# Automatic Correction of Grammatical Errors in Non-native English Text

by

John Sie Yuen Lee

BMath in Computer Science, University of Waterloo (2002)
S.M. in Electrical Engineering and Computer Science, Massachusetts Institute of Technology (2004)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author ..............................................................
Department of Electrical Engineering and Computer Science
May 20, 2009

Certified by..........................................................
Dr. Stephanie Seneff
Principal Research Scientist
Thesis Supervisor

Accepted by..........................................................
Dr. Terry Orlando
Chairman, Department Committee on Graduate Students

# Automatic Correction of Grammatical Errors in Non-native English Text

by

John Sie Yuen Lee

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

## Abstract

Learning a foreign language requires much practice outside of the classroom. Computer-assisted language learning systems can help fill this need, and one desirable capability of such systems is the automatic correction of grammatical errors in texts written by non-native speakers.

This dissertation concerns the correction of non-native grammatical errors in English text, and the closely related task of generating test items for language learning, using a combination of statistical and linguistic methods. We show that syntactic analysis enables extraction of more salient features. We address issues concerning robustness in feature extraction from non-native texts; and also design a framework for simultaneous correction of multiple error types. Our proposed methods are applied on some of the most common usage errors, including prepositions, verb forms, and articles. The methods are evaluated on sentences with synthetic and real errors, and in both restricted and open domains.

A secondary theme of this dissertation is that of user customization. We perform a detailed analysis on a non-native corpus, illustrating the utility of an error model based on the mother tongue. We study the benefits of adjusting the correction models based on the quality of the input text; and also present novel methods to generate high-quality multiple-choice items that are tailored to the interests of the user.

Thesis Supervisor: Dr. Stephanie Seneff
Title: Principal Research Scientist

## Acknowledgments

I would like to thank my advisor, Stephanie Seneff, for her patience with my shortcomings and her unfailing support during the ups and downs of these seven years.

This dissertation was made possible by research assistantships provided by the Spoken Language Systems group at the MIT Computer Science and Artificial Intelligence Laboratory.

The research in **Chapter 4** was partially conducted at Microsoft Research Asia in the summer of 2006. We thank Xiaohua Liu and Ming Zhou for their collaboration.

The research in **Chapter 5** was partially conducted at the Royal Institute of Technology, Stockholm, Sweden, in the summer of 2007. We thank Viggo Kann, Martin Volk, as well as the two anonymous reviewers, for their helpful discussions.

The research in **Chapter 6** utilized a non-native corpus generously provided by Prof. John Milton, Hong Kong University of Science and Technology. We also thank Tom Lee and Ken Schutte for their assistance with the evaluation.

The research in **Chapter 7** was in part supported by a fellowship from the National Sciences and Engineering Research Council of Canada, and by the NTT Corporation.

The research in **Chapter 8** was sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government. This work is also supported in part by the Natural Sciences Engineering Research Council of Canada. We thank Chao Wang and the four judges, Ed Filisko, Alex Gruenstein, Mitch Peabody and Daniel Schultz.

The research in **Chapter 9** was partially supported by MIT Lincoln Laboratory and by a post-graduate fellowship from the Natural Sciences and Engineering Research Council of Canada. We thank the four subjects and Hsu-Chih Wu for assisting in the user study, and Terry Koo for parsing the English corpus.

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

Non-native speakers of English, who outnumber their native counterparts by at least a factor of three (Ronowicz & Yallop, 2005), increasingly need to communicate in this *lingua franca* of the world. Many students enroll in English-as-a-Second-Language (ESL) courses at secondary schools and universities, but time in the classroom is hardly adequate for the immersion and practice necessary to learn to express oneself fluently in a foreign language. As a result, students often seek extra tutoring, and rely on teachers and proofreaders to correct mistakes in their writing.

Researchers have turned to computational methods to help in these time-consuming tasks. Computer-assisted language learning systems (Johnson et al., 2004; Fok & Ip, 2006; McGraw & Seneff, 2008) provide a non-threatening environment for language students to practice at their leisure. In particular, web-based applications, such as *Criterion* (Burstein et al., 2004) and *ESL Assistant* (Gamon et al., 2008), provided by Educational Testing Service and Microsoft Research, function as posteditors (Knight & Chander, 1994) for non-native speakers, allowing them to submit their writing and receive feedback. A large number of grammar checking systems (Izumi et al., 2003; Eeg-Olofsson & Knutsson, 2003; Chodorow et al., 2007; Felice & Pulman, 2007; Yi et al., 2008) have also been designed in recent years, typically specializing in usage errors for particular parts-of-speech. This dissertation seeks to advance the state-of-the-art in grammatical error correction using a combination of statistical and linguistic methods.

## 1.2 Scope of Research Topic

While English is by far the most spoken foreign language, non-native speakers of other languages have also created demand for grammar checkers, from Japanese (Uchimoto et al., 2002; Fujita et al., 2004) to Swedish (Eeg-Olofsson & Knutsson, 2003; Carlberger et al., 2005). This dissertation focuses on English texts, since most of the available corpora are in English. Although our focus is on English, our methods do not draw on any specificity of the language.

