	87.84*	86.28**	63.6*	60.5*	48.4	40.8
	BK	95.24	90.65	86.76	85.03	64.7	56.3	44.6	37.8
d-parsing	TB	–	90.53*	86.77	–	–	57.2	44.3	–
	MT	–	88.85	85.06	–	–	51.7	41.1	–
Max. Diff.		0.40	2.68	2.78	2.22	2.1	8.8	7.3	3.9

[a] SU: Stanford PCFG (unlexicalized) parser; SL: Stanford lexicalized parser; BS: BLLIP parser trained on OntoNotes-WSJ and Google Web Treebank; BW: BLLIP parser trained on PTB-WSJ and Gigaword; BK: Berkeley parser; TB: Turbo parser; MT: MaltParser.

[b] The marks of significance (chi-squared tests): *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$.

annotation of at least one parser disagreed with SU), and further classified these cases into two groups: one where the SPB annotation agreed with the pre-annotation parser SU, and the other where the SPB annotation disagreed with SU. Table 4.4 shows the number of annotation mismatches with regard to each non-SU parser setting and the proportion of the cases that were marked with correction ("C") or multiple options ("M") due to the correct reference provided by that parser setting.

In Table 4.4, the correction rates on the cases where the SPB annotation agreed with SU were much higher than where they disagreed. In the former situation, around 20% of the annotation mismatches with respect to individual parsers on the POS tag and head index required corrections. The correction rate on the dependency label varied across different parser settings but also went beyond 10% in most cases. By contrast, the correction rates on the cases where the SPB annotation disagreed with SU dropped to less than 0.5% on POS tags, 1.1% on head indices and 0.6% on dependency labels for each parser setting. This contrast meant that during the SPB annotation, the precision of correcting parsing errors was high (i.e., when a parsing error was corrected, the correction was accurate), but the recall of parsing errors was relatively low (i.e., the annotator accepted some wrong parsing choices during the SPB annotation).

The results also indicate that displaying the different output of various parsers helped the annotator to detect annotation errors. The contrast provided more information for reference during annotation, and helped to promote awareness of annotation errors. Nevertheless, the quality of the reference provided by different parsers varied. Table 3 shows that for the cases where the SPB annotation disagreed with SU, the correction rates with respect to the BLLIP parser (BS or BW) were highest. As revealed earlier, the BLLIP parser was most accurate on the learner dataset. Therefore, the annotation of BLLIP parser provided the best reference.

Table 4.4 Analysis of the annotation mismatches

Cases	Parsing approach	Parser[b]	POS			Head			Dependency label		
			#	C	M	#	C	M	#	C	M
SPB annotation agreed with SU[a]	c-parsing	SL	252	19.8	5.2	668	15.1	5.4	556	6.8	4.7
		BS	362	22.7	6.4	584	20.9	8.4	424	13.7	7.8
		BW	352	22.7	6.8	544	22.4	8.1	409	13.4	8.3
		BK	332	18.4	6.6	581	17.0	5.0	450	11.6	4.2
	d-parsing	TB	–	–	–	530	20.8	7.7	383	12.0	5.5
		MT	–	–	–	709	14.1	5.2	525	9.0	2.1
	All parsers		647	16.4	6.3	1,644	11.7	5.5	1,351	7.3	4.3
SPB annotation disagreed with SU	c-parsing	SL	200	0.5	0.5	579	1.0	0.7	594	0.2	0.2
		BS	274	0.0	0.0	642	0.8	0.9	682	0.3	0.9
		BW	274	0.0	0.0	610	0.8	1.3	664	0.5	1.1
		BK	229	0.4	0.9	610	1.1	1.1	650	0.0	0.6
	d-parsing	TB	–	–	–	494	0.8	0.6	530	0.6	0.2
		MT	–	–	–	515	0.4	0.2	544	0.6	0.6
	All parsers		338	0.6	0.6	893	1.2	1.2	969	0.6	0.9

[a] SU: Stanford PCFG (unlexicalized) parser.
[b] SL: Stanford lexicalized parser; BS: BLLIP parser trained on OntoNotes-WSJ and Google Web Treebank; BW: BLLIP parser trained on PTB-WSJ and Gigaword; BK: Berkeley parser; TB: Turbo parser; MT: MaltParser.

Other parsers also contributed useful reference, even though at lower accuracy. Meanwhile, since there was overlap in the correct references provided by different parsers (e.g., two or more parsers provided the same correct pre-annotation which led to the correction of an SPB annotation), the correction rates with respect to all parsers (i.e., the proportion of the annotation mismatches where at least one parser was correct) were generally lower than the correction rates with respect to individual parsers. This indicated that as the number of parsers adopted in the contrast-based annotation grew, the marginal benefit of adding a parser diminished.

We split the re-annotations with regard to the types of annotation, i.e., POS tags, head indices, and dependency labels, and summarized the linguistic structures that were prone to annotation bias.

Annotation errors on POS tags

Table 4.5 lists the types of POS annotation errors that occurred more than four times. Throughout this paper, we use the format of "wrong tag - correct tag" to refer to an annotation error or parsing error. Apart from the annotation error of "VBD-VBN" (past tense verb - past participle verb), all other annotation errors involved POS tag pairs which were listed as "easily confused" in the PTB annotation guideline (Santorini, 1990). In other words, most

Table 4.5 Annotation errors on POS tags (named by "wrong tag-correct tag")

Error type	Freq.
VB-VBP	21
RP-IN	11
RP-RB	7
NNP-NN	7
VBN-JJ	6
RB-NN	6
RB-IN	5
IN-WDT	5
VBD-VBN	4
VBG-NN	4
JJ-NN	4
IN-RB	4

of the annotation bias with respect to POS tags was related to choices between inherently confusing POS tag pairs. Further qualitative analysis revealed some prominent causes of the confusions as follows.

- Overlap between inflectionally defined and functionally defined POS tags

  For example, this factor contributed to the annotation errors of "VBN-JJ" (past participle verb - adjective), "VBG-NN" (gerund or present participle verb - singular or mass noun) and "VBG-JJ". VBN and VBG were defined by verbal inflection, whereas JJ and NN were defined by the function of words in context. The two sets of POS tags were not mutually exclusive. For instance, in *joy of learning*, *learning* can be either a noun or a verb. This overlapped domain was caused by the fact that VBN and VBG were intermediate syntactic categories between prototypical verbs and prototypical adjectives or nouns, as we have discussed in the theoretical review of POS classification in Chapter 2.1.1.

- Overlap between POS tags in a containment relation

  For example, annotation errors involving RP (particle) and RB (adverb) or IN (preposition) were related to this factor. RP was a subclass of RB and bore some functional characteristics of IN as well. Basically, RB or IN seemed plausible for many cases where RP was annotated. The PTB annotation guideline (Santorini, 1990) defined rules and diagnostic tests for distinguishing between RP, RB and IN. However, these rules and tests may not apply to all cases. For instance, *two websites will be compared with** included the redundant word *with*. According to the PTB annotation guideline,

with should be tagged as IN. Nevertheless, the learner error here seemed to indicate that the learner used *with* as an RP.

- Same word forms with different POS tags

  An example of the annotation errors related to this factor was "JJ-NN". If the word forms of a mass noun NN and an adjective JJ were the same, confusion can occur when the words were used as prenominals or predicatives. For instance, in *plastic bottles* and *they are fun*, *plastic* and *fun* can be either NN or JJ.

  Annotation errors involving the same word forms with different POS tags can also arise with ambiguous structures. For example, VB (base-form verb) was not usually confused with NN. However, in *go to work*, *work* can be regarded as a VB, with *to* as an auxiliary; on the other hand, *work* can also be regarded as a NN, with *to* as a preposition.

Annotation errors on head indices

The annotation errors regarding head indices mostly occurred in the following linguistic structures:

- Prepositional phrases

  In (m), the prepositional phrase *with the head teacher* can be regarded as dependent on *build*; this was syntactically acceptable and semantically plausible (i.e., the head teacher was involved in building up the cooperation). However, from the context, we can see that the intended meaning of the construction was "to cooperate with the head teacher". Therefore, the prepositional phrase should be annotated as dependent on *cooperation* rather than *build*.

- Modifiers

  For a sequence of nouns where one noun headed the others, selecting which noun for the head can be challenging. For example, for a locational phrase in the form of "city, country" like *in Manus, Brazil*, it was plausible to analyze the city as a modifier that specified an area of the country. On the other hand, the comma between the two nouns can indicate post modification, in which the country was the modifier of the city.

- Coordinating conjuncts

  The SD scheme determined that in a coordination the first conjunct should be the head of all the other conjuncts. However, if a parser failed to identify the first conjunct,

attaching a conjunct to e.g., the second conjunct should not be regarded as a parsing error, because the conjunct relation was established. For instance, in (n), normally *raincoat*, *flash light*, *clothes*, and *sleeping bags* should be attached to *umbrella* with the dependency label of "conj_and". However, if a parser attached *flash light* to *raincoat* by "conj_and", the conjunct relation was also established and should not be regarded as a parsing error.

(m) I will build up a better cooperation with the head teacher which will ensure a better relation between those who make decisions and us students who are mainly affected by them.

(n) I need to take my umbrella, my raincoat, my flash light, my clothes and my sleeping bag.

Annotation errors on dependency labels

The annotation errors on dependency labels were usually related to the following linguistic structures:

- Linguistic structures subject to annotation errors on POS tags or head indices

  Since dependency relations were established based on POS tags, the errors in the latter may affect the former. For example, the dependency error of "amod-nn" (adjectival modifier - noun compound modifier) was related to the POS error of "JJ-NN". Similarly, "prt-prep" (phrasal verb particle - prepositional modifier) was related to "RP-IN", and "prt-advmod" (phrasal verb particle - adverb modifier) was related to "RP-RB". Furthermore, since the assignment of dependency labels correlated with the assignment of head indices during parsing, the errors in the latter may also affect the former. For example, the confusion of "nn-prep_in" (noun compound modifier - prepositional modifier with the preposition *in*) was caused by the attachment errors on noun modifiers as mentioned in the previous section.

- Prepositional phrases and infinitive clauses (adjunct vs. complement)

  Sometimes adjuncts and complements were difficult to disambiguate, which led to the annotation error of "vmod-xcomp" (reduced non-finite verbal modifier - open clausal complement). For instance, in (o) the purpose clause *to raise fund* can also be taken as a complement clause in the absence of subcategorization information for individual verbs.

- Conjuncts (multiple dependencies)

  The SD dependency scheme dictated that each word can have only one head. However, sometimes a word may be dependent on multiple words. This happened most frequently in conjunct structures. When a conjunct involved elliptical material, the dependents of the elided element may have multiple heads. For instance, in (p), *year* may be seen as either an object of the verb *have*, or a modifier of the word *warranty* at the end of the sentence. It was hard to choose between these two heads: choosing one dependency led to the loss of information for the other dependency. Of course, it is worth noting that the ellipsis in (p) was ungrammatical, which added to the annotation challenge.

Summarized above are the linguistic structures that were prone to annotation bias mostly because of the inherent ambiguity in the linguistic structures and the parsing scheme. Another major source of annotation bias was learner errors. Since there were various learner errors and the same learner error can cause different types of ambiguity in different contexts, the linguistic structures that were subject to annotation bias because of learner errors vary. For example, in *I hope this help* you*, the word *help* should be corrected as *helps* or *will help*. When we annotated the dependency relation between *help* and its head *hope*, two options seemed acceptable: if we assumed that the learner had incorrect knowledge of the subcategorization frame of hope, regarding *this* as the object of *hope* rather than the subject of *help* (i.e., confusing the frame of *hope* as that of *let* in *let somebody do something*), "xcomp" (open clausal complement) should be chosen; however, if we assumed that the learner used the right frame but made a mistake in the tense or number, "ccomp" (clausal complement) should be chosen. Furthermore, (q) shows a sentence that was unintelligible due to learner errors. There were many ways to interpret the sentence, each of which led to a different dependency structure. For example, if *demand* was intended to be *demanded*, *professional* should be annotated as "nsubjpass" (passive nominal subject). Alternatively, if *is demand* was intended to be *demands*, *professional* should be annotated as "nsubj" (nominal subject).

(o) I'd lead the student council to raise fund

(p) Our notebooks have a 1 year*, our pens two weeks warranty

(q) My professional is demand to one teach in the idiom English*.

In summary, this section showed the accuracy of standard parsers on the EF1000 learner data (Table 4.3). We also confirmed the existence of annotation bias in the SPB annotation setting, and identified the linguistic structures that were prone to annotation bias. The next section moves on to evaluate the effect of learner errors on parsing.

Table 4.6 Distribution of parsing errors caused by learner errors

Level	PE	# containing PEs	LE-caused PEs (%)
Sentence	General[a]	626	41.4
	POS	359	38.7
	Head index	478	40.2
	Dependency label	473	46.3
Word	General	1866	39.2
	POS	568	37.5
	Head index	1243	40.4
	Dependency label	1232	43.2

[a] In the category of "general", a word was counted as containing a parsing error if any annotation type of the word, i.e., the POS tag, the head index or the dependency label, was incorrect.

4.2.2 Impact of learner errors on parser performance

The reasonably high accuracy of the parsers on the learner data may create the impression that, after all, the learner errors did not have a significant impact on parser performance. It was, therefore, crucial to understand if the parsers were indeed robust to learner errors.

Firstly, we analyzed the overall effect of learner errors on parsing from two aspects: the proportion of the parsing errors that were caused by learner errors (hereafter referred to as "LE-caused PEs"), and the proportion of the learner errors that caused parsing errors. Secondly, we analyzed the effect of individual learner errors on parsing, summarizing the most frequent types of the parsing errors that were caused by learner errors, and the most frequent types of the learner errors that caused parsing errors.

Table 4.6 summarizes the proportion of the parsing errors that were caused by learner errors. As we can see, among the words that contained PEs, 39.2% had at least one LE-caused PE. Furthermore, when categorizing the PEs by the annotation types, we can see that the percentage of the LE-caused PEs increased across POS tags (37.5%), head indices (40.4%) and dependency labels (43.2%). This meant that among the three annotation types of the parser, dependency labels were most vulnerable to learner errors. A similar trend can be observed on the sentence level.

Table 4.7 summarizes the number of LEs and the proportion of LEs that caused PEs. In our learner English dataset, 53.5% of the sentences contained at least one LE. Among all the LEs, 63% caused at least one PE. The high percentages of LE-caused PEs among PEs and LEs showed that learner errors had a great impact on the parsing of learner English.

We then investigated the most frequent LE-caused PEs on a fine-grain level. Table 4.8 shows the types of LE-caused POS errors that occurred more than five times in our dataset

Table 4.7 Distribution of learner errors which caused parsing errors

Level	# (containing) LEs	# (containing) LE-caused PEs (%)
Sentence	535	48.4
LE	1,131	63.0

(see Appendix A for an explanation of the learner error types). These POS errors made up 39.8% of all the LE-caused POS errors.

One of the most frequent LE-caused POS errors was "JJ-NN" (adjective - noun). The causes of this parsing error included wrong derivation of nouns ("DN", e.g., in *some different* the correct form of *different* should be *differences*), missing a determiner ("MD", e.g., missing *a* in *I'm a pensioner*) or spelling errors ("S").

Another frequent POS error was "NNP-NN" (proper noun - noun). The main causes were missing a determiner ("MD") at the beginning of a sentence or inaccurate capitalization of a common noun ("C"). Verbs were sometimes misrecognized as nouns ("NN-VB") because of erroneous argument structure ("AS", e.g., *are over love their own babies* should be corrected as *love their own babies very much*), missing a preposition ("MT", e.g., missing *to* in *like to play badminton*) or using a wrong verb form ("FV", e.g., *think about change my career*), etc.

The errors of misrecognizing proper nouns as common nouns ("NNP-NN") and pronouns as foreign words ("FW-PRP") were exclusively caused by capitalization errors. Specifically, "FW-PRP" was caused by using the lower-case *i* for the first-person singular pronoun *I*. Except for these two types of POS errors, most LE-caused POS errors involved varied learner errors; sometimes more than one learner error contributed to an LE-caused POS error.

Table 4.9 shows the LE-caused dependency label errors that occurred more than five times in our dataset. These errors made up 28.1% of the LE-caused dependency label errors: compared to LE-caused POS error, LE-caused dependency label errors were more varied. The two most frequent types of dependency label errors concerned the core structure of the sentences: "ccomp-root" referred to misjudging a root as a clausal complement, and "root-parataxis" referred to misrecognizing a parataxis clause (i.e., a coordinate or subordinate clause without an explicit link verb) as a root. The major cause of these errors was comma splice. This learner error was marked by the learner error code "RP" (punctuation needs replacing), as the commas should be replaced by semi-colons or full-stops. Apart from the aforementioned two dependency label errors, other dependency label errors had no dominantly related learner errors, and may be caused by different types of learner errors.

We ranked the learner errors according to the frequency of the parsing errors they caused. Table 4.10 shows the top learner error types. It turned out that erroneous punctuation caused most parsing errors. This can also be observed from Table 4.8 and Table 4.9. Apart from

Table 4.8 Most frequent types of LE-caused POS errors (named by "wrong tag-correct tag")

POS error	Most frequent relevant LEs	Freq.
JJ-NN	DN[a](4)[b], MD(3), S(3), FA, UY, CN, AS	11
NNP-NN	MD(6), C(4), S, DN, W	11
NN-VB	AS(3), MT(2), FV(2), MD, MC, DN, DA	10
FW-PRP	C(8)	8
NN-NNP	C(7)	7
VBG-NN	S(2), FN(2), RP, UN, MD, MC, AS	7
NN-RB	S(3), RP(2), UT	6
NN-VBP	RP(3), DA(2), TV, FV	6
VB-VBP	S(3), RP(2), M	6
NNP-JJ	MD(3), W, C	5
RB-IN	S(2), UC, M, FV	5

[a] DN: wrongly derived noun; MD: determiner missing; S: spelling error; C: capitalization error; AS: incorrect argument structure; MT: preposition missing; FV: wrong form of verb; FN: wrong form of noun; RP: punctuation needs replacing; DA: wrongly derived pronoun. For the description of the other LEs, please refer to Appendix A.

[b] The bracketed numbers denote the frequencies of the LE-caused PEs that were caused by a particular learner error more than once. Since some PEs were related to more than one learner error, the sum of the learner errors relating to a PE may be larger than the frequency of the PE.

the comma splice, another major punctuation error was substituting a backtick (`) for an apostrophe in the contracted forms of verbs (e.g., *I'm*, *I've*), negations (e.g., *don't*), and the possessive form of nouns (e.g., *Asia's*), which caused problems including misjudging present tense verbs as common nouns ("NN-VBP") and misjudging the possessive morpheme *'s* as a root ("root-erased").