Among grammar checkers for non-native speakers of English, some are especially aimed at users of a particular native language, such as American Sign Language (Michaud et al., 2000; Birnbaum, 2005). While the correction strategies in this dissertation do not take advantage of information about the author's native language, we will argue in §3.2 that

| |
|---|
| **Spelling Errors**<br>*Rita said [ too → to ] that if he wants to give her a ring ...* |
| **Context-dependent Errors**<br>*I am [ prepared → preparing ] for the exam.* |
| **Lexical Choice Errors**<br>*I had a cup of [ strong → powerful ] tea.* |
| **Function Words and Conjugation Errors**<br>*I have no medicine but just [ eat → eating ] 5 to 6 meals a day.*<br>*Can we discuss [ ⟨null⟩ → about ] this?*<br>*Let's go see [ a → ⟨null⟩ ] movie.* |

Table 1.1: Examples of different types of errors. Throughout this dissertation, the notation [⟨*crr*⟩ → ⟨*err*⟩] indicates that the correct word, ⟨*crr*⟩, is mistakenly replaced with the word ⟨*err*⟩ by the non-native speaker. The boundaries between these error types are not always clear-cut, and text written by language learners may contain any of these error types. This dissertation will focus on the "Function Words and Conjugation Errors" category. More detailed descriptions can be found in §1.2.

using such information may be a promising direction of research.

Non-native speakers may make many types of mistakes, and we make no pretense of being able to tackle them all. We now characterize some of these mistake types (see Table 1.1), and define the scope of this dissertation.

**Spelling Errors** Given an English dictionary, words that are not in the dictionary would be obvious candidates to be corrected. A noisy channel model is used in (Brill & Moore, 2000) to correct such words. Real-word spelling correction could also be viewed as a word disambiguation problem for a set of word pairs (Carlson et al., 2001), called *confusion sets*, e.g., {*principle, principal*}. In (Chodorow & Leacock, 2000), a usage model flags a word when its context deviates too far from the model. A limitation of both of these approaches is that the set of target words needs to be pre-defined. In contrast, (Hirst & Budanitsky, 2005) do not rely on confusion sets, and perform spelling correction by optimizing semantic cohesion.

In this dissertation, we consider spelling errors to be a separate problem that should be dealt with in a pre-processing step, as is the practice in (Tetreault & Chodorow, 2008b).

**Context-dependent Errors** Some errors are not evident in the target sentence itself, and may be detectable only at the discourse level. In the example in Table 1.1, semantic analysis in the context would be needed to determine whether the author means he is "*prepared*" or "*preparing*" for the examination. This kind of error is difficult to recover even with the state-of-the-art in natural language understanding.

In our correction algorithms, we will assume that only the target sentence is given. Nonetheless, there is evidence that contextual information is helpful for at least some parts-of-speech (see §10.3.1).

**Lexical Choice** Another frequent error category is lexical choice. A sentence may be perfectly grammatical and yet violates the habitual usage of words. These wrong lexical choices can result in awkward collocations, such as in the example given in Table 1.1.

Detecting these collocations is a research area in its own right, and has been explored both in the context of machine translation (Lu & Zhou, 2004), and in non-native texts (Gao, 2004). Lexical choice will not be considered in this dissertation.

**Machine Translation (MT) Output** Some MT systems make systematic fluency errors (Kauchak & Elkan, 2003; Seneff et al., 2006). To the extent that these errors resemble non-native errors, they can potentially be treated by techniques similar to those proposed in this dissertation. While some researchers identify these errors using the source sentence[1], others do so by looking only at the target sentence (Gamon et al., 2005). Although MT output is not our focus, we will discuss an application in §8.4.

**Function Words and Conjugation Errors** Although spelling errors, context-dependent errors, and inappropriate lexical choice may all appear in non-native writing, this dissertation will focus on another broad class of errors — those involving closed-class, or function words, and conjugation in verbs and nouns. The choice of these categories is not arbitrary; it is widely recognized that the usage of function words in English is difficult to master. In a set of 150 essays from the Test of English as a Foreign Language (TOEFL), written by native speakers of Chinese, Japanese and Russian, one article error was produced on average for every 16 noun phrases, or once every 3 or 4 sentences (Han et al., 2004). In an analysis of texts (Bitchener et al., 2005), written by students in ESL classes, errors involving prepositions form the largest category, at about 29%[2]. The misuse of verb forms is also among the most frequent types of grammatical errors found in a large non-native corpus compiled from Japanese native speakers (see §3.1.2). Mistakes associated with these function words are most commonly substitutions, but also include insertions and deletions[3].

Specifically, we will consider in detail the use of articles, prepositions and verb forms in the general domain in **Chapters 5, 6, and 7**; in **Chapter 8**, in a restricted domain, we will consider all of the above in addition to noun number and auxiliary verbs.

This dissertation is the first comprehensive study of the automatic correction of the usage of a wide variety of function words. It will do so via a combination of linguistic and statistical methods, while addressing issues of robustness and personalization.

We will also discuss a related task, the *detection* of non-native errors. Further, we describe two research directions that can benefit from this work — the editing of MT output, and the generation of test items for language-learning assessment.

[1] For example, a Japanese NP is classified as either definite or indefinite in (Heine, 1998). This information can then be used to improve its English translation.

[2] As cited in (Chodorow et al., 2007).

[3] While word-order mistakes are not uncommon, they do not figure among the most frequent error types (Izumi et al., 2004a), and will not be considered in this dissertation. There has been recent research on correcting word-order mistakes in Mandarin (Liu et al., 2008) and in English (Gamon et al., 2008), but little detail is given in the latter.

## 1.3 Contributions

Conceptually, the error correction task can be viewed as a type of *monolingual* "translation". Using this analogy, a source sentence is one that is written by a non-native speaker, and the target sentence should be a *fluent* and *adequate* sentence in the same language — i.e., grammatically correct, and preserving the intended meaning of its source.