To conclude, this section confirmed that learner errors did have an impact on dependency parsing. The question then became why the parsers still achieved high performance if learner errors did have a significant impact. We turn to this issue in the next section where we compare the performance of the parsers on learner and native data.

4.2.3 Parser performance on learner English and native English

We evaluated the parsers on the native English dataset and compared the results to the evaluation on the MPB annotation of the EF1000 learner dataset. The gold standard of native dependency structures was achieved by converting Section 23 of the PTB-WSJ (Marcus et al., 1993) to the collapsed SD format.

Table 4.11 presents the evaluation results of the standard parsers on learner English (MPB annotation) and native English (PTB-WSJ section 23) on the word level. The accuracy scores

Table 4.9 Most frequent types of LE-caused dependency label errors (named by "wrong label-correct label")

Dependency label error	Most frequent relevant LEs	Freq.
ccomp-root	RP[a](21)[b], MP(3), S(2), UA, MC, FV, DA, CN	29
root-parataxis	RP(24), MP(3), MC, DV	29
nsubj-dobj	AS(4), MP(3), RP(2), MC(2), C	13
amod-nn	S(3), FN(2), C(2), UN, RP, MD, MC, AS	12
dep-parataxis	MP(4), AS(4), RP(3), DA(2), C(2), RT, CE	10
appos-conj_and	MC(3), RC(2), AS	6
vmod-root	RP(4), SP, MV	6
nn-amod	MD(2), W, S, RJ, C	6
advmod-erased	S(3), W, UN, M, FV, AS	6
root-erased	RP(2), MD, M, AS	5
aux-root	W, S, RP, MD, C, AGV	5
dep-dobj	UV, RP, M, DV, DA, C, AS	5
nn-conj_and	AS (2), SP, S, RC, MT	5
rcmod-parataxis	MP(3), RP(2)	5
acomp-xcomp	MD(3), MA, RC	5
root-aux	UV, UT, UA, S, FV	5

[a] RP: punctuation needs replacing; MP: punctuation missing; S: spelling error; AS: incorrect argument structure; MC: conjunction missing; FN: wrong form of noun; C: capitalization error; DA: wrongly derived pronoun; RC: conjunction needs replacing; MD: determiner missing. For the description of the other LEs, please refer to Appendix A.

[b] The bracketed numbers denote the frequencies of the LE-caused PEs that were caused by a particular learner error more than once.

Table 4.10 Learner errors that caused parsing errors most frequently

LE	Description	# LE-caused PEs
RP	Punctuation error	100
S	Spelling	76
C	Capitalization	55
AS	Wrong argument structure	54
MP	Missing a punctuation	47
MD	Missing a determiner	44
MT	Missing a preposition	43

of each standard parser were significantly lower ($p < 0.001$ according to chi-squared tests) on learner English than on native English. On average, the parsers achieved 95.46% vs. 96.69% on POS accuracy, 90.35% vs. 92.48% on UAS, 86.53% vs. 90.09% LAS, and 85.23% vs. 88.66% on the accuracy of all tags. The average performance gap between learner English and native English increased across the POS tag (1.23%), unlabeled attachment (2.13%), and labeled attachment (3.43%). This indicated that compared to POS tagging, dependency parsing might be subject to more influence from the difference between learner English and native English.

Even though the accuracy gaps between learner English and native English may seem small to human eyes, it did not mean that the parsers were robust to learner errors, as demonstrated in the previous section. Geertzen et al. (2013) argued that the seemingly high accuracy scores of a standard parser on learner English might result from the prevalence of short and simple sentences in learner English. To testify whether parsers performed better on shorter sentences than on longer ones, we grouped the native English sentences by sentence length, calculating the average parsing accuracy scores of each group that had more than five sentences, and computing the Pearson correlation between the accuracy scores and sentence length. It turned out that the UAS and LAS were significantly and negatively correlated with the sentence length (UAS: $r = -0.776, p < 0.01$; LAS: $r = -0.603, p < 0.01$). This meant that the performance of the parsers was indeed better on shorter sentences. Since the average sentence length of our learner English dataset was 13.5 whereas that of the native one was 23.5, the UAS and LAS gaps between learner English and native English were partly offset by the differences in sentence length. Nevertheless, POS tagging showed a positive correlation with sentence length ($r = 0.415, p < 0.01$); careful examination showed that this was because POS tagging was already quite accurate; when few POS errors occurred, shorter sentences had fewer words in total, which dragged down their POS accuracy scores.

On the other hand, the performance of the parsers on learner English seemed to correlate with their performance on native English. The best parser setting for learner English, the BLLIP parser trained on PTB-WSJ and Gigaword, also performed the best on native English except on POS tagging where it came second following the Berkeley parser. To verify the correlation, we ranked the parsers according to their performance on each dataset and computed the Spearman's rho correlation between the two rankings. It turned out that the correlation was significant on UAS ($r = 0.857, p < 0.05$), LAS ($r = 0.821, p < 0.05$) and the combination of all tags ($r = 0.900, p < 0.05$). Nevertheless, there was no significant correlation between the rankings on POS tags alone; this was possibly because the performance of the parsers on POS tagging was similarly high (maximum performance

Table 4.11 Accuracy of the parsers on learner English data and native English data

Parsing approach	Parser[a]	MPB annotation (learner)				PTB-WSJ section 23 (native)			
		POS	UAS	LAS	All	POS	UAS	LAS	All
c-parsing	SU	95.41	89.77	86.05	84.67	96.37	90.70	88.11	86.49
	SL	95.38	89.70	85.46	84.06	96.65	90.98	88.16	86.61
	BS	95.63	91.43	87.77	86.09	96.71	94.09	91.89	90.12
	BW	95.64	91.53	87.84	86.28	96.76	94.22	92.08	90.33
	BK	95.24	90.65	86.76	85.03	96.98	93.44	91.32	89.76
d-parsing	TB	–	90.53	86.77	–	–	92.67	90.20	–
	MT	–	88.85	85.06	–	–	91.26	88.85	–
Average		95.46	90.35	86.53	85.23	96.69	92.48	90.09	88.66
Max. Diff.		0.40	2.68	2.78	2.22	0.61	3.52	3.97	3.84

[a] SU: Stanford PCFG (unlexicalized) parser; SL: Stanford lexicalized parser; BS: BLLIP parser trained on OntoNotes-WSJ and Google Web Treebank; BW: BLLIP parser trained on PTB-WSJ and Gigaword; BK: Berkeley parser; TB: Turbo parser; MT: MaltParser.

difference was 0.40 on learner English, and 0.61 on native English) which made the ranking on the POS tag less meaningful.

The aforementioned correlation between the performance of dependency parsing on learner English and native English seemed to contradict the result of Krivanek and Meurers (2011), who showed that the MaltParser performed better on native German but worse on learner German than the WCDG parser. However, their study only compared two parsers, which made it impossible to identify a reliable correlation between the performance on learner data and native data. Furthermore, their study compared a rule-based parser to a probabilistic parser, whereas our study compared a number of probabilistic parsers. Last but not least, they investigated learner German. German has a different word order and morphological cues on nouns and verbs compared to English; as a result, the impact of learner errors on the dependency parsing of German may well be different. Nevertheless, based on our study, we can safely conclude that the performance of a probabilistic parser on native English can predict its performance on learner English[3].

[3]There was also a difference in both the genre and the topic between the native English training data for the parsers (newswire, and mostly about finance) and the learner data (essay, and mostly about daily life). This did not affect our conclusion about the existence of the correlation between the parser performance on native English and learner English, because a parser was less likely to perform consistently (i.e., showed correlated performances) on datasets with larger differences, whereas the correlation existed despite such differences in the genre and the topic. Meanwhile, the result seemed to indicate that the parser performances across different genres and topics might also be correlated. However, the number of genres and topics here was limited – more research is needed before we can testify the general existence of such correlation.

4.3 Summary

In this chapter, we investigated the performance of multiple standard probabilistic parsers on learner English, the annotation bias in the evaluation, the effect of learner errors on parsing and the correlation between the performance of the standard parsers on native English and learner English. Our answers to the research questions posed at the beginning of this chapters were as follows.

1. What is the accuracy of different standard parsers on learner English?

We found that on average, current standard parsers achieved around 95% on POS accuracy, 90% on UAS, 87% on LAS, and 85% on the accuracy of all tags on learner English. The performance differences between the parsers were smaller on POS tags than on dependency relations.

2. Is there annotation bias in the gold standard created by manually correcting the output of a single parser? If there is, how does the annotation bias influence the accuracy scores?

We showed that there was annotation bias when the gold standard for evaluation was annotated by manually correcting the output of a parser. This annotation bias arose from the inherent ambiguity of some linguistic structures, the annotation schemes, and learner errors. The annotation bias reduced the recall of parsing errors during annotation; using the gold standard that contained the annotation bias can significantly influence the result of the parser evaluation in favor of the pre-annotation parser.

The annotation bias can be reduced in several ways. Firstly, displaying the annotation mismatches of several parsers can help annotators reduce the annotation bias. The effectiveness of the reference provided by a parser depends on its accuracy. Nevertheless, the marginal benefit of adding a reference parser diminishes as the number of reference parsers increases, because the correct references provided by the parsers may overlap whereas the reference from the additional parser needs more time to review. Therefore, one should weigh reducing annotation bias against maintaining annotation efficiency when using the contrast-based annotation method. Secondly, the annotation bias may be controlled for by improving the annotation scheme for parsing. In particular, we need principles that can help to distinguish the ambiguity arising from learner errors. Multi-layered annotation (Dickinson and Ragheb, 2009) which uses different layers of features to describe the contradictory aspects of learner errors may be a way forward. However, as our results show, learner errors may lead to ambiguity where many interpretations of the structure are possible. This ambiguity poses a challenge to the design of appropriate layers for annotation.

3. What is the impact of fine-grained learner errors on parsing?

We found that learner errors did have an impact on parsing output. More than one-third of the parsing errors were caused by learner errors, and over 60% of the learner errors caused at least one parsing error. These results indicated that the parsers were not very robust to learner errors. Learner errors on punctuation, spelling, capitalization, argument structures, determiners and prepositions caused most parsing errors. Correcting these learner errors can be an effective pre-processing technique to reduce parsing errors for downstream linguistic research and NLP applications based on learner English data [4]. Given the impact of learner errors on parsing, it was surprising that the accuracy scores of the parsers on learner English were lower than those on native English by only small margins. As we showed, this was because the average sentence length of learner English was shorter than that of native English. The impact of learner errors was therefore offset by the simplicity of learner language. Furthermore, not every learner sentence contained learner errors, and when a sentence did contain a learner error, it only affected the parses of a limited number of words in the sentence. The accurate parses of short learner sentences that had no learner errors helped maintain a high face value of the parsing accuracy for learner English.

4. What is the relation between the performance of standard parsers on learner English and native English?

We demonstrated that the performance of probabilistic parsers on learner English can be predicted by their performance on native English. The implication is that when it comes to choosing a probabilistic parser for learner English, the most accurate parser evaluated on native English is a good choice. Alternatively, if one wants to apply a probabilistic parser on a specific learner English dataset, he or she can roughly predict the accuracy of the parser according to its accuracy evaluated on native English.

[4] We propose that only the learner errors irrelevant to downstream research or application goals can be corrected. However, correcting learner errors can influence the authenticity of learner data. Future research may look into what degree of pre-processing is appropriate for learner data.

Chapter 5

Automatic SCF identification

In this chapter, we present an SCF identification system for learner language. As we have mentioned in Chapter 2, the acquisition of SCFs has been a long-standing interest in L2 research, and learner corpora can provide useful insight on SCF acquisition (Ellis and Ferreira–Junior, 2009; Meurers et al., 2013; Römer et al., 2014; Tono, 2004). However, existing corpus-based research on SCF acquisition relies on SCF annotation obtained by manually annotating the output of standard POS taggers and parsers. The output of such generic syntactic analysis systems requires discrimination between complements and adjuncts by humans. Furthermore, as we have demonstrated in the previous chapter, the performance of standard POS taggers and parsers drops on learner data, which adds to the burden of the manual edition. An SCF identification system is needed to facilitate efficient analysis and annotation of SCFs on large-scale learner corpora.

We propose an SCF identification system that can analyze the SCFs of individual verb tokens contextualized in sentences. We approach the task as a supervised classification problem, training a classifier on SCF annotation. This chapter first describes our annotation of SCFs on learner data. We then present the model, features and technical details of the system. Finally, we report on the training, evaluation and error analysis of the system.

5.1 SCF annotation on learner English

We annotated EF1000 for SCFs. To extract verb tokens for the annotation, we identified verbs based on the gold standard POS tags and dependencies of EF1000 (i.e., the MPB annotation in Chapter 4), choosing the verbs that were not auxiliaries, adjective modifiers or gerunds with the following criteria:

1. The POS tag of the word contained “VB”;

2. The dependency relation of the word was not "aux" (auxiliary, e.g., *has* in *he has left*), "auxpass" (passive auxiliary, e.g., *been* in *he has been understood*), "amod" (adjectival modifier, e.g., *frozen* in *frozen food*), or "nn" (noun compound modifier, e.g., *the swimming pool*).

In total, 1,987 verbs were identified. We then used the SCF inventory of native English to annotate the learner data. SCF learner errors were annotated based on surface evidence. For example, in the sentence *I waited John*, the SCF of *waited* was annotated as "dobj_N" (a direct object). For a learner SCF that was not in the SCF inventory (e.g., the SCF of *dream* in *I dream about travel around the world* contained a prepositional complement erroneously headed by a base-form verb, which can be termed as a new frame called "pcomp_VBARE"), we annotated it as "new frame".

Two Linguistics PhD students participated in the annotation of SCFs. The annotators first learned the SCF inventory and an annotation guideline developed based on the work of Meyers et al. (1996) and a previous SCF annotation project (Quochi et al., 2014). The annotators then went through two training sessions. In each session, they annotated 100 verb tokens independently. The author of this thesis also annotated the training sentences. At the end of each training session, the annotators and the author compared their annotations, discussing and resolving disagreements. After the training, the two annotators continued to annotate the rest of 1,787 verb tokens independently. 83.7% of their annotations agreed with each other. The author reviewed the disagreements and decided the final annotation. The final annotation showed that the incidence of SCF learner errors was low: 12 (0.6%) verb tokens were annotated as "new frame"; 68 (3.4%) verb tokens had wrong SCFs (e.g., *I waited John* instead of *I waited for John*) and 20 verb tokens (1.0%) had fine-grained errors in the choice of prepositions or particle. Since new frames were rare and varied, they cannot be reliably classified by a machine learning model. We, therefore, removed the verb tokens annotated as "new frame" from the dataset. As a result, the SCF learner corpus contained 1,966 verb tokens.

Table 5.1 shows the number of verb lemmas, verb types, verb tokens (i.e., the number of sentences), corpus word tokens, SCF types and overlapping SCF types between the two corpora. Note that the number of word tokens in the SCF learner corpus (32,196) was larger than the number of word tokens in EF1000 (12,003), because more than one verb may be identified in a sentence and the sentences of such verbs were duplicated in the SCF learner corpus. Moreover, the native and learner datasets had different scales in the numbers of verb lemmas and verb types, because these two datasets were created by using different approaches to select the verb tokens: The native dataset was created by sampling from BNC around 250 sentences for each of the 24 verb lemmas; these verb lemmas were chosen by

Table 5.1 Statistics of the SCF datasets

	BNC(native)	EF1000(learner)
# verb lemma	24	360
# verb type	99	568
# verb token	6,133	1,966
# corpus word token	186,534	32,196
# SCF type	43	38
# Overlapping SCF type	32	

Table 5.2 Distribution of the ten most frequent SCFs in the learner English SCF corpus

#[a]	SCF	%	Example (learner English SCF corpus)
23	dobj_N	32.7	If you **follow** [this advice], ...
50	pobj	11.8	At 7 o'clock I **go** [to work].
1	acomp	10.8	So it **became** [well-known] , worldwide.
66	xcomp_N	9.8	I want to **become** [the new president].
32	dobj_N:pobj	6.0	You can **bring** [them] [to the back entrance].
68	xcomp_VBARE=>aux_TO	5.6	John is **going** [to tell Isabella about that] .
65	su	4.9	I would like to **cook** on Saturday .
9	ccomp_VTENSED	3.1	I **think** [you should buy these].
42	dobj_N:xcomp_VBARE=>aux_TO	1.8	Can I **force** [them] [to fix the house]?
60	prt	1.4	I was ... and **fell** [down].

[a] # denotes the number of an SCF in Appendix B.

a linguist to represent a wide range of subcategorization behaviors (Quochi et al., 2014). Contrastingly, the learner dataset was created by identifying verb tokens from sentences without restriction on the verb lemmas. As a result, the learner dataset had a higher number of verb lemmas and verb types.

As we can see from Table 5.1, only about half of the SCF types in the inventory actually appeared in the datasets: the native English dataset had 43 SCF types (indicated by "N" in Appendix B), while the learner dataset had 38 SCF types (indicated by * or $\diamond$ in Appendix B). SCF distributions tend to be Zipfian (Korhonen et al., 2000), and the SCF types absent in the data are rare in real-world situations. This was proved by the distribution of the ten most frequent SCFs in the learner English data (Table 5.2) and the native English data (Table 3.3). Table 5.1 also shows the overlap of SCF types across the two datasets (32 types). Since each dataset contained SCF types that were absent in the other dataset, using both of them as training data can increase the coverage of the SCF types.

5.2 Model

Selecting an appropriate machine learning model is important for realizing accurate classification. Recent years have seen a great success of neural network models on many classification tasks. However, these tasks tend to have a large amount of annotated training data, whereas the annotated training data available for our task was limited (8,099 instances at the maximum, see Table 5.1). Indeed, we experimented with various neural network models, ranging from stacks of Long Short-term Memory (Hochreiter and Schmidhuber, 1997) recurrent neural networks or convolutional neural networks (LeCun et al., 1989) to variants of attention-based sequence-to-sequence architecture (Bahdanau et al., 2014). Results showed that even the best of these neural network models performed slightly worse than a simple maximum entropy (MaxEnt) model (Berger et al., 1996). We also experimented with the Support Vector Machine (SVM) model (Cortes and Vapnik, 1995), which was very useful for classification in high dimensional space. Results showed that SVM also performed slightly worse than MaxEnt. This meant that MaxEnt was most suitable for our case. MaxEnt has proved to be useful in automatic syntactic analysis such as POS tagging (Ratnaparkhi, 1996) and parsing (Charniak and Johnson, 2005). Furthermore, MaxEnt can handle mixtures of boolean, integer and real-valued features, and can distribute weights to correlated features. This suited the characteristics of our features (which are introduced below). As a result, we employed MaxEnt as our classifier for SCF identification.