Since non-native sentences are not available in large quantity, most researchers have recast the task as natural language generation, and used native sentences as substitute. Words that a non-native speaker is likely to misuse, if s/he were to compose these sentences, are removed; the goal, then, is for the system to predict what the original words should be. Typically, all words from a certain part-of-speech — one deemed to be problematic for non-native speakers[4] — are removed. Using preposition as an example, if the following sentence is presented to the system:

> Pierre Vinken joined the board ____ a nonexecutive director.

then the system would be expected to predict the preposition "*as*". This prediction task is more properly called *preposition generation*, but it is clearly related to the error correction task. If the system can accurately generate prepositions, then whenever the generated preposition differs from the actual one used in a non-native text, the system can suggest a correction. A similar set-up has also been used for *article generation* (Knight & Chander, 1994).

Since the number of candidates are limited for both prepositions and articles, the generation task is usually framed as a classification problem. A fixed window of words surrounding the word in question are extracted as features; a classifier is then trained on these features to predict the target. For instance, given the words preceding and following a noun, a statistical classifier can predict "*a/an*", "*the*" or *null* as its article (Knight & Chander, 1994).

This dissertation will tackle the error correction task from the natural language generation point of view, as describe above. The first of its two main contributions is the **use of linguistic features**. The neighboring words alone do not always provide adequate context for the generation task. In predicting prepositions, for example, we show that the prepositional phrase attachment site is an important consideration (Lee & Knutsson, 2008). However, the use of syntactic analysis must be robust on non-native texts; we investigate how robust analysis can be achieved for misused verb forms (Lee & Seneff, 2008b). For low-quality sentences with multiple errors, we demonstrate the usefulness of a sentence re-generation method in a restricted domain (Lee & Seneff, 2006).

The second contribution is in the area of **user customization**. Just as an elementary school teacher does not correct the writing of a third-grader in the same way a Wall Street Journal editor corrects a news story draft, grammar checkers should be optimized with respect to the characteristics and the needs of the author. Our research suggests that accuracy can be enhanced through awareness of the author's mother tongue (Lee & Seneff, 2008a) and proficiency (Lee, 2004). For the closely related task of generating test items for language learning, we propose methods to tailor the items to the interests of the user (Lee & Seneff, 2007).

[4] Most research has focused on function words such as articles and prepositions, but can also be different conjugations of nouns and verbs. See Chapter 3.

## 1.4 Outline

This introductory chapter is followed by a review of previous research in **Chapter 2**. In **Chapter 3**, an analysis of our data sets is presented. While a wide variety of grammatical mistakes may be observed in the speech or text produced by non-native speakers, the types and frequencies of these mistakes are not random (Wang & Garigliano, 1992). Certain parts of speech, for example, have been shown to be especially problematic for Japanese learners of English (Izumi et al., 2004a). Modeling these errors can potentially enhance the performance of grammar correction (Lee & Seneff, 2008a).

We will then start by attacking a more limited problem, that of *error detection*, in **Chapter 4**. Training statistical models to detect non-native sentences requires a large corpus of non-native writing samples, which is often not readily available. This chapter examines the extent to which machine-translated sentences can substitute as training data (Lee et al., 2007).

The next four chapters discuss the task of *error correction.* Each chapter highlights a novel feature or approach that addresses a deficiency in previous work:

**Use of Linguistic Features** In many state-of-the-art systems that use the classification approach, features are extracted only from the local context, such as a window of $n$ preceding and subsequent words (Chodorow et al., 2007). This window does not always include the most relevant context. In **Chapter 5**, we investigate the use of linguistic features so as to exploit words drawn from longer distances.

This investigation is carried out on the task of preposition generation. Relevant features for this task can range from lexical features, such as words and their part-of-speech tags in the vicinity of the preposition, to syntactic features that take into account the attachment site of the prepositional phrase (PP), as well as its argument/adjunct distinction. We compare the performance of these different kinds of features in a memory-based learning framework (Lee & Knutsson, 2008).

**Feature Extraction Robustness** The use of linguistic features raises a new concern: can such features be reliably extracted from ill-formed, noisy texts? **Chapter 6** discusses this issue in the context of verb forms. A basic approach for correcting English verb form errors is template matching in parse trees. We propose a method that improves on this approach in two ways. To improve recall, irregularities in parse trees caused by verb form errors are taken into account; to improve precision, $n$-gram counts are utilized to filter proposed corrections (Lee & Seneff, 2008b).

**Source Awareness** Error correction performance may improve if the characteristics of the author of the source text are taken into consideration. For example, the expectation of the types and frequency of errors in English articles may depend on whether the author's native language is French or Japanese, and the number of years of education in English.

We examine this issue in **Chapter 7** via the task of article insertion. One common mistake made by non-native speakers of English is to drop the articles *a*, *an*, or *the*. We apply the log-linear model to automatically restore missing articles based on features of the noun phrase. We first show that the model yields competitive results in article generation. Further, we describe methods to adjust the aggressiveness of the insertion algorithm based on the estimated quality of the input (Lee, 2004).

**Multiple Error Types** The approaches so far deal with only one error category in isolation, implicitly assuming correctness of the rest of the sentence. We now tackle sentences that may contain errors from multiple categories. In **Chapter 8**, we use a generation-based approach on a restricted domain, relying on parse scores to select the best sentence.