The MaxEnt model was defined as follows: given an SCF inventory $S = s_1, s_2, ..., s_n$, the probability of assigning an SCF $s_i (i = 1, 2, ..., n)$ to a verb token v was

$$p(s_i|v) = \frac{\exp(\theta_{s_i}^T \mathrm{f}(v))}{\sum_{i=1}^{n} \exp(\theta_{s_i}^T \mathrm{f}(v))} \tag{5.1}$$

where $\mathrm{f}(\cdot)$ was a feature function, and $\theta_{s_i}^T$ was the parameter of s_i. The formula basically calculated a score for s_i based on a linear combination of the features, and divided the score with the sum of the scores of all SCFs to obtain a probability score. The SCF of the highest probability score was assigned to the verb token.

We used four types of linguistic information to create our features: words, POS tags, dependency relations, and word embeddings. Words, POS tags, and dependency relations have proved to be useful in capturing SCF information (Baker et al., 2014), and word embeddings have been playing a key role in the recent improvement of many automatic syntactic annotation systems such as dependency parsers (Andor et al., 2016). A word embedding is a distributional vector representation of a word (Mikolov et al., 2013). Semantically similar words tend to have similar word embeddings. For example, the word embedding of *reply*

is more similar to that of *respond* than that of *stay*. Since semantically similar verbs tend to have similar SCFs (Levin, 1993), the word embeddings of predicates can be useful for identifying SCFs.

More specifically, we extracted the following features for a given predicate:

1. The combinations of the word, POS tag, and dependency relation of a child, a grandchild, or a great-grandchild that was *whether*, *if*, or a wh-word. These features were intended to capture the potential complements of the predicate. Take the predicate *thought* in Figure 5.1 for example. Its child *about*, grandchild *go*, and the great-grandchild *whether* were considered for feature extraction. The features extracted for the child *about*, of which the POS tag was IN and the dependency relation was "prep", included seven combinations: "ch_about", "ch_IN", "ch_prep", "ch_about_IN", "ch_about_prep", "ch_IN_prep" and "ch_about_IN_prep" ("ch" denoted that the features were related to a child).

2. The full combination of the word, POS tag and dependency relation of a parent, a grandparent or a sibling of the predicate. These features were intended to capture information in the head words and conjuncts of the predicate which may be useful for inferring SCF. For example, the feature extracted for the sibling *smiled* was "sb_thought_VBD_root" ("sb" denoted that the feature was related to a sibling).

3. The n-grams of the lexicalized or unlexicalized combinations of the word, POS tag, and dependency relation of the neighboring words of the predicate. These features were intended to capture the context of the predicates. At most one word to the left, and at most three words to the right of the predicate were considered. This imbalanced window was designed following the observation that most SCF information was located to the right of a predicate. We extracted unigram and bigram within the window. The lexicalized features involved both words and dependency relations, whereas the unlexicalized ones involved the dependency relation and the position of the word with regard to the predicate. For example, the unlexicalized bi-gram feature for the two neighbouring words *about* and *whether* was "du_1_prep_2_mark" ("du" denoted that the feature was related to the neighbouring words and was unlexicalized). The word position information was excluded from the lexicalized features to avoid data sparsity issue for machine learning.

4. The word and word embedding of the predicate. These features were intended to capture information about the predicate. For example, we extracted "tg_thought" ("tg" denoted that the feature was related to the predicate) and the word embedding of *thought* as a feature.

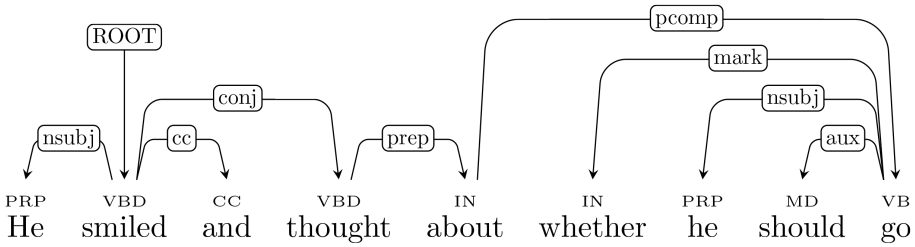


Fig. 5.1 POS tags and dependency structure of an example sentence

5.3 Technical details

As our parser evaluation in Chapter 4 shows, if a parser achieves better accuracy than other parsers on native English, this parser also tends to be more accurate on learner English. As a result, we used SyntaxNet – a state-of-the-art syntactic parser for native English (Andor et al., 2016) – to extract POS tags and dependency relations. SyntaxNet achieved an accuracy of 97.4% in POS tagging and 92.8% LAS in dependency parsing on PTB-WSJ[1].

We used a word embedding model trained on the English Polyglot Wikipedia corpus (Al-Rfou et al., 2013) with skip-gram negative sampling in the bag-of-word context with the embedding dimensionality of 300 and the window size of 2 (Gerz et al., 2016, hereafter referred to as the PW model). Out-of-vocabulary words were mapped to vectors of zeros. Although the PW model was not trained on the same domains as all of our SCF corpora, it was strong for our study: we also experimented with training word embedding models on each SCF corpus domain, performing grid search across various hyper-parameters of word2vec (Mikolov et al., 2013) to determine the best settings for SCF classification. None of these in-domain word embedding models outperformed the PW model, possibly because the SCF domain corpora were much smaller than the English Polyglot Wikipedia corpus. While future work can look into developing good in-domain word embedding models, the PW model sufficed for the purposes of this study.

5.4 Training and evaluation

We trained the model in two data settings. The first setting used learner data only. We conducted a 10-fold cross-validation on the learner data. In other words, we partitioned the learner data into ten subsets; we trained the model on nine subsets at a time and tested the model on the remaining subset; this process was repeated ten times so that all subsets

[1]SyntaxNet was absent in the parser evaluation on learner data in Chapter 4 because it was released after the evaluation.

have been used for testing. The average accuracy of this setting was 82.1%. In the second setting, we added native data to training. The average accuracy of this setting was 84.2%. This meant that adding native data during training helped to improve the accuracy of SCF identification on learner data [2]. As a result, we trained our model on both learner and native data. Furthermore, we conducted a leave-one-out experiment with 10-fold cross-validation on the model, i.e., we removed one type of information (words, POS tags, dependency relations or word embedding) from the features at a time during the training. All experiments led to decreased accuracy, which meant all types of information were important for SCF identification. As a result, we used the full features to train the model. The final model was regarded as the SCF identification system.

In the following sections, we report a more detailed evaluation of the SCF identification system. We evaluated the accuracy statistics of the system on individual SCFs. We also analyzed the SCF errors made by the system.

5.4.1 SCF identification accuracy

We evaluated the precision, recall and F1 score of individual SCF types during the 10-fold cross-validation of the SCF identification system. The system was able to classify 49 SCF types, which were the union of the SCF types that occurred in both learner data and native data. However, 11 SCF types appeared only in native data (indicated by "N" alone in Appendix B), which meant we cannot evaluate their accuracy on the learner data. Moreover, some SCFs were rare in the learner data, which made their evaluation unreliable. For example, when an SCF type had only two verb tokens, the training set might include none of the verb tokens, which made it impossible for the model to classify the SCF type correctly; even if the training set and testing set had one verb token each, the accuracy scores of this SCF type would be either a hundred or zero per cent, depending on whether the verb token in the testing set was classified correctly or not. Such accuracy rates were uninformative. As a result, we omitted 14 SCFs that had less than five verb tokens in the learner data (indicated by $\diamond$ in Appendix B). Table 5.3 lists the remaining 24 SCF types. The first column denotes the number of an SCF in Appendix B, where the guideline examples of the SCFs are available.

[2] It will be useful to evaluate how the accuracy increases with more training data gradually, as the result can help to predict whether annotating more learner data can improve the accuracy. We leave this to future work.

Table 5.3 Precision (P.), recall (R.) and F1 score of SCF identification of individual SCF types on EF1000

#	SCF	Example	P.	R.	F1	Freq.
68	xcomp_VBARE=>aux_TO	I **prefer** [to avoid sitcoms at all].	.93	.98	.96	110
66	xcomp_N	I want to **become** [the new president].	.97	.87	.92	192
1	acomp	... helps us **feel** [easier].	.86	.95	.90	212
23	dobj_N	I urge you to **consider** [it].	.90	.90	.90	643
10	ccomp_VTENSED=>mark-that	Someone **mention** [that you are untidy].	.87	.93	.90	28
50	pobj	I **go** [to bed] at twelve o'clock.	.83	.89	.86	232
42	dobj_N:xcomp_VBARE=>aux_TO	Can I **force** [them] [to fix the house]?	.86	.86	.86	35
9	ccomp_VTENSED	I **think** [the beige sweater is expensive].	.80	.92	.86	61
69	xcomp_VING	I **like** [playing tennis].	.86	.72	.78	25
60	prt	To **sum** [up], ...	.76	.79	.77	28
41	dobj_N:xcomp_VBARE	**Let** [me] [tell you why ...]	.77	.77	.77	13
39	dobj_N:xcomp_ADJ	... to **make** [them] [heavier].	.80	.73	.76	11
24	dobj_N:iobj	Let me **tell** [you] [what I did].	.77	.65	.71	26
65	su	I can drive and **sing**.	.68	.74	.71	96
32	dobj_N:pobj	John is going to **tell** [Isabella] [about that].	.70	.68	.69	112
36	dobj_N:prt	... if you **give** [up] [your studies].	.71	.65	.68	26
48	pcomp_VING	... **forget** [about asking the prices].	1.0	.44	.62	9
54	pobj:prt	I **look** [forward] [to the start of classes].	.75	.50	.60	12
44	dobj_N:xcomp_VING	I must **spend** [four years] [finishing my university life].	.60	.50	.55	6
34	dobj_N:pobj:prt	... **put** [down] [your ideas] [as bullet points].	.67	.44	.53	9
18	ccomp_WHCOMP	... you will **learn** [how to handle emergent case timely].	.55	.46	.50	13
3	advmod	Then **turn** [left] at Green Ave.	.35	.62	.44	13
67	xcomp_VBARE	... I **like** [take a walk].	.60	.33	.43	9
4	advmod:dobj_N	... **spend** [our time] [there].	.00	.00	.00	6

Table 5.4 SCF confusion pairs during testing

Target	Prediction	Freq.
dobj_N	dobj_N:pobj	22
dobj_N:pobj	dobj_N	19
xcomp_N	acomp	19
dobj_N	su	12
dobj_N:pobj	pobj	11
su	pobj	10
pobj	dobj_N:pobj	7
dobj_N:iobj	dobj_N	6
dobj_N	ccomp_VTENSED	5
su	dobj_N	5
dobj_N	dobj_N:iobj	5
pobj	su	5
pobj:pobj	pobj	5

As we can see, the majority of the SCF types were classified accurately. Eight SCF types, which accounted for 77% of the learner data, were identified with an F1-score of over 85%. Contrastingly, six SCF types, which accounted for only 3% of the learner data, were identified with an F1-score of less than 60%. To some extent, the low accuracy of the rare SCF types was caused by the scarcity of their training data for the model.

5.4.2 SCF error analysis

We analyzed the SCF identification errors during testing to find out what SCF types were challenging for our system, and to diagnose the cause of the SCF identification errors. Table 5.4 lists the SCF misanalysis pairs that occurred at least five times during testing.

The most frequent misanalysis was found between “dobj_N” (a direct object) and “dobj_N:pobj” (a direct object and a prepositional object), which related to a decision on whether a prepositional object should be considered as part of an SCF or not (i.e., a complement or an adjunct). Similarly, the misanalysis pair of “su” (intransitive) and “pobj”, as well as the misanalysis pair of “pobj:pobj” (two prepositional objects) and “pobj” involved a decision about a prepositional object. Further analysis revealed that there were two main causes of the misidentification of SCFs with regard to prepositional objects as follows.

Distinction between arguments and adjuncts

The SCF identifier erroneously considered the temporal prepositional object in (s) as an adjunct, misidentifying “dobj_N:pobj” as “dobj_N” (we highlight the predicate with square

brackets and denote the SCF assigned by the identifier in subscript). Even though the verb *do* rarely takes a prepositional object as a complement, and a temporal prepositional object is usually an adjunct, the phrase *in 1874* was a complement of *done* due to the criterion of obligatoriness (Meyers et al., 1996): sentence (s) would be ungrammatical if the temporal prepositional object was removed. Sentence (t) illustrates an SCF misanalysis in the opposite direction: the SCF identifier erroneously included the locational prepositional object *on her birthday party* in (t) as a complement, misidentifying "dobj_N" as "dobj_N:pobj".

(s) It's an oil painting $[\text{done}]_{\text{dobj\_N}*}$ in 1874.

(t) Jane would like to $[\text{see}]_{\text{dobj\_N:pobj}*}$ you on her birthday party.

Sentence (s) illustrates that the boundary between complements and adjuncts is not straightforward. As a result, the difficulty in distinguishing between a complement and an adjunct for a prepositional object was caused not only by the limitation of our model, but also the inherent fuzziness between complements and adjuncts (Somers, 1984).

Prepositional attachment

Misidentification errors regarding prepositional objects were often caused by prepositional attachment errors. For example, the prepositional object in sentence (u) should be attached to (i.e., be a dependent of) the noun phrase *sales figure*. However, the SCF identification system erroneously considered the prepositional object as a complement of the predicate *provides*, resulting in the misidentification of "dobj_N" as "dobj_N:pobj". This problem was mainly caused by the dependency parser. Prepositional attachment is a notoriously difficult task in NLP.

(u) The graph $[\text{provides}]_{\text{dobj\_N:pobj}*}$ sales figures for international sales and ...

(v) As you can $[\text{see}]_{\text{pobj}*}$ on my CV ...

(w) What do I wish to $[\text{do}]_{\text{su}*}$?

The SCF identifier also misidentified "xcomp_N" (a nominal complement) as "acomp" (an adjectival complement) sometimes. Further analysis showed that most of such errors happened on nominal complements headed by proper nouns (e.g., *Werner* in *My name is Werner* was misidentified as an adjectival complement). These errors might be caused by the scarcity of cases where a nominal complement was headed by a proper noun in the training sets during the cross-validation. Furthermore, the SCF identifier sometimes omitted a direct object, misidentifying "dobj_N:pobj" as "pobj", e.g., in (v), and misidentifying "dobj_N" as

"su" (intransitive), e.g., in (w). Such errors were mainly found when a direct object preceded the predicate. This problem was also caused by the scarcity of such structures in the training sets during the cross-validation.

5.5 Summary

In this chapter, we described the development of the first SCF identification system for learner language. The system can label individual verb occurrences in learner corpora for a set of 49 distinct SCFs ranging from basic transitive and intransitive frames to complicated frames that involve prepositional, verbal or clausal complements. The system included a MaxEnt model trained on both learner English data and general-domain native English data, achieving an accuracy of 84.2%. Although direct comparison of different SCF systems was difficult because of the varying degree of supervision involved, the differences in SCF inventories, and the use of different training and evaluation sets, this level of accuracy was among the highest reported among contemporary systems (see Chapter 2.3.3) and was likely to be sufficient for benefit in downstream tasks. In the next chapter, we will illustrate how the system can be useful for L2 research [3].

Furthermore, the development of our SCF identification system provided useful implications for linguistics, the annotation of non-native language data and the development of NLP techniques for such data. First, the usefulness of our features for the model proved that words, syntactic categories, syntactic relations, and distributional semantics were all important for the identification of SCFs. Second, our annotation practice demonstrated that native language schemes can be used to annotate non-native language data, as our native English SCF inventory proved to cover 99.4% of the learner SCFs. Third, our results demonstrated that for machine learning on a small training dataset, a simple MaxEnt model can perform better than deep neural network models. Furthermore, we showed that when the non-native data was limited, including native language data can help to improve the accuracy of the classification.

[3] In the next chapter, we present a study of some general linguistic phenomena over a large amount of data. For such studies, the SCF identification system can be used without any human correction, because it is infeasible to annotate the large amount of data, and the scale of such data guarantees that the observed linguistic phenomena is statistically robust despite some noise in the data. However, for the tasks that require completely accurate linguistic information, the system can only be used as a preprocessing technique and human correction is needed.

Chapter 6

Application of automatic SCF identification: investigating L2 SCF diversity

In this chapter, we illustrate how automatic SCF identification can be useful for L2 research. The major advantage of the SCF identification system lies in the scale of SCF data it can produce. The system can facilitate SCF annotation, and can support searching and analyzing SCFs on large-scale corpora. To illustrate the interesting studies we can do with a large amount of SCF data, we investigate how L2 learners diversify their use of SCFs in text and how this diversity changes with L2 proficiency.

No research has been conducted to investigate the diversity of SCF use in L2 learning, as it requires a large amount of SCFs data. Nevertheless, such research has potential value for L2 research and education. First, while it is intuitive to hypothesize that L2 learners can use a wider range of SCFs as their L2 proficiency develops, it is unclear how L2 learners diversify their use of SCFs in text and how this diversity changes across different proficiency levels. For example, do L2 learners repeat fewer SCFs in text when their proficiency improves? How about their distribution of SCFs in text – do L2 learners distribute different SCFs more evenly as their proficiency improves? Answers to such questions can help researchers to better understand L2 SCF learning, and can assist L2 educators to teach and develop educational material for L2 learners at different stages.

Second, the diversity of SCF use may reflect an aspect of linguistic complexity, and has the potential to contribute to linguistic complexity research. Linguistic complexity is the ability to use a wide range of sophisticated elements in L2 (Bulté and Housen, 2012), in which the diversity of the elements is an important factor. Previous studies have shown that lexical diversity indices are useful predictors of language proficiency (Jarvis, 2013). It is

possible that the diversity of morphosyntactic behaviors such as SCFs may also reflect an important aspect of language proficiency. Furthermore, researchers are calling for more specific and multidimensional metrics of linguistic complexity, as different dimensions of linguistic complexity may not increase linearly with proficiency and it is more informative to portrait them separately (Norris and Ortega, 2009). While SCF diversity may relate to existing syntactic complexity to some extent (e.g., larger SCF diversity may be related to more varied use of subordination), we hypothesize that, as a construct that links lexis and morphosyntax, SCF has its unique properties and SCF diversity may contribute a new aspect to linguistic complexity. To test this hypothesis we design a number of SCF diversity metrics, and investigate two research questions based on analysis on EFCAMDAT[1]:

1. How does SCF diversity change with English L2 proficiency in general?
2. Does SCF diversity contribute to the prediction of English L2 proficiency beyond current syntactic complexity measures?