We describe our research on a sentence-level, generation-based approach to grammar correction: first, a word lattice of candidate corrections is generated from an ill-formed input. A traditional $n$-gram language model is used to produce a small set of N-best candidates, which are then reranked by parsing using a stochastic context-free grammar. We evaluate this approach in a flight domain with simulated ill-formed sentences (Lee & Seneff, 2006), and discuss its applications in machine translation (Seneff et al., 2006).

In **Chapter 9**, we turn to an educational application — generating fill-in-the-blank, multiple-choice items for prepositions (Lee & Seneff, 2007). These items are commonly used in language learning applications, but there has not been any attempt to generate them automatically for prepositions, whose usage often poses problems for non-native speakers of English. The construction of these items requires automatic generation of *wrong* choices of words, a task which is the opposite of post-editors and can draw on some similar techniques. The benefits of personalized items, tailored to the user's interest and proficiency, have motivated research on their automatic generation.

Finally, **Chapter 10** makes concluding remarks and sketches some directions for future research.

# Chapter 2

# Previous Research

In this chapter, we review previous research on the correction of grammatical errors in non-native texts. We first consider a few closely related tasks (§2.1). We then summarize previous research in three main paradigms — machine translation (§2.2), parsing (§2.3), and natural language generation (§2.4).

## 2.1 Related Tasks

Feedback about the quality of a sentence or text can come in many forms. In some applications, instead of actually correcting the error, which is the focus of this dissertation, it may also be useful to simply *detect* sentences that contain errors (§2.1.1), or to give a holistic score (§2.1.2) reflecting the quality of the text.

### 2.1.1 Error Detectors

Research in this area was originally designed for machine translation (MT) systems, aimed at automatically identifying low-quality output without looking at the source sentences; in other words, evaluating the "fluency" rather than the "adequacy" aspect. In a restricted domain, one way to detect such output is to parse the output with a semantic parser, e.g. (Seneff, 1992b). A parse failure would strongly suggest that the translation is flawed (Wang & Seneff, 2004).

In an open domain, a rule-based approach using lexical features is employed in (Eeg-Olofsson & Knutsson, 2003) for Swedish prepositions, and parts-of-speech sequences are used in (Tomokiyo & Jones, 2001). Another simple approach for identifying errorful sentences is to use the perplexity score from an $n$-gram model. Any sentence that scores below a certain threshold would be flagged. This baseline was improved upon by combining the score with SVMs trained on linguistic features (Corston-Oliver et al., 2001; Gamon et al., 2005). In **Chapter 4**, we will use a similar framework to identify non-native sentences, which are arguably more challenging to identify than MT output.

### 2.1.2 Essay Scoring

The task in §2.1.1 requires a binary decision: whether a sentence is good enough or not. A natural extension is to give the sentence a score. Research in this direction has mostly been conducted at the document level. Automatic essay scorers, such as (Ishioka & Kameda, 2006) and *e-rater* (Burstein et al., 2004), can provide holistic scores that correlate well with those of human judges. In *e-rater*, features about grammatical errors, usage, mechanics and style are incorporated.

### 2.1.3 Robust Parsing

In some applications, the goal is not to correct grammatical errors, non-native or otherwise, but rather to ensure that they do not hinder the system's downstream processes; in other words, syntactic analysis should be robust. Evaluation is often performed by comparing the hypothesized parse tree, or semantic frame, with the one that would be produced had there been no errors. Ideally, the two should be identical, or at least should differ only within subtrees where errors are present. Research in this area has been applied to noisy output of automatic speech recognition systems (Seneff, 1992a), as well as written texts with errors (Foster, 2003; Bender et al., 2004). Robust parsing will be a significant issue in the correction of verb forms, to be presented in **Chapter 6**.

## 2.2 Machine Translation

Having considered these related tasks, we now turn our attention back to grammar checking, which will be presented in three paradigms. The most recent paradigm is to view the task as monolingual "translation" from "bad English" to "good English", which then allows one to leverage and adapt existing techniques in statistical machine translation (SMT). These techniques generally rely on large parallel corpora of the source (i.e., non-native texts) and target (i.e., corrected version of those texts) languages. Since these resources are not yet available in large quantity, they are simulated by introducing errors into well-formed text. Since the "translation" model is induced by the simulated data, the simulation quality is crucial. The quality depends on both the frequency and authenticity of the errors.

Based on patterns observed in a non-native corpus, errors are artificially inserted into mass nouns (Brockett et al., 2006) in sentences from a well-formed corpus. The resulting parallel corpus is then fed to a phrasal SMT system (Quirk et al., 2005). Without *a priori* knowledge of the error frequency, an equal number of grammatical and ungrammatical sentences were represented in the "source language" in (Brockett et al., 2006). Ideally, the proportion should depend on how likely it is for the errors to appear in the expected input text.

This dissertation will not pursue this research direction for two reasons. First, in the language learning context, one should make corrections only when necessary; in fact, it is sometimes taken as a principle "the corrected form should match the input in all ways except those affected by the correction." (Bender et al., 2004). However, even in "monolingual" MT, output can diverge from the input in significant ways. Second, authentic, parallel learner corpora are not yet available in large enough quantity to use SMT techniques, nor to simulate non-native sentences in a variety of errors. However, in §3.2, we will discuss how an error model can potentially help estimate the error frequency and improve the authenticity of simulated training data.