In this chapter, we first design multi-dimensional SCF diversity metrics. We then measure the SCF diversity of the learner essays in EFCAMDAT, and investigate the relation between SCF diversity and L2 proficiency.

6.1 Design of SCF diversity

We drew inspiration from the work of Jarvis (2013) about lexical diversity to design the SCF diversity metrics. Jarvis (2013) argues that while the linguistic community mainly uses variants of the type-token ratio as lexical diversity metrics, these metrics reflect only one aspect of lexical diversity, i.e., word repetition, and lack construct validity. He suggests that a construct-valid measure of lexical diversity should include a composite of metrics gauging different dimensions of lexical distribution; such metrics can be established by analogy to species diversity metrics in ecology such as size (the number of tokens), richness (the number of types), effective richness (the number of types adjusted by the number of tokens for each type), evenness (the degree to which the tokens are portioned equally among the types), disparity (the taxonomical difference of the types), importance (the relative frequency of a type in a general corpus), and dispersion (the average distance between the tokens of the same type).

[1]To maintain a narrow focus for the thesis, our investigation of SCF diversity is preliminary. We leave an in-depth theoretical exploration of the relation between SCF diversity and linguistic complexity to future work.

We believed that SCF diversity was also multidimensional, and designed the SCF diversity metrics from various angles. The following section introduces the basic concept of each metric. We then introduce how to control these metrics for text length in the next section.

6.1.1 The basic concepts of SCF diversity metrics

We introduce the basic concepts of SCF diversity metrics and illustrate the calculation of the metrics with an example learner text below (extracted from the EF-Cambridge Language Database (EFCAMDAT) (Geertzen et al., 2013)). The text contains nine verbs (we highlight the predicate with square brackets and denote the SCF in subscript). These verbs instantiate a range of SCFs. Two verbs instantiate the intransitive SCF ("su", subject): *progressing* and *married*. Three verbs instantiate the transitive SCF ("dobj_N", direct object): *moved*, *love*, *find*. Two verbs take a prepositional object ("pobj"): *break*, *fall*, while *break* additionally takes a particle *up*; one verb, *decided*, takes an infinitive complement clause ("xcomp_VBARE=>aux_TO") and one verb, *feeled* (a learner spelling error), takes a tensed clausal complement ("ccomp_VTENSED").

> After some time, the affection between them is [progressing]$_{\text{su}}$ well. John's personality deeply [moved]$_{\text{dobj\_N}}$ Isabella. So Isabella [decided]$_{\text{xcomp\_VBARE=>aux\_TO}}$ to [break]$_{\text{pobj:prt}}$ up with Tom and [fell]$_{\text{pobj}}$ in love with John. John also [feeled]$_{\text{ccomp\_VTENSED=>mark-that}}$ that Isabella was the woman he [loved]$_{\text{dobj\_N}}$ deeply. To his joy, he could [find]$_{\text{dobj\_N}}$ his true love during his travel. In the end, they [married]$_{\text{su}}$ together.
>
> (Level 6; Unit 1; Lesson aim: Writing a Movie Plot)

We designed 13 SCF diversity metrics from varied diversity angles as follows.

SCF size: SCF size referred to the number of SCF tokens or verb tokens. The SCF size of the example text was 9.

SCF richness: SCF richness referred to the number of SCF types. This concept reflected the SCF types at the command of a learner. The SCF richness of the example text was 6.

SCF type-token ratio (SCF TTR): SCF TTR referred to the ratio between SCF richness and SCF size. SCF TTR reflected the degree of repetition of SCFs. The SCF TTR of the example text was 66.7% (6/9), as the learner used "dobj_N" three times and "su" twice.

SCF entropy: SCF entropy reflected the level of uncertainty in deciding the SCF type of a verb token. Entropy, or Shannon's index, was defined as the negative sum of the product between the proportion of each individual type and the natural logarithm of the proportion (Shannon, 1948). For now and the rest of the chapter, we use p_i to denote the proportion of the i-th SCF type among all SCF types, and we use R to denote SCF richness. The formula for SCF entropy was:

$$Entropy = -\sum_{i=1}^{R} p_i \log(p_i) \tag{6.1}$$

To illustrate, the proportion of "dobj_N" in the example text was 33.3 (3/9), and the SCF entropy of the text was 1.68. When we calculated the entropy using logarithms to base two, the entropy was the minimum number of yes/no questions required on average (Jost, 2006) to determine the type of an SCF. SCF Entropy was affected by SCF richness and the evenness of the proportions of individual SCF types: higher SCF richness and more even proportions of individual SCF types led to higher SCF entropy.

SCF Gini-Simpson index (SCF GS): SCF GS referred to the probability that two SCFs randomly sampled in succession had different SCF types. Similar to SCF TTR and SCF entropy, SCF GS reflected the degree of SCF repetition. The formula of SCF GS was:

$$GS = 1 - \sum_{i=1}^{R} p_i^2 \tag{6.2}$$

The SCF GS of the example text was 0.79.

Effective SCF richness: Effective SCF richness referred to the number of equally-abundant SCF types required for an SCF repetition index. In the field of ecology, effective richness is considered to reflect 'true diversity', because it depends on both richness and evenness, assigning less weight to rare species which does not effectively contribute to the diversity of a community (MacArthur, 1965). Effective SCF richness can be calculated from either SCF entropy or SCF GS. When the SCF repetition index was SCF entropy, the effective SCF richness was calculated by the exponential of SCF entropy, e.g., 5.35 ($e^{1.68}$) for the example text. When the SCF repetition index was SCF GS, the effective SCF richness was calculated by subtracting SCF GS from unity and inverting the value, e.g., 4.76 ($\frac{1}{1-0.79}$) for the example text. The entropy-based effective SCF richness weighted each SCF type according to the number of the verb

tokens of the SCF type, whereas the GS-based effective SCF richness biased even more towards frequent SCF types. More specifically, let us look at the formula for effective SCF richness:

$$^{q}D = (\sum_{i=1}^{R} p_i^q)^{\frac{1}{1-q}} \tag{6.3}$$

The parameter q reflected the relative weight placed on frequent SCF types versus rare SCF types: the higher q was, the higher the weight was put on frequent SCF types. When q approached one, the formula corresponded to the entropy-based effective SCF richness, while $q = 2$ corresponded to the GS-based SCF effective richness. Furthermore, $q = 0$ corresponded to SCF richness, which assigned an equal weight to every SCF type.

SCF evenness: SCF evenness referred to how evenly verb tokens were allocated among different SCF types. We evaluated SCF evenness from two angles. The first one was to divide SCF entropy by the maximum value of SCF entropy given the same SCF richness:

$$Evenness\_entropy = \frac{H'}{H'_{max}} = \frac{H'}{-\sum_{i=1}^{R} \frac{1}{R} \log(\frac{1}{R})} = \frac{H'}{\ln R} \tag{6.4}$$

SCF entropy reached the maximum when all SCF types had the same number of verb tokens, i.e., the probability of each SCF type was $\frac{1}{R}$. The entropy-based SCF evenness ranged between zero and one: the higher it was, the more evenly the SCFs were allocated among different SCF types. The entropy-based evenness of the example text was 0.94. Note that the denominator of the equation became zero when the text had only one SCF type. In this case, the entropy-based SCF diversity was inapplicable.

The second metric for SCF evenness was based on the standard deviation of the numbers of verb tokens among different SCF types (SD-based evenness):

$$Evenness\_SD = \sqrt{\frac{\sum_{i=1}^{R}(n_i - \bar{n})^2}{R-1}} \tag{6.5}$$

where n_i was the number of verb tokens for the i-th SCF type, and $\bar{n}$ was the average number of verb tokens across all SCF types. Opposite to the entropy-based SCF evenness, the SD-based evenness became lower when the SCFs were allocated more evenly among different SCF types. The SD-based evenness of the example text was

0.84. SD-based evenness also required the existence of more than two SCF types, otherwise, the denominator in the formula became zero.

SCF dispersion: SCF dispersion referred to the average distance between the verb tokens of the same SCF type. We calculated SCF dispersion as follows:

$$Disp = \frac{\sum_{j=1}^{R} \frac{\sum_{i=1}^{M_j-1} |position_{i,j} - position_{i+1,j}|}{M_j}}{R} \tag{6.6}$$

where $position_{i,j}$ referred to the position of the i-th verb token of the j-th SCF type which had $M_j (M_j >= 2)$ verb tokens in total. Note that this formula was inapplicable when every SCF type in the text had only one verb token. We used two kinds of positions: one was the word position in the text, and the other was the position among verbs. To illustrate, the word distance between $[\text{loved}]_{\text{dobj\_N}}$ and $[\text{find}]_{\text{dobj\_N}}$ was 9 (we calculated words by segmented units, and punctuation was considered as a segmented unit), while the verb distance was 1. The word-based SCF dispersion reflected the word distance between verb tokens, which was affected not only by how far away these verb tokens were located across different verbs, but also by the number of words of other syntactic categories, e.g., nouns, between the verbs. The verb-based SCF dispersion, by contrast, focused on only the relative distance by verbs.

SCF disparity: SCF disparity reflected the degree of taxonomic difference between the SCF types. Some SCF types were taxonomically closer than others. For example, "pobj" (a prepositional object) was similar to "pcomp" (a prepositional complement of which the dependent was a phrase or a clause) since they both involved a complement introduced by a preposition, whereas "dobj_N" was more different. We vectorized SCF types according to their complements, and calculated SCF disparity by analogy to ecological disparity (Novack-Gottshall, 2007). To vectorize SCF types, we classified the complements into eight major types: "acomp" (adjectival complement), "advmod" (adverbial complement), "ccomp" (clausal complement), "dobj" (direct object), "iobj" (indirect object), "pp" (prepositional object or complement), "prt" (particle), "xcomp" (non-finite complement). We further classified four subtypes for "ccomp" (i.e., "VBARE=>mark-that", "VTENSED", "VTENSED=>mark-that", "WHCOMP"), three subtypes for "pp" (i.e., "pobj", "pcomp", "pcomp_VING") and seven subtypes for "xcomp" (i.e., "N", "ADJ", "VBARE", "VBARE=>aux_TO", "VEN", "VING", "WHCOMP"). Each complement type and subtype became a dimension in a vector space. As a result, the vector space had 22 dimensions.

> The taxonomic distance between two SCF types was calculated as follows: first, the absolute value of the difference between the major complement type was calculated. If the two SCF types had the same major complement type that had some subtypes, the absolute value of the difference between the subtype dimensions, weighted by a parameter 0.25, was added to the total distance. For example, the difference between "dobj_N" and "ccomp_VTENSED" (a clausal complement headed by a finite verb) was 2, whereas the difference between "ccomp_VTENSED" and "ccomp_WHCOMP" (a wh-clausal complement) was 0.5.
>
> We calculated two SCF disparity metrics based on the taxonomic distance: the maximum and the average of the pairwise taxonomic distance between SCF types. To illustrate, the max-based disparity of the example text was 3, due to the taxonomic distance between "pobj:prt" (a preposition and a particle) and e.g., "su" (intransitive).

6.1.2 Controlling SCF diversity metrics for text length

The SCF diversity metrics were susceptible to text length. For example, as the text became longer, SCF TTR tended to decrease, because the number of SCF tokens increased whereas the increase of SCF types slowed down and stopped when the writer had used all the types he or she knew. To compare the SCF diversity of texts with different length, we controlled the SCF diversity metrics for text length.

We standardized each SCF metric by calculating its average over a moving window of a fixed number of verbs. For example, if we set the window size to be five verbs, the first window step for the example text spanned from the predicate *progressing* to the predicate *fell*. The window then moved by one verb, with the second step spanning from the predicate *moved* to the predicate *feeled* (a learner spelling error). The window moved until it reached the last predicate in the text. This standardization method was inspired by the calculation of mean moving-average type-token ratio (MATTR) for words (Covington and McFall, 2010). MATTR is more informative than the commonly used mean segment TTR (MSTTR) (Johnson, 1944) due to the following reason:

> MSTTR is computed on successive non-overlapping segments of the text whereas MATTR uses a smoothly moving window. Thus MATTR yields a value for every point in the text except for those less than one window length from the beginning, while MSTTR is only a stepwise approximation to this. Thus MATTR is better for tracking changes within texts, and MATTR is not affected by accidental interactions between segment boundaries... (Covington and McFall, 2010)

In our experiments, we standardized all SCF diversity metrics over the window sizes of 5, 10 and 20 verbs respectively. We never went beyond a text to get a larger window. When a text had fewer verbs than the window size, the standardized metrics were considered as inapplicable for the text (i.e., the text was excluded from analysis). Naturally, a larger window size applied to fewer texts. We avoided window sizes of more than 20 verbs because the number of applicable texts would be too small. The window size also determined other properties of standardized SCF diversity metrics. First, the window size corresponded to the size of linguistic unit for observation. A small window size might be close to the sentence level, whereas a large window size captured a piece of discourse. Second, a small window size made it easy for a standardized metric to "saturate", i.e., reach the maximum possible value, whereas a large window size resulted in the opposite. For example, it was easier to find completely different SCF types for 5 verbs than for 10 verbs. Third, a larger window size led to a finer granularity. For example, the SCF TTR for a window size of 5 verbs can take values of only 0.2, 0.4, 0.6, 0.8, and 1, which corresponded to 1, 2, 3, 4 and 5 SCF types within the window, whereas the SCF TTR for a window of 10 verbs can take values of 0.1, 0.2, 0.3,..., and 1.

For an SCF diversity metric that had requirements on the SCF distribution within a window, e.g., entropy-based SCF evenness required that the window of text had more than two SCF types, we considered only the window steps which met the requirement; if no window step met the requirement, the metric was considered as inapplicable for the text.

6.2 Data selection

We applied our SCF identifier to the whole EFCAMDAT, and calculated the SCF metrics for each text. The L2 proficiency was operationalized as the 16 proficiency levels of EFCAMDAT. To facilitate comparison between different dimensions of SCF diversity, we selected the texts on which all SCF metrics standardized at a window size were applicable, resulting in 508,192, 301,255 and 51,719 texts for the window sizes of 5, 10 and 20 verbs respectively. The three text groups are hereafter referred to as DAT5, DAT10 and DAT20 (see Table 6.1 for the detailed statistics of the datasets). While the number of applicable texts decreased as the window size increased, even the smallest dataset (DAT20) had more than 323 texts for each L2 proficiency level. The size of each group of applicable texts was large enough for statistical analysis.

Table 6.1 Distribution of words and texts in the learner datasets across L2 proficiency levels

Proficiency level	DAT5		DAT10		DAT20	
	# text	avg. # word	# text	avg. # word	text #	avg. # word
1	97,446	42	15,563	72	404	133
2	70,186	50	25,079	65	659	125
3	45,810	52	14,850	74	323	126
4	102,117	71	76,976	77	7,532	118
5	47,705	75	37,216	79	2,294	132
6	26,051	74	20,016	77	1,517	126
7	49,318	99	45,091	102	8,504	136
8	17,446	94	16,027	96	2,523	126
9	12,054	104	11,379	106	3,322	143
10	20,414	129	19,828	130	11,205	144
11	7,215	137	7,028	138	4,305	151
12	4,103	135	4,041	136	2,443	153
13	4,674	172	4,562	175	3,610	184
14	1,949	171	1,925	173	1,662	179
15	929	175	918	177	829	182
16	775	172	756	174	587	182
All	508,192	71	301,255	90	51,719	142

6.3 Statistical analysis methods

We explored the first research question by checking the scatter plots and the line graphs of the SCF diversity metrics versus L2 proficiency. When it turned out that there were linear relations between the SCF diversity metrics and L2 proficiency (see Section 6.4.1), we conducted correlation analysis. We then conducted multiple regression analysis to investigate how much the combined SCF metrics accounted for the variances in L2 proficiency level.

We studied the second research question by comparing our SCF metrics with current syntactic complexity metrics (Kyle, 2016) on EFCAMDAT, investigating how well these metrics predicted L2 proficiency, and whether the inclusion of SCF diversity metrics added to the accuracy of the prediction. The existing syntactic complexity metrics were extracted by TAASSC (Kyle, 2016). We used 14 traditional large-grained indices that were related to sentence length or clausal subordination (Lu, 2010), 32 clausal complexity indices, 132 phrasal complexity indices, and 38 syntactic sophistication indices for which the reference frequencies were calculated from the whole COCA.

Our procedure for conducting multiple regression analysis was as follows: first, we ensured a linear relation between each independent variable and the dependent variable by choosing the metrics that showed an absolute correlation of $|r| > 0.1$ (the threshold

for showing a small effect, Cohen, 1988) with L2 proficiency. Second, we prevented multicollinearity between the independent variables by conducting a pairwise correlation test on all the selected metrics. For each pair of metrics that had an absolute correlation of more than $|r| > 0.7$, we kept the metric that had the highest absolute correlation with L2 proficiency. We then conducted stepwise multiple regression analysis on the selected indices, setting the probability of F to enter at $p <= 0.05$ and the probability of F to remove at $p >= 0.1$. When the Variance Inflation Factor of a variable was higher than 5 (Rogerson, 2001), we removed that variable.

Note that the texts were distributed unevenly across L2 proficiency levels (Table 6.1). More specifically, each L2 proficiency level corresponded to 8 writing tasks and the texts were distributed unevenly across the writing tasks. In order to realize balanced contribution of residuals across different proficiency level in the statistical analyses, we weighted each data point by the inverse of the frequency of the writing task.

6.4 Results

In this section, we report the results of our experiments on EFCAMDAT with regard to the relation between SCF diversity and L2 proficiency, and whether SCF diversity metrics can contribute to the prediction of L2 proficiency beyond the existing syntactic complexity metrics.

6.4.1 SCF diversity metrics and L2 proficiency

Figure 6.1 shows how the mean (and its 95% confidence interval) of each SCF metric standardized at the window size of 5 verbs changed with L2 proficiency on DAT5. The relations between the repetition-based SCF diversity metrics and L2 proficiency were similar so we display only the relation between SCF TTR and L2 proficiency here. Also, the figures for the SCF diversity metrics standardized at other window sizes and/or applied to other datasets were similar. As we can see, there was a near-linear relation between each SCF diversity metric and L2 proficiency.

We then analyzed the correlation between the SCF diversity metrics and L2 proficiency. Table 6.2 shows the Pearson correlation between the SCF diversity metrics and L2 proficiency. All correlations were significant at the level of $p < 0.001$. We do not report standardized SCF size and standardized SCF richness because these metrics were the same as the window size and standardized SCF TTR respectively. Furthermore, the SCF richness, effective SCF richness, SCF repetition, and SCF disparity metrics standardized on a smaller window size

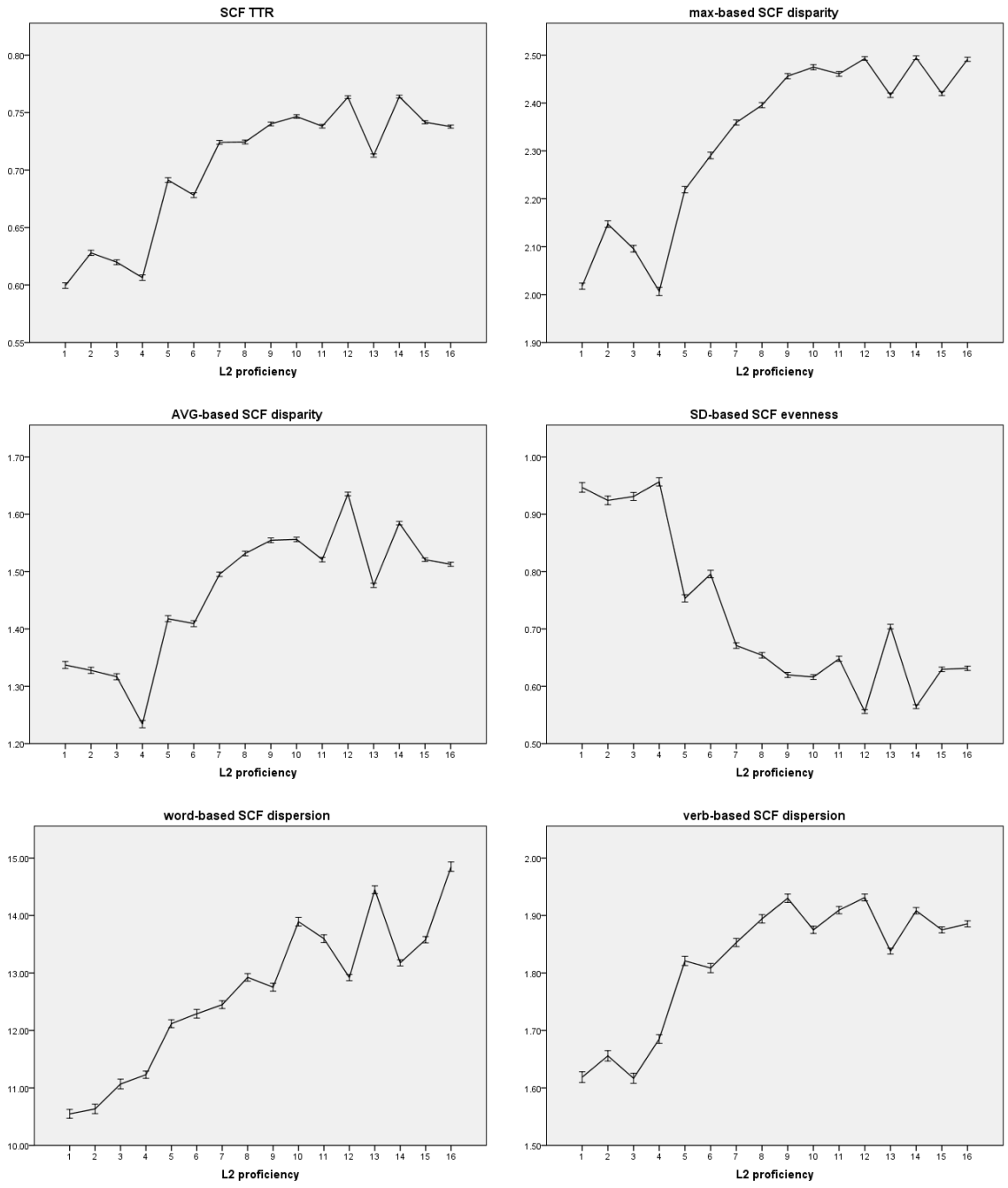


Fig. 6.1 Relation between the average of some SCF diversity metrics and L2 proficiency (DAT5)

can apply to the datasets prepared for a larger window size, because these metrics required a text to have verbs no fewer than the window size, and the datasets prepared for a larger window size automatically satisfied this requirement. However, SCF evenness and SCF dispersion metrics may not apply to a dataset prepared for a larger window size. This was because in addition to the same requirement of the other SCF metrics, SCF evenness metrics required the existence of at least two verb tokens for one SCF type in a window step, while SCF dispersion metrics required the existence of at least two SCF types in a window step. When these additional requirements were satisfied at a larger window, the requirements may not be satisfied at a smaller window. In our case, the SCF dispersion metrics standardized at the window size of 5 verbs were inapplicable to DAT10.

As we can see from Table 6.2, SCF richness, effective SCF richness and SCF repetition metrics showed positive medium correlation ($r > 0.3$) with L2 proficiency. This meant that more advanced learners tended to use more varied SCF types, or repeat SCF types less frequently.

The strength in reflecting the increase in L2 proficiency differed across different metrics. SCF TTR or SCF richness showed stronger correlation than entropy-based SCF effective richness, which in turn showed stronger correlation than GS-based SCF effective richness. This meant that the effective richness metrics calculated with higher weight on rare SCF types can better reflect the increase in L2 proficiency (See Equation 6.3). Furthermore, effective SCF richness metrics showed stronger correlation than their corresponding repetition indices.

Meanwhile, on DAT10 and DAT20, SCF TTR showed increasing correlation as the window size for standardization became larger. For example, on DAT20, SCF TTR standardized at the window size of 5 verbs showed a correlation at 0.315, whereas SCF TTR standardized at the window size of 20 verbs showed a higher correlation at 0.355. This might be attributed to the following factors: First, the SCF TTR or SCF richness of a larger linguistic might reflect the increase in L2 proficiency better. Second, a larger window size showed better correlation possibly due to their lower rate of saturation and finer granularity. Nevertheless, effective SCF richness and their related repetition metrics showed a decreasing correlation as the window size became larger. Since these metrics placed lower weight on rare SCF types than SCF richness or SCF TTR, this opposite trend indicated that the higher weight SCF TTR placed on rare SCF types was important for the increasing strength of a larger standardization window size for SCF TTR in reflecting L2 proficiency change.

SCF disparity also showed medium or close-to-medium positive correlation with L2 proficiency. This meant more advanced learners used SCFs that were taxonomically more different. Max-based SCF disparity showed stronger correlation with L2 proficiency than AVG-based SCF diversity. The former reached $r = 0.389$ when standardized by the window

Table 6.2 Correlation between standardized SCF diversity metrics and L2 proficiency

Metrics	Window size (# of verbs)	DAT5	DAT10	DAT20
SCF TTR / SCF richness	5	.368	.353	.315
	10	–	.357	.352
	20	–	–	.355
SCF entropy	5	.349	.339	.301
	10	–	.329	.322
	20	–	–	.317
SCF GS	5	.319	.317	.278
	10	–	.284	.276
	20	–	–	.247
Entropy-based effective SCF richness	5	.363	.346	.307
	10	–	.333	.325
	20	–	–	.313
GS-based effective SCF richness	5	.367	.342	.303
	10	–	.305	.295
	20	–	–	.250
Max-based SCF disparity	5	.364	.389	.369
	10	–	.376	.380
	20	–	–	.281
AVG-based SCF disparity	5	.275	.288	.267
	10	–	.257	.272
	20	–	–	.238
Entropy-based SCF evenness	5	.267	.259	.224
	10	–	.189	.185
	20	–	–	.141
SD-based SCF evenness	5	-.306	-.292	-.256
	10	–	-.240	-.235
	20	–	–	-.223
Word-based SCF dispersion	5	.248	–	.331
	10	–	.314	.310
	20	–	–	.290
Verb-based SCF dispersion	5	.183	–	.139
	10	–	.190	.153
	20	–	–	.159

size of 5 verbs and applied to DAT10. However, the correlation of max-based SCF diversity metrics with L2 proficiency dropped by almost 0.1 when they were standardized by the window size of 20 verbs rather than 10 verbs, which meant that the maximum taxonomic difference between SCF types at the larger window size was less indicative of the change in L2 proficiency. This might be related to the fact that in a much larger size of text, the chance of having taxonomically different SCF types was higher and the maximum taxonomic difference between these SCF types became more similar across different L2 proficiency levels.

As for SCF evenness, most metrics had positive small absolute correlation ($0.1 < |r| < 0.3$) with L2 proficiency, except the SD-based SCF evenness standardized at the window size of 5 verbs, which reached a positive correlation at 0.307. (Note again that a lower SD-based SCF evenness figure meant higher SCF evenness.) This meant more advanced learners used different SCF types more evenly than beginner learners. SD-based SCF evenness showed a stronger correlation than the entropy-based one. Moreover, on DAT10 and DAT20, the absolute correlation between all SCF evenness metrics and L2 proficiency decreased as the metrics were standardized on a larger window size. This meant that unlike SCF TTR, SCF evenness reflected the increase in L2 proficiency better at a smaller window size. Meanwhile, the absolute correlation of entropy-based SCF evenness dropped much more dramatically than SD-based SCF evenness as the window size grew. This meant that the ability of entropy-based SCF evenness to reflect the increase in L2 proficiency was much more sensitive to the window size.

We now come to SCF dispersion. Word-based SCF dispersion generally showed medium positive correlation with L2 proficiency, whereas verb-based SCF dispersion showed small positive correlation. This meant that more advanced learners located the verb tokens of the same SCF types further away from each other. The effect was more obvious when the distance was evaluated by words rather than verbs, which meant that advanced learners also used more words between the verbs, a result in line with the previous findings that the mean length of utterance increased with proficiency. This result could also reflect the findings that more advanced learners used more elaborate noun phrases, e.g., using more modifiers for nouns (Biber et al., 2011; Kyle, 2016; Taguchi et al., 2013).

6.4.2 Comparing SCF diversity and current syntactic complexity measures in predicting L2 proficiency

We conducted multiple regression analysis on the SCF diversity metrics, current syntactic complexity (SC) metrics, and the combination of both to predict L2 proficiency on each

Table 6.3 Model statistics of multiple regression analysis

Metrics	DAT5		DAT10		DAT20	
	# MEM	Adjust r^2	# MEM	Adjust r^2	# MEM	Adjust r^2
SCF diversity	4	.188	4	.203	5	.251
Current SC	61	.660	57	.633	59	.534
Both	60+4[a]	.667	56+3	.639	60[b]+5	.542

[a] The number of current SC metrics + the number of SCF diversity metrics.

[b] While 59 current SC metrics alone entered the modal for DAT20, 60 current SC metrics entered the model when the SCF diversity metrics were included. This was because we used stepwise multiple regression analysis, which selected a different set of current SC metrics when the SCF diversity metrics were also considered.

dataset. Table 6.3 shows the number of metrics entering each model (# MEM) after we pruned the metrics following the procedure in Section 6.3, and how well the models predicted L2 proficiency.

As we can see, SCF diversity metrics can explain from 18.8% to 25.1% of the variance in L2 proficiency. Furthermore, while the numerous current syntactic complexity metrics already explained a large proportion of the variance in L2 proficiency, SCF diversity metrics can complement current syntactic complexity metrics in the prediction, improving the model effect by 0.5% to 0.8%. This proved that the SCF diversity metrics contributed a unique perspective to syntactic complexity which had not been captured by current syntactic complexity metrics.

Moreover, the SCF diversity metrics that always entered a model were SCF TTR, max-based SCF disparity, and word-based SCF dispersion. These metrics represent the most important aspects that SCF diversity can contribute to the prediction of L2 proficiency.

6.5 Summary

In this chapter, we illustrated the usefulness of the SCF identification system with an L2 linguistic research based on a large-scale learner corpus. We proposed the first SCF diversity metrics, and investigated how SCF diversity changed with L2 development. Our answers to the research questions posed at the beginning of this chapters were as follows.

1. How does SCF diversity change with English L2 proficiency in general?

We found that as their L2 proficiency developed, learners tended to use more diverse SCF types which were taxonomically more different from each other. Also, more advanced learners tended to use different SCF types more evenly, and locate the verb tokens of the same SCF type further away from each other.

2. Does SCF diversity contribute to the prediction of English L2 proficiency beyond current syntactic complexity measures?

We empirically showed that the SCF diversity metrics made a unique contribution to the measurement of syntactic complexity. We also found that the design of the SCF metrics and the standardization window affected the ability of the SCF metrics in reflecting the increase of L2 proficiency.

These empirical findings on SCF diversity not only shed light on L2 SCF acquisition, but also provided useful implications for L2 education. For example, L2 educators can consider the factor of SCF diversity in L2 assessment and curriculum development. SCF diversity can also be useful for developing educational NLP applications such as automatic essay scoring and intelligent language tutoring systems.

The empirical study demonstrated the power of the SCF identification system in helping researchers to gain insights into L2 SCF acquisition from large-scale learner corpora. Based on the automatical analysis of SCFs, L2 researchers can investigate how lexical and syntactic knowledge develops in L2 acquisition and look into other questions such as whether and how SCF use change across different first language backgrounds, and whether there is task effect on SCF use in L2 writing, etc. The SCF identification system can also serve as a preprocessing technique for finding or filtering useful patterns for linguistic research.

Furthermore, the SCF system can facilitate the development of NLP applications which involve SCF information. For example, SCFs can be useful for automatic summarization, which requires discrimination between complements and adjuncts to decide what information is important for the summary. SCFs can also be useful for semantic role labeling. Furthermore, our SCF system was adapted to learner English, and can be useful for NLP applications that process non-native English data, such as native language identification.

Chapter 7

Conclusion

In this concluding chapter, we summarize the contributions of the thesis and outline directions for future research.

7.1 Contributions of the thesis

In the automatic analysis of learner language, researchers mainly use standard POS taggers and parsers developed for native language to analyze learner language. There has been the need to investigate the performance of such systems on learner language, and to develop an SCF identification system for analyzing the SCFs of learner language. Previous studies had evaluated the accuracy of some standard POS taggers or parsers on learner English, and investigated the effect of some learner errors on POS tagging. However, these studies obtained the gold standards by manually correcting the output of a system, probably introducing bias to the evaluations. Meanwhile, more comprehensive research was needed to support the development of strategies for minimizing the cross-domain effects, such as evaluating how fine-grained learner errors influence the performance of standard parsers on learner English, comparing the performance of multiple parsers on learner English, and investigating the relation between the performance of a parser on native language and learner language.

As for automatic SCF analysis, previous research had developed many NLP systems regarding SCFs. However, most systems were intended to acquire SCF lexicons, and cannot identify SCFs for individual verb tokens. Meanwhile, all previous systems were developed for native language, producing results which can be inaccurate for learner language. An SCF identification system adapted for learner language was needed for L2 SCF research and downstream NLP applications.

This thesis fulfilled the aforementioned research gaps. First, we provided an in-depth evaluation of how standard POS taggers and dependency parsers performed on learner English.

Second, we developed an SCF identification system for learner English. We demonstrated the usefulness of the SCF identification system in linguistic research by investigating how SCF diversity developed in L2 acquisition. Our work resulted in various experimental findings and methodological proposals which we summarize as follows.

1. Annotation bias in creating gold standard syntax

The evaluation of syntactic analysis systems required human annotation of gold standard syntax. Previous parser evaluations commonly obtained gold standards by manually correcting the output of a parser. We empirically demonstrated that such an annotation method can lead to annotation bias, which significantly influenced the result of parser evaluation in favor of the pre-annotation parser. More specifically, the annotation bias reduced the recall of parsing errors during annotation. Our analyses showed that the annotation bias arose from the inherent ambiguity of some linguistic structures, the annotation schemes, and learner errors.

Our annotation experiment suggested an effective way to reduce the annotation bias – contrast-based annotation, wherein the annotation mismatches of several parsers were displayed to annotators. We also found that the effectiveness of the reference provided by a parser depended on its accuracy. Nevertheless, the marginal benefit of adding a reference parser diminished as the number of reference parsers increased, because the correct references provided by the parsers may overlap whereas the reference from the additional parser needed more time to review. Therefore, the contrast-based annotation method requires one to strike a balance between the reduction of annotation bias and the maintenance of annotation efficiency.

Our investigation into the causes of the annotation bias also suggested that the annotation bias can be reduced by improving the annotation scheme for parsing. Clearer distinctions need to be made for some intermediate syntactic categories, such as in what situations the present and past particle of verbs should be regarded as adjectives rather than verbs. Furthermore, our results indicated that nominal modifiers, prepositional phrases, infinitive clauses, and conjuncts were difficult cases that frequently gave rise to annotation bias. As a result, special attention is required to distinguish the syntactic relations involving these structures.

Our findings and suggestions on annotation may be extensible to the annotation of any other information that involves automatic analysis systems. For example, when annotating medical entities based on the result of a named-entity recognizer, annotation bias may also arise. In such situations, our suggestions on the contrast-based annotation method and improving annotation schemes may also be useful for reducing the annotation bias.

2. The performance of standard parsers on learner English

We found that on average, current standard parsers achieved around 95% on POS accuracy, 90% on UAS, 87% on LAS, and 85% on the accuracy of all tags on learner English. The performance gaps between these figures and the accuracy of the standard parsers on native English were 1.2%, 2.1%, 3.5% and 3.4 % respectively. Meanwhile, the performance gaps between different parsers were smaller on POS tags than on dependency relations. This meant that the performance of different parsers on POS tags was higher and more similar than on dependency relations.

We quantitatively investigated the relations between fine-grained learner errors and parsing errors. Our results showed that learner errors did have an impact on parsing output. More than one-third of the parsing errors were caused by learner errors, and over 60% of the learner errors caused at least one parsing error. These results indicated that the parsers were not very robust to learner errors. Our analyses showed that learner errors on punctuation, spelling, capitalization, argument structures, determiners and prepositions caused most parsing errors. Correcting these learner errors can be an effective pre-processing technique to reduce parsing errors for downstream linguistic research and NLP applications based on learner English data.