## 2.3 Parsing

A second paradigm is to attempt a full syntactic analysis of the sentence, and use the parsing process to discover errors and propose corrections. They are to be distinguished from those approaches outlined in §2.4 where, although syntactic parsing is also performed, it is mainly used as a means of extracting features for further consideration.

Among various strategies, we discuss the two most common ones: those that relax rules of the grammar (§2.3.1), and those that explicitly model errors in the grammar (§2.3.2).

### 2.3.1 Constraint Relaxation

One approach to handling a broad class of errors is to relax constraints in a unification framework (Fouvry, 2003; Vogel & Cooper, 1995; Bolioli et al., 1992). The grammar rules, e.g., subject-verb agreement, are progressively relaxed until the sentence can be parsed. In (Fouvry, 2003), the order of relaxation is determined by a notion of information loss.

An attractive feature of this approach is its common mechanism for parsing both grammatical and ungrammatical sentences. When an ungrammatical sentence triggers the relaxation of a constraint, it is easy to identify the error based on this constraint. One disadvantage, however, is that it is not well suited to parsing sentences with missing or extra words. A detailed discussion can be found in (Foster, 2005). Moreover, the constraints need to be pre-defined, limiting the variety of mistakes that can be processed.

### 2.3.2 Mal-Rules

A more popular approach, however, is to use error-production rules, or *mal-rules*, to augment context-free grammars (CFG) to model errorful text. Each of these rules corresponds to a category of grammatical mistakes.

Like constraint relaxation, the main advantage of mal-rules is the ease with which they can generate feedback. However, the mal-rules also need to be pre-defined. As more and more types of errors need to be handled, the grammars become increasingly complicated, exponentially growing the number of ambiguous parses, which can degrade parsing performance. Some prominent systems that utilize mal-rules are:

- ICICLE (*Interactive Computer Identification and Correction of Language Errors*) was designed for American Sign Language signers to learn English (Michaud et al., 2000). Mal-rules were manually derived from analysis of writing samples (McCoy et al., 1996). The system offers broad coverage in English. Instead of proposing corrections, it provides feedback, such as, "*This phrase is missing a preposition in front of it*". It then expects the student to revise his/her writing accordingly, and to re-submit it to the system.

  Improving upon ICICLE, the system in (Michaud & McCoy, 2001) models the user's L2 proficiency based on past interactions. A significant innovation is that, in cases where multiple corrections are possible for an ill-formed sentence, the user model guides the selection of the best correction, and tailors the feedback message according to the perceived L2 proficiency of the user.

- The system in (Foster, 2003) focuses on performance errors, and offers corrections in addition to error detection. The system first attempts to parse with a normal CFG. If no parse is possible, it then proceeds to the error-production grammar.

  Perhaps partly to limit computational complexity, only one type of operation (insertion, deletion, substitution) can be applied to a sentence. A sentence like *But not one of them is capable [of → to] [dealing → deal] with robustness as a whole*, which requires two substitutions, cannot be properly treated.

This limitation may be reasonable for the corpus used, which focuses on "performance errors" rather than language learning errors. In this corpus, only 10.6% of the sentences in the corpus required more than one operation. We suspect that a higher percentage of non-native sentences may require more than one operation.

- In ARBORETUM (Bender et al., 2004), mal-rules are used in conjunction with the LKB parser (Copestake, 2002). A significant novelty is an aligned generation strategy to ensure that the corrected sentence is as close as possible to the input sentence. An evaluation was performed on 221 items in a subset of the Japanese Learners of English corpus (Izumi et al., 2003), of which 80.5% were parsed correctly.

- In one of the earliest applications of this paradigm (Park et al., 1997), a "combinatory categorial grammar" is used. This is a collection of heuristic rules, coupled with a set of "rules" covering anticipated errors, which essentially correspond to what we call "mal-rules".

## 2.4 Natural Language Generation

In §2.3, carefully hand-crafted constraints and mal-rules usually yield high precision. However, they may be less equipped to detect verb form errors within a perfectly grammatical sentence, such as the example given in §10.3.2. Moreover, the grammatical errors to be covered must be anticipated in advance; typically, the set of errors is compiled using anecdotal observations from domain experts.

To eliminate the need to anticipate all errors, a class of grammar checkers adopts methods centered on natural language generation (NLG) techniques. NLG methods may in turn be divided into two main strands. The first (§2.4.1) can be roughly understood as sentence generation from concepts. The system identifies the main content words ("concepts", or "keywords"), and attempts to reconstruct the sentence. This technique is appropriate for sentences that are relatively errorful and need major repair. The second (§2.4.2) views the correction task as classification. It typically considers only those words in the sentence belonging to a certain part-of-speech (e.g., prepositions), and selects the best candidate (e.g., out of all possible prepositions) given the context.

### 2.4.1 Generating a Sentence from Concepts

This approach assumes the user is able to supply the important content words, even if s/he may not be capable of combining them into a fluent sentence. It consists of two main steps. First, the system extracts a *meaning representation* from the input sentence, disregarding grammatical errors and other noise; then, it generates a correct sentence from the meaning representation. Past research has utilized a wide spectrum of such representations, from hierarchical frames to simply a list of content words.

**Semantic Frames** One example is a language learning system for Mandarin at the Spoken Language Systems Group (Seneff, 2006). A parser analyzes the user's utterance to produce a semantic representation (Seneff, 1992b), while ignoring grammatical errors, such as inappropriate usage of Mandarin counters. The NLG system GENESIS (Baptist & Seneff, 2000) then paraphrases the semantic representation into a surface string. This system operates on formal generation rules, which specify the order in which components in the frame are to be processed into substrings. The rules also consult a generation lexicon to obtain multiple word-sense surface-form mappings and appropriate inflectional endings. Flags can be used to vary the surface strings generated by the same frame.

Although not designed for language learning purposes, NITROGEN is a similar NLG system (Langkilde & Knight, 1998). From an "Abstract Meaning Representation", which specifies information such as the agent, patient, temporal or spatial locations, etc., a word lattice is generated using grammar rules. An $N$-best list of surface strings is then produced based on bigram statistics. The memory and speed of this framework has since been improved using a forest ranking algorithm (Langkilde, 2000), and a lexicalized syntax model (Daume et al., 2002). Another comparable system is FERGUS (Bangalore & Rambow, 2000), which uses trigrams.

**Keywords** Rather than a hierarchical frame, a set of key-attribute pairs[1], such as {`city-from=Boston, city-to=Seattle, day-departure=Wednesday`}, is taken as the input in (Ratnaparkhi, 2000). A sentence is generated from these pairs, using word order and word choices learned from training data in the form of an $n$-gram model and a dependency model. In (Uchimoto et al., 2002), the input is a sequence of three Japanese keywords, to which particles and connectives are added to form a complete Japanese sentence. This method was tested on 30 Japanese keyword triplets, whose word orders are assumed to be correct. In an experiment, 19 out of the 30 outputs were judged appropriate; in 6 other cases, the top ten outputs contain at least one that is appropriate.

While ensuring that all necessary corrections are made, it is pedagogically desirable that the resulting sentence remain as close to the input as possible, a principle that is also followed in (Bender et al., 2004). One issue with the above approaches is that the generated sentence could potentially be quite different from the original, since these NLG systems may make substitutions with synonyms or changes in word order even when the original word choice or word order are not necessarily erroneous. In **Chapter 8**, we will present a framework that honors all the content words in the input sentence, then re-generates only the function words and inflectional endings, and finally re-ranks them using a probabilistic context-free grammar.

### 2.4.2 Correction as Classification

Another way of using NLG is to predict one word at a time. Systems in this category aim mostly at input sentences that are already relatively well-formed. An example of a rule-based approach is the use of "templates" (Heidorn, 2000), manually designed by linguists, that recognize specific errors via parse tree patterns, and correct them. Another example is GRANSKA, a Swedish grammar checker (Carlberger et al., 2005). Words in the input sentence are assigned a morphosyntactic tag; then, a rule-matching mechanism tries to match error rules to the sequence of tags, typically a sequence that is rarely seen. The rules then edit the words within this sequence.

In statistical approaches, the prediction is typically framed as a classification task for a specific linguistic class, e.g., prepositions, or a set of pre-determined classes. The classifier is trained on a large body of well-formed English text, with features extracted from the local context of the target word, such as a window of $n$ preceding and subsequent words.

[1] There are up to 26 attributes in the flight domain, where the system was evaluated.

The classifier should, for example, predict the preposition "*in*" to be the most likely choice for the input sentence:

Input: *He participated* **at?** *the competition.*
Corrected: *He participated* **in** *the competition.*

If the predicted preposition differs from the original one, a *confidence* model would then need to decide whether to suggest the correction to the user. In this case, confidence in the predicted preposition "*in*" should be much higher than the original "*at*", and correction would be warranted. The confidence measure can be, for example, the difference between the scores of the top- and second-place candidates given by the NLG component (Chodorow et al., 2007). Such measures have not received much attention in non-native error correction, and further studies are well worth considering (see §10.2.2).

This dissertation will emphasize the generation task rather than confidence measures, as does most research to-date. For example, in (Izumi et al., 2003), a maximum entropy model is trained on words in a window of two words before and after, using the word itself, its POS and roots as features. A decision is made at each word — it may be left alone, replaced by another word, or deleted. An extra word may also be inserted. An evaluation, performed on 13 error categories, achieves 55% precision and 23% recall overall.

Most previous work focuses on one of the following three specific parts-of-speech[2], and so will the rest of this dissertation. We now review the most influential work in each of them.

### Articles

The most studied of the various parts-of-speech is the articles. In one of the earliest studies, decision trees are used to pick either *a/an* or *the*[3] for NPs extracted from the Wall Street Journal (Knight & Chander, 1994). Its motivation was to improve the output of a Japanese-to-English translation system. There are over 30,000 features in the trees, including lexical features, e.g., the two words before and after the NP, and abstract features, e.g., the word after the head noun is a past tense verb. By classifying the more frequent head nouns with the trees, and guessing *the* for the rest, the overall accuracy is 78%.

A memory-based learning approach is applied in (Minnen et al., 2000) to choose between *a/an*, *the* and *null*. Their features are drawn from the Penn Treebank, such as the NP head and its part-of-speech (POS) and functional tags, the category and functional tags of the constituent embedding the NP, and other determiners in the NP. Additional features are drawn from a Japanese-to-English translation system, such as the countability preference and semantic class of the NP head. The best result is 83.6% accuracy.

In **Chapter 7**, we will tackle this problem, with the key insight that performance can be further improved by estimating the initial quality of the input text.