While the large impact of learner errors on parsing seemed to contradict the small performance gaps between the accuracy scores of the parsers on learner English and native English, we empirically testified that this was because the impact of learner errors was offset by the simplicity of learner language. Parsers performed better on shorter sentences, and the average sentence length of learner English was shorter than that of native English. Furthermore, not every learner sentence contained learner errors, and when a sentence did contain a learner error, it affected only the parses of a limited number of words in the sentence. The accurate parses of short learner sentences that had no learner errors helped maintain a high face value of the parsing accuracy for learner English.

Finally, we demonstrated that the performance of standard probabilistic parsers on learner English can be predicted by their performance on native English. The implication was that when it comes to choosing a probabilistic parser for learner English, the most accurate parser evaluated on native English is a good choice. Alternatively, if one wants to apply a probabilistic parser to a specific learner English dataset, he or she can roughly predict the accuracy of the parser on learner English according to its accuracy on native English.

While our evaluation was conducted on learner data, our conclusions and implications may apply to the syntactic analysis of non-standard data in general. For example, recent years have seen increasing interests in developing NLP applications based on language data from social media, where many users are non-native speakers and the language is informal

(Rijhwani et al., 2017). Our results on the most influential language errors for parsing and the correlation between the performance of standard parsers on learner data and native English data may provide useful implications on how to choose appropriate preprocessing techniques and proper parsers for such non-standard data.

3. SCF identification system for learner English

We developed the first SCF identification system for learner English. The system can label individual occurrences of verbs in learner corpora for a set of 49 distinct SCFs ranging from basic transitive and intransitive frames to complicated frames that involve prepositional, verbal or clausal complements. The system includes a MaxEnt model based on features of words, POS tags, dependency relations, and word embeddings. We adapted the model to learner English by training the model on learner English data, and improved the accuracy of the model by including general-domain native English training data. Our 10-fold cross-validation showed that the system achieved an accuracy of 84.2%. This level of accuracy was among the highest reported among contemporary systems and was likely to be sufficient for benefit in downstream tasks.

Our development of the SCF identification system provided useful implications on linguistics, the annotation of non-native language data and the development of NLP techniques for such data. First, the usefulness of our features for the model proved that words, syntactic categories, syntactic relations, and distributional semantics were all important for the identification of SCFs. Second, our annotation practice demonstrated that native language schemes can be used to annotate non-native language data, as our native English SCF inventory proved to cover 99.4% of the learner SCFs. Third, our results demonstrated that for machine learning on a small training dataset, a simple MaxEnt model can perform better than deep neural network models. Furthermore, we showed that when the non-native dataset was small, including native language datasets can help to improve the accuracy of the classification.

4. SCF diversity metrics in L2 research

To illustrate the usefulness of the SCF identification system, we proposed the first multi-dimensional SCF diversity metrics and investigated how SCF diversity changed with L2 development. Our results shed interesting light on L2 SCF acquisition: We found that more advanced learners tended to use more diverse SCF types which were taxonomically more different from each other. Meanwhile, more advanced learners tended to use different SCF types more evenly, and locate the verb tokens of the same SCF type further away from each other.

We also empirically showed that the proposed SCF diversity metrics can be a useful measure of linguistic complexity, at the interface between lexicon and syntax. The SCF diversity metrics can improve the prediction of L2 proficiency on top of existing syntactic complexity metrics. Furthermore, we identified how the design of the SCF metrics and the standardization window affected the ability of the SCF metrics in predicting L2 proficiency: SCF TTR, max-based SCF disparity, and word-based SCF dispersion represented the most important aspects that SCF diversity can contribute to the prediction of L2 proficiency; SCF TTR showed an increasing correlation with L2 proficiency as the window size for standardization became larger, while max-based SCF diversity metrics showed the opposite trend. This provided important implications on how to choose SCF diversity metrics and appropriate standardization units to gauge L2 development.

The SCF diversity metrics can be useful for L2 education. For example, L2 educators can consider the factor of SCF diversity in L2 assessment and curriculum development. SCF diversity may also be useful for developing educational NLP applications such as automatic essay scoring and intelligent language tutoring systems.

Our linguistic research demonstrated the power of the SCF identification system in helping researchers to gain insights into L2 SCF acquisition from large-scale learner corpora. Based on the automatical analysis of SCFs, L2 researchers can investigate how lexical and syntactic knowledge develops in L2 acquisition and look into other questions such as whether and how SCF use change across different first language backgrounds, and whether there is task effect on SCF use in L2 writing, etc. The SCF identification system can also serve as a preprocessing technique for finding or filtering useful patterns for linguistic research.

Furthermore, the SCF system can be useful for NLP applications which involve SCF information. For example, automatic summarization (Cheung and Penn, 2014) requires discrimination between complements and adjuncts to decide what information is important for the summary; SCFs are also closely related to semantic role labeling (Roth and Lapata, 2016). Meanwhile, our SCF system was adapted to learner English, and can be useful for NLP applications that process non-native English data, such as native language identification (Jiang et al., 2018).

7.2 Directions for future research

We identified some directions for future research as follows.

1. Evaluation of standard parsers on learner English

This thesis investigated how fine-grained learner errors influenced the performance of standard parsers on learner essays that spanned across the full learner English proficiency spectrum. Further research can be conducted to investigate how standard parsers are impacted by learner errors at each proficiency level and how the impact changes across these levels. This information will be useful for downstream linguistic research and NLP applications. For example, syntactic complexity metrics have been used to gauge L2 proficiency levels, and the automatic analysis of these metrics rely on standard parsers (Kyle, 2016; Lu, 2010). If the impact of the learner errors on standard parsers changes dramatically with proficiency levels, the efficacy of the automatically analyzed syntactic complexity metrics in predicting L2 proficiency will be undermined (Meurers and Dickinson, 2017).

2. Improving SCF identification

Further research can be conducted to improve the accuracy of SCF identification. While this thesis focused on a supervised training method, future research can look into semi-supervised methods which can utilize a large amount of un-annotated data. More specifically, it will be interesting to investigate how to improve the identification of relatively rare SCFs. Meanwhile, our error analysis of the SCF system showed that most errors made by our SCF identification system were related to prepositional phrases. Future research can investigate solutions to this problem. Moreover, we can explore multi-task learning (Collobert and Weston, 2008) which combines SCF identification with potentially relevant NLP tasks such as prepositional attachment disambiguation (Gelbukh and Calvo, 2018) and semantic role labeling (Roth and Lapata, 2016) during training.

3. Development of SCF complexity metrics

In this thesis, we developed some SCF diversity metrics, focusing on the breadth aspect of linguistic complexity. Future research can be conducted to develop SCF-related linguistic complexity metrics from the depth aspect. For example, we can develop SCF sophistication metrics following the design of VAC sophistication metrics (VAC includes all dependents of a predicate verb) (Kyle, 2016), and compare the two sets of metrics, investigating how the discrimination between complements and adjuncts affects the properties of the metrics. Meanwhile, we standardized the SCF diversity metrics by verb windows. It will be interesting to investigate whether other standardization units, such as word windows, will be a better unit in some situations.

4. Downstream applications of the SCF system and the SCF diversity metrics

Our SCF identification system opens up a lot of opportunities for linguistic research, language education, and NLP applications involving SCFs. For linguistic research, L2 researchers can investigate how lexical and syntactic knowledge develops in L2 acquisition based on the automatically analyzed SCFs. Researchers can also investigate how SCF use changes across different L1 backgrounds, and whether there is any L1 transfer on L2 SCF use from the typological aspect. Furthermore, task effects are widely recognized as an important aspect of learner language analysis (Alexopoulou et al., 2017), and it will be interesting to investigate how writing tasks affect SCF use. For education, researchers can include SCFs or SCF diversity into the design of course materials and language assessment. For NLP applications, researchers can investigate whether SCFs or SCF diversity metrics can be useful features or a sub-task in a joint-learning scenario for NLP applications such as automatic summarization and semantic role labeling.

5. Extension to other learner languages

This thesis focused on automatic syntactic analysis for learner English. Future research can be conducted on other learner languages, investigating whether the conclusions we have reached on learner English also hold for other learner languages. For example, how does SCF diversity change across the different proficiency levels of other L2s? This requires the construction of other resources, research into relevant NLP techniques and corresponding linguistic studies.

In summary, the advances reported in our thesis contribute to the automatic analysis of learner English, making it possible to design strategies for improving the performance of standard parsers on learner English, and identify SCFs for large-scale learner English efficiently. Our experiments also provide useful implications on the annotation and modeling of small non-standard datasets in general. The advances in our thesis bring in the potential to improve L2 research, education, and NLP applications.

References

Aarts, J., Van Halteren, H., and Oostdijk, N. (1998). The linguistic annotation of corpora: The TOSCA analysis system. *International Journal of Corpus Linguistics*, 3(2):189–210.

Al-Rfou, R., Perozzi, B., and Skiena, S. (2013). Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia. ACL.

Alexopoulou, T., Michel, M., Murakami, A., and Meurers, D. (2017). Task effects on linguistic complexity and accuracy: A large-scale learner corpus analysis employing natural language processing techniques. *Language Learning*, 67(S1):180–208.

Allen, J. (1995). *Natural language understanding*. New York: Pearson.

Altamirano, I. R. (2010). IRASubcat, a highly customizable, language independent tool for the acquisition of verbal subcategorization information from corpus. In *Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas*, pages 84–91, Uppsala. ACL.

Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., and Collins, M. (2016). Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2442–2452, Berlin. ACL.

Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. (2016). Learning to compose neural networks for question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1545–1554, San Diego. ACL.

Aston, G. and Burnard, L. (1998). *The BNC handbook: Exploring the British National Corpus with SARA*. Edinburgh: Edinburgh University Press.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baker, S., Reichart, R., and Korhonen, A. (2014). An unsupervised model for instance level subcategorization acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 278–289, Dora. ACL.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Berzak, Y., Huang, Y., Barbu, A., Korhonen, A., and Katz, B. (2016a). Anchoring and agreement in syntactic annotations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2215–2224, Austin. ACL.

Berzak, Y., Kenney, J., Spadine, C., Wang, J. X., Lam, L., Mori, K. S., Garza, S., and Katz, B. (2016b). Universal dependencies for learner English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 737–746, Berlin. ACL.

Biber, D. (1988). *Variation across speech and writing*. New York: Cambridge University Press.

Biber, D., Gray, B., and Poonpon, K. (2011). Should we use characteristics of conversation to measure grammatical complexity in L2 writing development? *TESOL Quarterly*, 45(1):5–35.

Bird, S. and Loper, E. (2004). NLTK: The natural language toolkit. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, page 31, Barcelona. ACL.

Bley-Vroman, R. (1989). What is the logical problem of foreign language learning. In Gass, S. M. and Schachter, J., editors, *Linguistic perspectives on second language acquisition*, pages 41–68. New York: Cambridge University Press.

Bley-Vroman, R. and Joo, H.-R. (2001). The acquisition and interpretation of English locative constructions by native speakers of Korean. *Studies in Second Language Acquisition*, 23(2):207–219.

Bley-Vroman, R. and Yoshinaga, N. (1992). Road and narrow constraints on the English dative alternation: Some fundamental differences between native speakers and foreign language learners. *University of Hawai'i Working Papers in English as a Second Language 11 (1)*.

Boguraev, B. and Briscoe, T. (1987). Large lexicons for natural language processing: Utilising the grammar coding system of LDOCE. *Computational Linguistics*, 13(3-4):203–218.

Bohnet, B. and Nivre, J. (2012). A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465, Jeju Island. ACL.

Bolinger, D. (1980). *Syntactic diffusion and the definite article*. Bloomington: Indiana University Linguistics Club.

Brent, M. R. (1991). Automatic acquisition of subcategorization frames from untagged text. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 209–214, Berkeley. ACL.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 152–155, Trento. ACL.

Brinker, K. (1972). *Konstituentenstrukturgrammatik und operationale Satzgliedanalyse. Methodenkritische Untersuchungen zur Syntax des einfachen Satzes im Deutschen*. Frankfurt am Main: Athenaum.

Briscoe, T. and Carroll, J. (1997). Automatic extraction of subcategorization from corpora. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 356–363, Washington, D.C. ACL.

Brown, R. (1973). *A first language: The early stages*. Cambridge, MA: Harvard University Press.

Bulté, B. and Housen, A. (2012). Defining and operationalising L2 complexity. In Housen, A., Kuiken, F., and Vedder, I., editors, *Dimensions of L2 performance and proficiency: Complexity, accuracy and fluency in SLA*, pages 21–46. Philadelphia: John Benjamins.

Cai, Z. G. and Liu, H. (2017). Microscopic and macroscopic approaches to the mental representations of second languages. *Behavioral and Brain Sciences*, 40:e285.

Cer, D. M., De Marneffe, M.-C., Jurafsky, D., and Manning, C. D. (2010). Parsing to Stanford dependencies: Trade-offs between speed and accuracy. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 1628–1632, Valletta. ELRA.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 173–180, Stroudsburg. ACL.

Chesley, P. and Salmon-Alt, S. (2006). Automatic extraction of subcategorization frames for French. In *Proceedings of the Language Resources and Evaluation Conference*, Genoa. ELRA.

Cheung, J. C. K. and Penn, G. (2014). Unsupervised sentence enhancement for automatic summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 775–786, Doha. ACL.

Choi, J. D. (2016). Dynamic feature induction: The last gist to the state-of-the-art. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 271–281, San Diego. ACL.

Choi, Y. and Cardie, C. (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 793–801, Honolulu. ACL.

Chomsky, N. (1957). *Syntactic structures*. Berlin: Mouton.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT press.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. Hilsdale: Lawrence Earlbaum Associates.

Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of the Eighth Conference on European chapter of the Association for Computational Linguistics*, pages 16–23, Madrid. ACL.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167, Helsinki. ACM.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

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

Covington, M. A. and McFall, J. D. (2010). Cutting the Gordian Knot: The Moving-Average Type–Token Ratio (MATTR). *Journal of Quantitative Linguistics*, 17(2):94–100.

Croft, W. (1991). *Syntactic categories and grammatical relations: The cognitive organization of information*. London: University of Chicago Press.

Crosthwaite, P. (2016). A longitudinal multidimensional analysis of EAP writing: Determining EAP course effectiveness. *Journal of English for Academic Purposes*, 22:166–178.

Culicover, P. (1982). *Syntax*. New York: Academic Press.

Dandapat, S., Biswas, P., Choudhury, M., and Bali, K. (2009). Complex linguistic annotation–No easy way out! A case from Bangla and Hindi POS labeling tasks. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 10–18, Suntec. ACL.

Das, D. and Petrov, S. (2011). Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 600–609, Portland. ACL.

Davidson, T. (1874). The grammar of dionysios thrax. *The Journal of Speculative Philosophy*, 8(4):326–339.

Davies, M. (2008). *The corpus of contemporary American English*. Provo: Brigham Young University.

De Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, pages 449–454, Genoa. ELRA.

De Marneffe, M.-C. and Manning, C. D. (2008a). Stanford typed dependencies manual. Technical report, Stanford University.

De Marneffe, M.-C. and Manning, C. D. (2008b). The Stanford typed dependencies representation. In *COLING 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester. ACL.

Dębowski, Ł. (2009). Valence extraction using EM selection and co-occurrence matrices. *Language Resources and Evaluation*, 43(4):301–327.

Díaz-Negrillo, A., Meurers, D., Valera, S., and Wunsch, H. (2010). Towards interlanguage POS annotation for effective learner corpora in SLA and FLT. *Language Forum*, 36(1–2):139–154.

Dickinson, M. and Lee, C. M. (2013). Modifying corpus annotation to support the analysis of learner language. *CALICO Journal*, 26(3):545–561.

Dickinson, M. and Ragheb, M. (2009). Dependency annotation for learner corpora. In *Proceedings of the Eighth Workshop on Treebanks and Linguistic Theories*, pages 59–70, Milan. EDUCatt.

Dickinson, M. and Ragheb, M. (2015). On grammaticality in the syntactic annotation of learner language. In *Proceedings of the 9th Linguistic Annotation Workshop*, pages 158–167, Denver. ACL.

Dušek, O., Hajic, J., and Uresova, Z. (2014). Verbal valency frame detection and selection in Czech and English. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 6–11, Baltimore. ACL.

Ejerhed, E., Källgren, G., Wennstedt, O., and Åström, M. (1992). The linguistic annotation system of the stockholm-umeå corpus project-description and guidelines. Technical report, Department of Linguistics, Umeå University.

Ellis, N. C. and Ferreira–Junior, F. (2009). Construction learning as a function of frequency, frequency distribution, and function. *The Modern Language Journal*, 93(3):370–385.

Ellis, N. C., O'Donnell, M. B., and Römer, U. (2014). Second language verb-argument constructions are sensitive to form, function, frequency, contingency, and prototypicality. *Linguistic Approaches to Bilingualism*, 4(4):405–431.

Ellis, R. (1994). *The study of second language acquisition*. Oxford: Oxford University Press.

Emons, R. (1974). *Valenzen englischer Prädikatsverben*. Linguistische Arbeiten (22). Tübingen: Niemeyer.

Engel, U. and Schumacher, H. (1978). *Kleines Valenzlexikon deutscher Verben*. Forschungsberichte des Instituts für deutsche Sprache (31). Tübingen: Narr.

Evans, N. and Levinson, S. C. (2009). The myth of language universals: Language diversity and its importance for cognitive science. *Behavioral and brain sciences*, 32(5):429–448.

Fitzpatrick, E. and Seegmiller, M. S. (2004). The Montclair electronic language database project. *Language and Computers*, 52(1):223–237.

Fort, K. and Sagot, B. (2010). Influence of pre-annotation on POS-tagged corpus development. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 56–63, Uppsala. ACL.

Foth, K. A. and Menzel, W. (2006). Hybrid parsing: Using probabilistic models as predictors for a symbolic parser. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 321–328, Sydney. ACL.

Francis, G., Hunston, S., and Manning, E. (1996). *Collins COBUILD grammar patterns, 1: Verbs*. London: Harper Collins.

Francis, W. N. (1964). A standard sample of present-day English for use with digital computers. Technical report, Department of Linguistics, Brown University.

Garside, R., Leech, G., and McEnery, A. (1997). A hybrid grammatical tagger: CLAWS 4. In Garside, R., editor, *Corpus annotation: Linguistic information from computer text corpora*, pages 102–121. Essex: Addison Wesley Longman.