### Prepositions

Usage of prepositions is also a difficult task for non-native speakers to master (Izumi et al., 2004b). Most previous work did not consider syntactic structure, such as the attachment site of prepositional phrases. In (Chodorow et al., 2007), a variety of lexical and POS

[2] Many further restrict attention to a specific aspect, such as mass vs count nouns (Nagata et al., 2006).
[3] The *null* article was not considered.

features, including noun and verb phrases in the vicinity of the preposition, as well as their word lemmas and POS tags, are utilized. The evaluation data consist of newspaper text and a corpus of essays written by 11th and 12th grade students, covering 34 prepositions. A maximum entropy model achieved 69% generation accuracy.

To our best knowledge, the only previous work on preposition generation that utilizes syntactic features is (Felice & Pulman, 2007). In addition to a variety of POS features and some WordNet categories, it also considers grammatical relations (e.g., direct or indirect object) extracted from a parser. The grammatical relation feature is identified as a strong feature. A voted perceptron algorithm, trained on five prepositions, yielded 75.6% accuracy on a subset of the British National Corpus. In **Chapter 5**, we will directly compare the utility of the lexical and syntactic features.

### Verb Forms

The verbs in a 75,000-word corpus of essays, written by French-speaking university students of English in their second year of study, are examined in (Granger, 1999). In this corpus, six types of verb errors (auxiliary, finite/non-finite, concord, morphology, voice and tense) were tagged, and 534 such errors were found. That number decreased to 240 for students in their fourth year of study.

Aside from this study, the correction of verb form usage has received relatively little attention beyond subject-verb agreement, such as in (Park et al., 1997). Perhaps one reason is that errors of verb forms, more than articles and prepositions, tend to affect the performance of parsers trained on well-formed text. We study the correction of verb forms in **Chapter 6**, highlighting the issue of robust feature extraction.

# Chapter 3

# Non-Native Corpora

Earlier efforts in building grammar checkers relied mostly on formal linguistic rules (Park et al., 1997; Thurmair, 1990; Bustamante & Leon, 1996). With the advent of large corpora and statistical machine learning methods, they can now be complemented with data-driven approaches. Since this dissertation draws upon methods from statistical natural language processing, we prepare the stage by discussing the data used in both the training and evaluation stages. The research community in non-native text processing, a relatively new area, has not yet settled on a benchmark set of data. This chapter sketches the main resources that are available.

We first give an overview of the data sets (§3.1), both standard and non-native corpora, pointing out the trade-offs among them and relevant issues in evaluation. We then analyze one of our evaluation corpora (§3.2), the Japanese Learners of English (JLE) Corpus, illustrating tendencies in errors associated with articles and prepositions.

## 3.1 Overview of Corpora

### 3.1.1 Standard Corpora

Grammar checking systems should, ideally, be evaluated on a corpus of learners' writing, annotated with acceptable corrections. However, since such corpora are expensive to compile, many researchers have instead approximated the task by measuring the accuracy of predicting what a native writer originally wrote in well-formed text (Knight & Chander, 1994; Minnen et al., 2000).

An obvious advantage of this type of evaluation is that it does not require any non-native corpus; standard corpora of well-formed English text would suffice. However, it can have two adverse effects on evaluation. On the one hand, performance may be overestimated, if the feature extraction process is not robust on non-native, noisy text, which the system will eventually have to process. On the other hand, performance may also be underestimated. The evaluation effectively makes the assumption that there is one correct form of native usage per context, which may not always be the case.

A possible remedy is to artificially inject errors into the well-formed sentences. Now, without any manual effort, both a "non-native" corpus and its "correct" version (i.e., the original text) are available. Moreover, the frequencies and types of errors can be carefully controlled. This synthetic parallel corpus can be used in the machine translation paradigm (§2.2) or other machine learning methods (Sjöbergh & Knutsson, 2005). The "one-answer"

| Corpus | Error-<br>tagged | Target<br>Words | Spelling<br>Mistakes | Native<br>Language | Size |
|---|---|---|---|---|---|
| JLE | yes | yes | no | Japanese | 2 million words |
| HKUST | yes | no | yes | Cantonese | 38219 words |
| HEL | no | yes | yes | Japanese | 11156 words |
| ICLE | no | no | yes | various European | 2 million words |
| Foster | no | yes | yes | n/a | 923 sentences |

Table 3.1: Properties of the non-proprietary corpora (see §3.1.2 for full names) that have been used in grammar correction research, either in this dissertation or elsewhere. A corpus is considered *error-tagged* if the parts-of-speech of the source and target words are either explicitly or at least indirectly identified. *Target Words* refer to explicit correction of the source text. Data that are not corrected would be difficult for automatic evaluation; however, as long as it is error-tagged, it is still useful for finding typical language learner mistakes. All corpora that derive originally from texts written by non-native speakers contain spelling mistakes; one exception is JLE, which consists of transcripts of spoken utterances rather than texts. For a discussion on spelling errors, see §1.2.

assumption remains[1], and how well the simulated errors match those in real input is a critical factor in the performance of the system.

The evaluation in **Chapters 5** and **8** will be conducted in the above manner.