Garside, R., Sampson, G., and Leech, G. (1988). *The computational analysis of English: A corpus-based approach*. London: Longman.

Gazdar, G. (1985). *Generalized phrase structure grammar*. Cambridge MA: Harvard University Press.

Geertzen, J., Alexopoulou, T., and Korhonen, A. (2013). Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge Open Language Database (EFCAMDAT). In *Proceedings of the 31st Second Language Research Forum: Building Bridges Between Disciplines*, Somerville. Cascadilla Proceedings Project.

Gelbukh, A. and Calvo, H. (2018). Prepositional phrase attachment disambiguation. In *Automatic syntactic analysis based on selectional preferences*, pages 85–110. Cham: Springer.

Gerz, D., Vulić, I., Hill, F., Reichart, R., and Korhonen, A. (2016). SimVerb-3500: A large-Scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin. ACL.

Gilquin, G. and Gries, S. T. (2009). Corpora and experimental methods: A state-of-the-art review. *Corpus linguistics and linguistic theory*, 5(1):1–26.

Giménez, J. and Marquez, L. (2004). SVMTool: A general POS tagger generator based on Support Vector Machines. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon. ELRA.

Goldberg, A. E. (1995). *Constructions: A construction grammar approach to argument structure*. Chicago: University of Chicago Press.

Goldberg, A. E. (1999). The emergence of the semantics of argument structure constructions. In MacWhinney, B., editor, *The emergence of language*, pages 215–230. Mahwah: Lawrence Erlbaum Associates.

Goldberg, A. E. (2006). *Constructions at work: The nature of generalization in language*. Oxford: Oxford University Press.

Graesser, A. C., McNamara, D. S., and Kulikowich, J. M. (2011). Coh-Metrix: Providing multilevel analyses of text characteristics. *Educational researcher*, 40(5):223–234.

Granger, S., Dagneaux, E., Meunier, F., and Paquot, M. (2009). *The International Corpus of Learner English. Version 2. Handbook and CD-ROM*. Louvain-la-Neuve: Presses Universitaires de Louvain.

Greenberg, J. H. (2005). *Language universals: With special reference to feature hierarchies*. Berlin: Mouton de Gruyter.

Greene, B. B. and Rubin, G. M. (1971). Automatic grammatical tagging of English. Technical report, Department of Linguistics, Brown University.

Gries, S. T. and Berez, A. L. (2017). Linguistic annotation in/for corpus linguistics. In Ide, N. and Pustejovsky, J., editors, *Handbook of linguistic annotation*, pages 379–409. Berlin: Springer.

Gries, S. T. and Wulff, S. (2005). Do foreign language learners also have constructions? *Annual Review of Cognitive Linguistics*, 3(1):182–200.

Grishman, R., Macleod, C., and Meyers, A. (1994). COMLEX syntax: Building a computational lexicon. In *Proceedings of the 15th Conference on Computational Linguistics-Volume 1*, pages 268–272, Kyoto. ACL.

Han, X., Zhao, T., Qi, H., and Yu, H. (2004). Subcategorization acquisition and evaluation for Chinese verbs. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 723–728, Geneva. ACL.

Hardie, A. (2003). Developing a tagset for automated part-of-speech tagging in Urdu. In *Proceedings of the Corpus Linguistics 2003 Conference*, Lancaster. Department of Linguistics, Lancaster University.

Hatori, J., Matsuzaki, T., Miyao, Y., and Tsujii, J. (2011). Incremental joint POS tagging and dependency parsing in Chinese. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1216–1224, Chiang Mai. ACL.

Heine, B. (1975). Language typology and convergence areas in Africa. *Linguistics*, 13(144):27–48.

Heine, B. and Reh, M. (1984). *Grammaticalization and reanalysis in African languages*. Hamburg: Helmet Buske Verlag.

Helbig, G. and Schenkel, W. (1991). *Wörterbuch zur Valenz und Distribution deutscher Verben*. Leipzig: VEB Bibliographisches Institut.

Herbst, T. (1984). Adjective complementation: A valency approach to making EFL dictionaries. *Applied linguistics*, 5(1):1–11.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Housen, A., Kuiken, F., and Vedder, I. (2012). *Dimensions of L2 performance and proficiency: complexity, accuracy and fluency in SLA*. Philadelphia: John Benjamins.

Housen, A. and Simoens, H. (2016). Introduction: Cognitive perspectives on difficulty and complexity in L2 acquisition. *Studies in Second Language Acquisition*, 38(2):163–175.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Hudson, R. (1980a). A second attack on constituency: A reply to Dahl. *Linguistics*, 18(5-6):489–504.

Hudson, R. (1984). *Word grammar*. Oxford: Blackwell.

Hudson, R. A. (1980b). Constituency and dependency. *Linguistics*, 18(3-4):179–198.

Hunt, K. W. (1970). Do sentences in the second language grow like those in the first? *TESOL Quarterly*, pages 195–202.

Ienco, D., Villata, S., and Bosco, C. (2008). Automatic extraction of subcategorization frames for Italian. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, pages 2094–2100, Marrakech. ELRA.

Inagaki, S. (1997). Japanese and Chinese learners' acquisition of the narrow-range rules for the dative alternation in English. *Language Learning*, 47(4):637–669.

Jackendoff, R. (1992). *Semantic structures*. Cambridge, MA: MIT press.

James, C. (2013). *Errors in language learning and use: Exploring error analysis*. New York: Addison Wesley Longman.

Jarvis, S. (2013). Capturing the diversity in lexical diversity. *Language Learning*, 63(s1):87–106.

Jespersen, O. (1924). *The philosophy of grammar*. London: Allen and Unwin.

Jiang, X., Huang, Y., Guo, Y., Geertzen, J., Alexopoulou, T., Sun, L., and Korhonen, A. (2018). Native language identification on efcamdat. In Poibeau, T. and Villavicencio, A., editors, *Language, Cognition, and Computational Models*, pages 159–184. Cambridge: Cambridge University Press.

Johansson, R. and Nugues, P. (2007). Extended constituent-to-dependency conversion for English. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 105–112, Tartu. University of Tartu.

Johnson, W. (1944). Studies in language behavior: I. A program of research. *Psychological Monographs*, 56(2):1–15.

Joo, H.-R. (2003). Second language learnability and the acquisition of the argument structure of English locative verbs by Korean speakers. *Second Language Research*, 19(4):305–328.

Jost, L. (2006). Entropy and diversity. *Oikos*, 113(2):363–375.

Juffs, A. (1996). Semantics-syntax correspondences in second language acquisition. *Second Language Research*, 12(2):177–221.

Juffs, A. (1998). Some effects of first language argument structure and morphosyntax on second language sentence processing. *Second Language Research*, 14(4):406–424.

Kaplan, R. M., Bresnan, J., et al. (1982). Lexical-functional grammar: A formal system for grammatical representation. *Formal Issues in Lexical-Functional Grammar*, (47):29–130.

Kawahara, D. and Kurohashi, S. (2010). Acquiring reliable predicate-argument structures from raw corpora for case frame compilation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 1389–1393, Valletta. ELRA.

Khoja, S. (2001). APT: Arabic part-of-speech tagger. In *Proceedings of the NAACL Student Research Workshop*, pages 20–25, Pittsburgh.

Kim, H., Hwang, H., and Rah, Y. (2017). Young EFL students' reliance on path-breaking verbs in the use of English argument structure constructions. *Journal of Cognitive Science*, 18(3):341–366.

Kim, H. and Rah, Y. (2016). Effects of verb semantics and proficiency in second language use of constructional knowledge. *The Modern Language Journal*, 100(3):716–731.

Klein, D. and Manning, C. D. (2003a). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics-Volume 1*, pages 423–430, Sapporo. ACL.

Klein, D. and Manning, C. D. (2003b). Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15*, pages 3–10, Cambridge, MA. MIT Press.

Kong, L. and Smith, N. A. (2014). An empirical comparison of parsing methods for Stanford dependencies. *arXiv preprint arXiv:1404.4314*.

Korhonen, A. (2002). Subcategorization acquisition. Technical report, Computer Laboratory, University of Cambridge.

Korhonen, A., Gorrell, G., and McCarthy, D. (2000). Statistical filtering and subcategorization frame acquisition. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 199–206, Hong Kong. ACL.

Krause, T. and Zeldes, A. (2016). ANNIS3: A new architecture for generic corpus query and visualization. *Digital Scholarship in the Humanities*, 31(1):118–139.

Krivanek, J. and Meurers, D. (2011). Comparing rule-based and data-driven dependency parsing of learner language. In *Proceedings of the First International Conference on Dependency Linguistics*, pages 310–317, Barcelona. IOS Press.

Kyle, K. (2016). *Measuring syntactic development in L2 writing: Fine grained indices of syntactic complexity and usage-based indices of syntactic sophistication*. PhD thesis, Georgia State University.

Langacker, R. W. (1987). *Foundations of cognitive grammar: Theoretical prerequisites*. Stanford: Stanford university press.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

Lenci, A., McGillivray, B., Montemagni, S., and Pirrelli, V. (2008). Unsupervised acquisition of verb subcategorization frames from shallow-parsed corpora. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, pages 3000–3006, Marrakech. ELRA.

Levin, B. (1993). *English verb classes and alternations: A preliminary investigation*. London: University of Chicago Press.

Levin, B. and Hovav, M. R. (1995). *Unaccusativity: At the syntax-lexical semantics interface*. Cambridge MA: MIT press.

Lewis, M. P., Simons, G. F., Fennig, C. D., et al. (2009). *Ethnologue: Languages of the world*. Dallas: SIL international.

Li, Z., Zhang, M., Che, W., Liu, T., Chen, W., and Li, H. (2011). Joint models for Chinese POS tagging and dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1180–1191, Edinburgh. ACL.

Lippincott, T., Séaghdha, D. O., and Korhonen, A. (2012). Learning syntactic verb frames using graphical models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 420–429, Jeju Island. ACL.

Lu, X. (2010). Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15(4):474–496.

MacArthur, R. H. (1965). Patterns of species diversity. *Biological Reviews*, 40(4):510–533.

Manning, C. D. (1993). Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 235–242, Columbus. ACL.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Martins, A. F. T., Almeida, M., and Smith, N. A. (2013). Turning on the Turbo: Fast third-order non-projective Turbo parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 617–622, Sofia. ACL.

Martins, A. F. T., Smith, N. A., Xing, E. P., Aguiar, P. M. Q., and Figueiredo, M. A. T. (2010). Turbo parsers: Dependency parsing by approximate variational inference. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 34–44, Cambridge, MA. ACL.

McDonald, R., Crammer, K., and Pereira, F. (2005). Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 91–98, Ann Arbor. ACL.

McDonald, R. and Pereira, F. (2006). Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento. ACL.

McDonough, K. (2006). Interaction and syntactic priming: English L2 speakers' production of dative constructions. *Studies in Second Language Acquisition*, 28(2):179–207.

McDonough, K. and Nekrasova-Becker, T. (2014). Comparing the effect of skewed and balanced input on English as a foreign language learners' comprehension of the double-object dative construction. *Applied Psycholinguistics*, 35(2):419–442.

McDonough, K. and Trofimovich, P. (2016). The role of statistical learning and working memory in L2 speakers' pattern learning. *The Modern Language Journal*, 100(2):428–445.

Mel'čuk, I. A. (1988). *Dependency syntax: Theory and practice*. New York: SUNY press.

Meng, F., Lu, Z., Wang, M., Li, H., Jiang, W., and Liu, Q. (2015). Encoding source language with convolutional neural network for machine translation. 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 20–30. ACL.

Messiant, C., Poibeau, T., and Korhonen, A. (2008). LexSchem: A large subcategorization lexicon for French verbs. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, pages 41–68, Marrakech. ELRA.

Meurers, D. (2012). Natural language processing and language learning. In Chapelle, C. A., editor, *The encyclopedia of applied linguistics*, pages 4193–4205. Oxford: Wiley.

Meurers, D. (2015). Learner corpora and natural language processing. In Granger, S., Gilquin, G., and Meunier, F., editors, *The Cambridge handbook of learner corpus research*, pages 537–566. Cambridge: Cambridge University Press.

Meurers, D. and Dickinson, M. (2017). Evidence and interpretation in language learning research: Opportunities for collaboration with computational linguistics. *Language Learning*, 67(S1):66–95.

Meurers, D., Krivanek, J., and Bykh, S. (2013). On the automatic analysis of learner corpora: Native language identification as experimental testbed of language modeling between surface features and linguistic abstraction. In Sintes, A. A. and Hernández, S. V., editors, *Diachrony and synchrony in English corpus studies*. Frankfurt am Main: Peter Lang.

Meyers, A., Macleod, C., and Grishman, R. (1996). Standardization of the complement adjunct distinction. In *Proceedings of EURALEX 96 (International Conference on Lexicography)*, Gothenberg. EURALEX.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*, pages 1–12, Scottsdale.

Montrul, S. A. (1998). The L2 acquisition of dative experiencer subjects. *Second Language Research*, 14(1):27–61.

Murakami, A. and Alexopoulou, T. (2015). L1 influence on the acquisition order of English grammatical morphemes. *Studies in Second Language Acquisition*, 38(3):365–401.

Nagata, R. and Sakaguchi, K. (2016). Phrase structure annotation and parsing for learner English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1837–1847, Berlin. ACL.

Naseem, T., Snyder, B., Eisenstein, J., and Barzilay, R. (2009). Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of Artificial Intelligence Research*, 36(1):341–385.

Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., and Bryant, C. (2014). The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Baltimore. ACL.

Nicholls, D. (2003). The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics 2003 Conference*, pages 572–581, Lancaster. UCREL.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R. T., Petrov, S., Pyysalo, S., Silveira, N., et al. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, Portorož. ELRA.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E. (2007). MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02):95–135.

Nivre, J., Hall, J., Nilsson, J., Eryigit, G., and Marinov, S. (2006). Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 221–225, New York.

Norris, J. M. and Ortega, L. (2009). Towards an organic approach to investigating CAF in instructed SLA: The case of complexity. *Applied Linguistics*, 30(4):555–578.

Novack-Gottshall, P. M. (2007). Using a theoretical ecospace to quantify the ecological diversity of Paleozoic and modern marine biotas. *Paleobiology*, 33(2):273–294.

O'Donovan, R., Burke, M., Cahill, A., Van Genabith, J., and Way, A. (2005). Large-scale induction and evaluation of lexical resources from the Penn-II and Penn-III Treebanks. *Computational Linguistics*, 31(3):329–366.

Osborne, T. (2015). Diagnostics for constituents: Dependency, constituency, and the status of function words. In *Proceedings of the Third International Conference on Dependency Linguistics*, pages 251–260, Uppsala.

Ott, N. and Ziai, R. (2010). Evaluating dependency parsing performance on German learner language. In *Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories*, pages 175–186, Tartu. NEALT.

Owens, J. (1984). On getting a head: A problem in dependency grammar. *Lingua*, 62(1):25 – 42.

Perkins, R. (2015). Native language identification (NLID) for forensic authorship analysis of weblogs. In Dawson, M. and Omar, M., editors, *New threats and countermeasures in digital crime and cyber terrorism*, pages 213–234. Hershey: IGI Global.

Petrov, S., Das, D., and McDonald, R. (2012). A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, Istanbul. ELRA.

Petrov, S. and Klein, D. (2007). Improved inference for unlexicalized parsing. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, volume 7, pages 404–411, Rochester. ACL.

Pinker, S. (1989). *Learnability and cognition: The acquisition of argument structure*. Cambridge, MA: MIT press.

Pollard, C. and Sag, I. A. (1994). *Head-driven phrase structure grammar*. Chicago: University of Chicago Press.

Preiss, J., Briscoe, T., and Korhonen, A. (2007). A system for large-scale acquisition of verbal, nominal and adjectival subcategorization frames from corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 912–919, Prague. ACL.

Procter, P. (1978). *Longman dictionary of contemporary English*. England: Longman.

Przepiórkowski, A. (2009). Towards the automatic acquisition of a valence dictionary for polish. In Marciniak, M. and Mykowiecka, A., editors, *Aspects of natural language processing: Essays dedicated to Leonard Bolc on the occasion of his 75th birthday*, pages 191–210. Berlin: Springer.

Przepiórkowski, A. and Woliński, M. (2003). A flexemic tagset for Polish. In *Proceedings of the 2003 EACL Workshop on Morphological Processing of Slavic Languages*, pages 33–40, Budapest. ACL.

Pustejovsky, J. (1991). The generative lexicon. *Computational linguistics*, 17(4):409–441.

Quochi, V., Frontini, F., Bartolini, R., Hamon, O., Poch, M., Padró, M., Bel, N., Thurmair, G., Toral, A., and Kamram, A. (2014). Third evaluation report. Evaluation of PANACEA v3 and produced resources.

Ragheb, M. and Dickinson, M. (2011). Avoiding the comparative fallacy in the annotation of learner corpora. In *Selected Proceedings of the 2010 Second Language Research Forum: Reconsidering SLA Research, Dimensions, and Directions*, pages 114–124, Somerville. Cascadilla Proceedings Project.

Ragheb, M. and Dickinson, M. (2012). Defining syntax for learner language annotation. In *Proceedings of COLING 2012: Posters*, pages 965–974, Mumbai. IIT Bombay.

Ragheb, M. and Dickinson, M. (2013). Inter-annotator agreement for dependency annotation of learner language. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 169–179, Atlanta. ACL.

Ragheb, M. and Dickinson, M. (2014). Developing a corpus of syntactically-annotated learner language for English. *CLARIN-D*, pages 292–300.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, Philadelphia. ACL.

Rehbein, I., Hirschmann, H., Lüdeling, A., and Reznicek, M. (2012). Better tags give better trees – or do they? *Linguistic Issues in Language Technology*, 7(10):1–18.

Reichart, R. and Korhonen, A. (2013). Improved lexical acquisition through DPP-based verb clustering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 862–872.

Reznicek, M., Ludeling, A., and Hirschmann, H. (2013). Competing target hypotheses in the Falko corpus. In *Automatic treatment and analysis of learner corpus data*, pages 101–123. Amsterdam: John Benjamins.

Rijhwani, S., Sequiera, R., Choudhury, M., Bali, K., and Maddila, C. S. (2017). Estimating code-switching on twitter with a novel generalized word-level language detection technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1971–1982, Vancouver. ACL.

Robinson, J. J. (1970). Dependency structures and transformational rules. *Language*, 46(2):259–285.

Robinson, P. (2005). Cognitive complexity and task sequencing: Studies in a componential framework for second language task design. *IRAL-International Review of Applied Linguistics in Language Teaching*, 43(1):1–32.

Rogerson, P. (2001). *Statistical Methods for Geography*. London: Sage.

Römer, U., O'Donnell, M. B., and Ellis, N. C. (2015). Using COBUILD grammar patterns for a large-scale analysis of verb-argument constructions. In Groom, N., Charles, M., and John, S., editors, *Corpora, grammar and discourse: In honour of Susan Hunston*, pages 43–72. Philadelphia: John Benjamins.

Römer, U., Roberson, A., O'Donnell, M. B., and Ellis, N. C. (2014). Linking learner corpus and experimental data in studying second language learners' knowledge of verb-argument constructions. *ICAME Journal*, 38(1):115–135.

Rosen, A., Hana, J., Štindlová, B., and Feldman, A. (2014). Evaluating and automating the annotation of a learner corpus. *Language Resources and Evaluation*, 48(1):65–92.

Roth, M. and Lapata, M. (2016). Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1192–1202, Berlin. ACL.

Santorini, B. (1990). *Part-of-speech tagging guidelines for the Penn Treebank Project(3rd revision, 2nd printing)*. Technical report,Department of Computer and Information Science, University of Pennsylvania.

Schachter, P. and Shopen, T. (1985). Parts-of-speech systems. In Shopen, T., editor, *Language typology and syntactic description*, volume 1, pages 3–61.

Schmid, H. and Laws, F. (2008). Estimation of conditional probabilities with decision trees and an application to fine-grained pos tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 777–784, Manchester. ACL.

Schubert, K. (1987). *Metataxis: Contrastive dependency syntax for machine translation*. Dordrecht: Foris.

Sgall, P., Hajicová, E., and Panevová, J. (1986). *The meaning of the sentence in its semantic and pragmatic aspects*. Dordrecht: D. Reidel.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656.

Silveira, N., Dozat, T., De Marneffe, M.-C., Bowman, S. R., Connor, M., Bauer, J., and Manning, C. D. (2014). A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 2897–2904, Reykjavik. ELRA.

Sinicrope, C. and Byrnes, H. (2009). Advancedness and the development of relativization in L2 German: A curriculum-based longitudinal study. In Ortega, L. and Byrnes, H., editors, *The longitudinal study of advanced L2 capacities*, pages 125–154. New York: Routledge.

Skehan, P. (1998). *A cognitive approach to language learning*. Oxford: Oxford University Press.

Skjærholt, A. (2013). Influence of preprocessing on dependency syntax annotation: Speed and agreement. In *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse*, pages 28–32, Sofia. ACL.

Snyder, B., Naseem, T., Eisenstein, J., and Barzilay, R. (2009). Adding more languages improves unsupervised multilingual part-of-speech tagging: A Bayesian non-parametric approach. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 83–91, Boulder. ACL.

Somers, H. L. (1984). On the validity of the complement-adjunct distinction in valency grammar. *Linguistics*, 22(4):507–530.

Souza, R. A. (2011). Argument structure in L2 acquisition: Language transfer re-visited in a semantics and syntax perspective. *Ilha do Desterro: A Journal of English Language, Literatures in English and Cultural Studies*, (60):153–188.

Steinitz, R. and Lang, E. (1973). *Adverbial-syntax*. Berlin: Akademie Verlag.

Sun, X. (2014). Structure regularization for structured prediction. In *Advances in Neural Information Processing Systems 27*, pages 2402–2410, Montréal. Curran Associates, Inc.

Svartvik, J. (1990). *The London-Lund corpus of spoken English: Description and research*. Lund: Lund University Press.

Taguchi, N., Crawford, W., and Wetzel, D. Z. (2013). What linguistic features are indicative of writing quality? A case of argumentative essays in a college composition program. *TESOL Quarterly*, 47(2):420–430.

Tallerman, M. (2013). *Understanding syntax*. New York: Routledge.

Taylor, A., Marcus, M., and Santorini, B. (2003). The Penn Treebank: An overview. In Abeillé, A., editor, *Treebanks: Building and using parsed corpora*, pages 5–22. Dordrecht: Springer.

Tesnière, L. (1965). *Eléments de syntaxe structurale*. Philadelphia: John Benjamins.

Tetreault, J., Blanchard, D., and Cahill, A. (2013). A report on the first native language identification shared task. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 48–57, Atlanta.

Tetreault, J., Foster, J., and Chodorow, M. (2010). Using parse features for preposition selection and error detection. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 353–358, Uppsala. ACL.

Tono, Y. (2004). Multiple comparisons of IL, L1 and TL corpora: the case of L2 acquisition of verb subcategorization patterns by Japanese learners of English. *Corpora and Language Learners*, 17:45–66.

Tono, Y. and Díez-Bedmar, M. B. (2014). Focus on learner writing at the beginning and intermediate stages: The ICCI corpus. *International Journal of Corpus Linguistics*, 19(2):163–177.

Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., and Tsujii, J. (2005). Developing a robust part-of-speech tagger for biomedical text. In *Panhellenic Conference on Informatics*, pages 382–392. Berlin: Springer.

Van de Cruys, T., Rimell, L., Poibeau, T., and Korhonen, A. (2012). Multi-way tensor factorization for unsupervised lexical acquisition. In *Proceedings of COLING 2012: Technical Papers*, pages 2703–2720, Mumbai. IIT Bombay.

Van Rooy, B. and Schäfer, L. (2002). The effect of learner errors on POS tag errors during automatic POS tagging. *Southern African Linguistics and Applied Language Studies*, 20(4):325–335.

Vyatkina, N. (2013). Specific syntactic complexity: Developmental profiling of individuals based on an annotated learner corpus. *The Modern Language Journal*, 97(S1):11–30.

Wang, Y., Li, S., and Wang, H. (2017). A two-stage parsing method for text-level discourse analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 184–188, Vancouver. ACL.

White, L. (1987). Markedness and second language acquisition: The question of transfer. *Studies in Second Language Acquisition*, 9(3):261–285.

White, L. (1991). Argument structure in second language acquisition. *Journal of French Language Studies*, 1(2):189–207.

Wolfe-Quintero, K., Inagaki, S., and Kim, H.-Y. (1998). *Second language development in writing: Measures of fluency, accuracy, & complexity*. Honolulu: University of Hawai'i Press.

Xi, C. and Hwa, R. (2005). A backoff model for bootstrapping resources for non-English languages. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 851–858, Vancouver. ACL.

Xia, F. (2000). The part-of-speech tagging guidelines for the Penn Chinese Treebank (3.0). Technical report, University of Pennsylvania.

Yamada, H. and Matsumoto, Y. (2003). Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop of Parsing Technologies*, volume 3, pages 195–206, Nancy. ACL.

Yarowsky, D. and Ngai, G. (2001). Inducing multilingual POS taggers and NP brackets via robust projection across aligned corpora. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, Pittsburgh. ACL.

Zhang, Y. and Clark, S. (2008). A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 562–571, Honolulu. ACL.

Zobl, H. (1989). Canonical typological structures and ergativity in English L2 acquisition. In Gass, S. M. and Schachter, J., editors, *Linguistic perspectives on second language acquisition*, pages 203–221. Cambridge: Cambridge University Press.

Appendix A

Taxonomy of learner errors

Table A.1 Taxonomy of learner errors

Tag	Explanation
AGA	pronoun agreement error
AGD	determiner agreement error
AGN	noun agreement error
AGV	verb agreement error
AS	incorrect argument structure
C	capitalization error
CD	wrong determiner for noun countability
CE	compound error
CL	collocation error
CN	countability of noun error
CQ	wrong quantifier for noun countability
DA	wrongly derived pronoun
DC	wrongly derived conjunction
DD	wrongly derived determiner
DJ	wrongly derived adjective
DN	wrongly derived noun
DQ	wrongly derived quantifier
DT	wrongly derived preposition
DV	wrongly derived verb
DY	wrongly derived adverb
FA	wrong form of pronoun

Tag	Explanation
FC	wrong form of conjunction
FD	wrong form of determiner
FJ	wrong form of adjective
FN	wrong form of noun
FQ	wrong form of quantifier
FT	wrong form of preposition
FV	wrong form of verb
FY	wrong form of adverb
IA	Incorrect formation of pronoun
ID	idiom error
IJ	incorrect formation of adjective
IN	incorrect formation of noun plural
IQ	incorrect formation of quantifier
IV	incorrect verb inflection
IY	Incorrect formation of adverb
L	inappropriate register (label)
M	something missing
MA	pronoun missing
MC	conjunction missing
MD	determiner missing
MJ	adjective missing
MN	noun missing
MP	punctuation missing
MQ	quantifier missing
MT	preposition missing
MV	verb missing
MY	adverb missing
R	something needs replacing
RA	pronoun needs replacing
RC	conjunction needs replacing
RD	determiner needs replacing
RJ	adjective needs replacing
RN	noun needs replacing
RP	punctuation needs replacing

Tag	Explanation
RQ	quantifier needs replacing
RS	wrong split/concatenation
RT	preposition needs replacing
RV	verb needs replacing
RY	adverb needs replacing
S	spelling error
SX	spelling confusion error
TV	wrong tense of verb
U	something unnecessary
UA	pronoun unnecessary
UC	conjunction unnecessary
UD	determiner unnecessary
UJ	adjective unnecessary
UN	noun unnecessary
UP	punctuation unnecessary
UQ	quantifier unnecessary
UT	preposition unnecessary
UV	verb unnecessary
UY	adverb unnecessary
W	incorrect word order
X	incorrect formation of negative

Appendix B

SCF inventory and examples

Table B.1 shows the SCF inventory. The symbol * indicates that an SCF has identification accuracy statistics on our learner data in Table 5.3; the symbol ⋄ indicates that an SCF appears in the learner data less than 5 times and therefore has no informative accuracy statistics; "N" indicates that the SCF appears in our native data. The column of "FS #" denotes the fine-grained SCF type of Preiss et al. (2007) for each example, demonstrating the mapping between the fine-grained and coarse-grained SCFs.

Table B.1 SCF inventory and examples

#		SCF	FS #	Examples
1	*N	acomp	1	His reputation sank low.
			2	He appears crazy / distressed.
			4	He seems well.
2		acomp:prt	137	He started out poor.
3	*N	advmod	3	He meant well.
			160	It carves easily.
4	*N	advmod:dobj_N	27	He put it there
			28	They mistakenly thought him here.
5		advmod:prt	126	He came off badly.
6	N	ccomp_VBARE=>mark-that	106	She demanded that he leave.
7		ccomp_VBARE=>mark-that:iobj	133	He petitioned them that he be freed.
8		ccomp_VBARE=>mark-that:pobj	98	They suggested to him that he go.
9	*N	ccomp_VTENSED	104	They thought he was always late.
			159	He seems as if he is clever.
10	*N	ccomp_VTENSED=>mark-that	105	To report the theft indicates that he was n't guilty.
			107	It seems that they left.
			109	He complained that they came.
11	◇N	ccomp_VTENSED=>mark-that:dobj_N	6	It annoys them that she left.
			166	I take it that kim left.
			158	It is believed that he came.
12		ccomp_VTENSED=>mark-that:dobj_N:iobj	132	He bet her ten pounds that he came.

#		SCF	FS #	Examples
13		ccomp_VTENSED=>mark-that:dobj_N:prt	130	He had her on that he attended.
14	N	ccomp_VTENSED=>mark-that:iobj	52	He told the audience that he was leaving.
15	◇	ccomp_VTENSED=>mark-that:pobj	12	It matters to them that she left.
			97	They admitted to the authorities that they had entered illegally.
16		ccomp_VTENSED=>mark-that:pobj:prt	131	She gets through to him that he came.
17	N	ccomp_VTENSED=>mark-that:prt	83	They figured out that she had n't done her job.
			128	It turns out that he did it.
18	*N	ccomp_WHCOMP	16	He asked how she did it.
			113	He asked whether he should come.
			114	He asked what he should do.
19	◇	ccomp_WHCOMP:dobj_N	134	I would appreciate it if he came.
20	◇N	ccomp_WHCOMP:iobj	59	They asked him whether he was going.
			60	They asked him what he was doing.
			156	He asked him how he came.
21		ccomp_WHCOMP:pobj	89	He explained to her how she did it.
			100	They asked about everybody whether they had enrolled.
			101	They asked about everybody what they had done.
			135	It dawned on him what he should do.
22	N	ccomp_WHCOMP:prt	79	They figured out whether she had n't done her job.
			80	They figured out what she had n't done.
23	*N	dobj_N	7	That she left annoys them.
			8	To read pleases them.
			24	He loved her.

#		SCF	FS #	Examples
			36	He combed the woods looking for her.
			123	It cost ten pounds.
24	*N	dobj_N:iobj	37	She asked him his name.
			124	It cost him ten pounds.
25		dobj_N:iobj:pobj:xcomp_VBARE=>aux_TO	167	It cost kim a pound for us to go.
26		dobj_N:iobj:prt	117	I opened him up a new bank account.
			125	It set him back ten pounds.
27		dobj_N:iobj:xcomp_VBARE=>aux_TO	168	It took us an hour to find.
28	◇N	dobj_N:pcomp	45	They helped me with whatever I was doing.
			143	He strikes me as foolish.
			147	He condemned him as stupid.
			162	He accepted him as associated.
			163	He accepted him as being normal.
29		dobj_N:pcomp:prt	148	He put him down as stupid.
30	N	dobj_N:pcomp_VING	39	I prevented her from leaving.
			40	I accused her of murdering her husband.
			41	He wasted time on fussing with his hair.
			42	He told her about climbing the mountain.
			43	He attributed his failure to noone buying his books.
			44	They asked him about his participating in the conference.
31		dobj_N:pcomp_VING:prt	152	He talked him around into leaving.
32	*N	dobj_N:pobj	29	I sent him as a messenger.
			30	She served the firm as a researcher.
			31	She bought a book for him.

#		SCF	FS #	Examples
			49	She added the flowers to the bouquet.
			50	I considered that problem of little concern.
			56	He gave a big kiss to his mother.
			118	He made use of the money.
33	◇N	dobj_N:pobj:pobj	122	He turned it from a disaster into a victory.
34	*N	dobj_N:pobj:prt	77	I separated out the three boys from the crowd.
35	N	dobj_N:pobj:xcomp_VBARE=>aux_TO	165	I arranged it with kim to meet.
			157	It requires ten pounds for him to go.
36	*N	dobj_N:prt	76	I looked up the entry.
37		dobj_N:prt:xcomp_ADJ	145	He makes him out crazy.
			146	He sands it down smooth.
38	N	dobj_N:prt:xcomp_VBARE=>aux_TO	149	He made him out to be crazy.
			150	He spurred him on to try.
39	*N	dobj_N:xcomp_ADJ	25	He painted the car black.
			26	She considered him foolish.
40	◇N	dobj_N:xcomp_N	38	They appointed him professor.
41	*N	dobj_N:xcomp_VBARE	32	He made her sing.
			33	He helped her bake the cake.
42	*N	dobj_N:xcomp_VBARE=>aux_TO	11	It pleases them to find a cure.
			53	I advised Mary to go.
			54	John promised Mary to resign.
			55	They badgered him to go.
			57	I found him to be a good doctor.

#		SCF	FS #	Examples
43	◇	dobj_N:xcomp_VEN	58	He wanted the children found.
44	*N	dobj_N:xcomp_VING	34	I kept them laughing.
			35	I caught him stealing.
45		iobj:xcomp_WHCOMP	61	He asked him whether to clean the house.
			62	He asked him what to do.
46	◇	pcomp	70	He thought about whether he wanted to go.
			71	He thought about what he wanted.
			72	He thought about whether to go.
			73	He thought about what to do.
47		pcomp:pobj	91	I agreed with him about whether he should kill the peasants.
			92	I agreed with him about what he should do.
			93	I agreed with him about what to do.
			94	I agreed with him about whether to go.
48	*N	pcomp_VING	63	They failed in attempting the climb.
			64	They disapproved of attempting the climb.
			69	They argued about his coming.
49	N	pcomp_VING:prt	140	He got around to leaving.
50	*N	pobj	5	I worked as an apprentice cook.
			14	That she left matters to them.
			65	They worried about him drinking.
			87	They apologised to him.
			96	The matter seems in dispute.
51	◇N	pobj:pobj	95	They flew from London to Rome.
52		pobj:pobj:prt	121	He came down on him for his bad behavior.

#		SCF	FS #	Examples
53		pobj:pobj:xcomp_VBARE=>aux_TO	88	They contracted with him for the man to go.
54	*N	pobj:prt	78	She looked in on her friend.
55		pobj:prt:xcomp_VBARE=>aux_TO	151	He kept on at him to join.
56	◇	pobj:xcomp_VBARE	153	He looked at him leave.
57	◇N	pobj:xcomp_VBARE=>aux_TO	10	It remains for us to find a cure.
			13	It occurred to them to watch.
			15	I prefer for her to do it.
			66	He conspired with them to do it.
			67	He beckoned to him to come.
			68	She appealed to him to go.
			99	He appeared to her to be ill.
58	N	pobj:xcomp_VING	85	She attributed his drinking too much to his anxiety.
			86	They limited smoking a pipe to the lounge.
59		pobj:xcomp_WHCOMP	90	He explained to them how to do it.
			102	They deduced from kim whether to go.
			103	They deduced from kim what to do.
60	*N	prt	74	She gave up.
61	N	prt:xcomp_N	136	He turned out a fool.
62	◇	prt:xcomp_VBARE=>aux_TO	138	He turned out to be a crook.
			139	He set out to win.
63	N	prt:xcomp_VING	75	He ruled out paying her debts.
64		prt:xcomp_WHCOMP	81	They figured out whether to go.
			82	They figured out what to do.

#		SCF	FS #	Examples
65	*N	su	22	He went.
			23	They met.
			129	That he came matters.
			154	To see them hurts.
			164	It rains.
66	*N	xcomp_N	51	He seemed a fool.
67	*N	xcomp_VBARE	18	He helped bake the cake.
			142	He dared dance.
68	*N	xcomp_VBARE=>aux_TO	9	It remains to find a cure.
			110	He helped to save the child.
			111	He seemed to come.
			112	I wanted to come.
69	*N	xcomp_VING	19	His hair needs combing.
			20	She stopped smoking.
			21	She discussed writing novels.
			84	He dismissed their writing novels.
70	◇N	xcomp_WHCOMP	17	He explained how to do it.
			115	He asked whether to clean the house.
			116	He asked what to do.