### 3.1.2 Non-native Corpora

While many detailed analyses have been performed on a small number of subjects[2] or a specific grammatical construction[3], few large-scale non-native corpora are available. We now describe some of them, with a summary in Table 3.1. We begin with two that will be utilized in this dissertation, but also include others that have been used by other researchers and might be useful in future work:

**Japanese Learners of English** (JLE) The National Institute of Information and Communications Technology (NICT) in Japan assembled this corpus (Izumi et al., 2004b), which consists of 2 million words from 1200 Japanese students. These students were interviewed for the Standard Speaking Test (SST), an English-language proficiency test conducted in Japan, and graded on a scale between 1 and 9. The 15-minute interview includes informal chat, picture description, role-playing and story telling. These interviews were transcribed; for 167 of these interviews, grammatical errors in 45 categories have been manually annotated and corrected. For example,

> *I lived in* <AT crr="">*the*</AT> *New Jersey.*

where AT is the error label for an article error, "the" is the word that should be removed, and crr is the word (in this case, the null string) that should be inserted. Sentence segmentation (Reynar & Ratnaparkhi, 1997) was performed on the interviewee turns in these transcripts, discarding sentences with only one word. This procedure yielded 15637 sentences, with over 153K words. The three most frequent

[1] We do not address this issue, but will discuss possible ways to tackle it in §10.4.1.

[2] e.g., (Hakuta, 1976).

[3] e.g., (Habash, 1982; Liu & Gleason, 2002).

error classes are articles, noun number and prepositions, followed by a variety of errors associated with verbs.

Although this corpus consists of speech transcripts, the content is much closer to formal English than, say, Switchboard. First, the examinees have a strong incentive to use formal, correct English. Second, the transcripts mark all self-corrections, back-channel expressions, etc. Some examples of article and preposition errors drawn from this corpus are shown in Tables 3.2, 3.4 and 3.6. The evaluation in **Chapters 4** and **6** will utilize this corpus.

**Hong Kong University of Science and Technology** (HKUST) Professor John Milton has collected[4] a set of essays written by Cantonese-speaking students at HKUST. Topics include romantic relationships among teenagers, and institutes of higher learning in Hong Kong. The corpus contains a total of 2556 sentences. They tend to be longer and have more complex structures than their counterparts in the JLE. Target words (i.e., corrections) are not provided[5]; however, the parts-of-speech in all essays are manually tagged for the original words, and for the intended target words. For example:

> *They{pro} regard{vps} examination{ns#npl\#art} as_if{conj} it{pro\#agr} were{subjt} a{arti#r} hell{nu}.*

The evaluation in **Chapter 6** will utilize this corpus.

**Test of English as a Foreign Language** (TOEFL) This popular test, administered by the Educational Testing Service (ETS), contains an essay writing section. A set of 150 TOEFL essays, written by native speakers of Chinese, Japanese and Russian, were used in a study of article correction (Han et al., 2004). A similar set was used in a study on prepositions (Tetreault & Chodorow, 2008b). Unfortunately, these essays are proprietary to ETS and are not available to the public.

**Cambridge Learner Corpus** Like TOEFL, this corpus is culled from English-proficiency examinations. Over 30 million words have been annotated with grammatical errors[6]. Part of the corpus was used in a study on preposition correction (Felice & Pulman, 2008).

**Chinese Learner English Corpus** (CLEC) This corpus consists of writings of Chinese students from secondary school up to university level (Gui & Yang, 2003). One million words have been error tagged. The corpus is used in the evaluation of (Sun et al., 2007).

**English Taiwan Learners Corpus** This corpus (Wible et al., 2001) is partially annotated by English teachers in Taiwan with comments such as "*word choice*", "*delete this*" and "*word form*". These comments are not sufficient for reconstruction of the reference sentence.

**Foster** A 923-sentence corpus collected from newspaper, personal writing, etc (Foster, 2005). Most errors are performance errors.

[4] Personal communication. The corpus is not publicly available.

[5] Implications on our evaluation procedure are discussed in §6.5.4.

[6] http://www.cambridge.org/elt/corpus/learner_corpus.htm

**Hiroshima English Learners' Corpus** (HEL) Prof. Shogo Miura collected English translations of Japanese sentences by Japanese-speaking students at the University of Hiroshima, Japan.

**International Corpus of Learner English** (ICLE) This corpus consists of over 2 million words from 3640 essays on various topics written by students of English. They are divided into 11 subcorpora according to the writer's mother tongue, mostly European, and are further classified according to L2 exposure, age, gender, etc. It is not error-tagged.

Some of the above corpora, such as TOEFL and Cambridge Learner Corpus, are proprietary; others, including ICLE and CLEC, have very limited annotation. This leaves the JLE as an ideal candidate for corpus evaluation, and we will now present a detailed analysis for this corpus.

## 3.2 Corpus Analysis

This section gives a flavor of the types of errors in the Japanese Learners of English (JLE) corpus, which will serve as evaluation data in **Chapters 4** and **6**. In particular, we focus on articles and prepositions, two of the most frequent error categories. Our analyses will be based on parse trees automatically produced by (Collins, 1999).

To the extent that these errors are specific to native Japanese speakers, the utility of the acquired error model may be limited outside this linguistic community. However, our analysis techniques can potentially be applied to corpora from other communities as they become available.

### 3.2.1 Analysis on Articles

A noun may be preceded by a determiner, most commonly an article, i.e., "*a*", "*an*"[7] or "*the*". In the corrected version of the JLE corpus, nouns have no determiner ("*null*") 41% of the time, and have "*the*" and "*a*" 26% and 24% of the time, respectively. The majority of the remaining 9% are quantifiers such as "*some*", "*that*", "*this*", and "*those*".

#### Confusion among Articles

The overall distribution of article deletions, insertions and substitutions (Table 3.2) shows that deletion is the overwhelming type of error. This may be expected, since there is no functional equivalent of articles in the Japanese language. Among deletion errors, "*a*" is more often deleted, even though it occurs almost as frequently as "*the*".

It would not be reasonable, however, to assume that non-native speakers omit articles randomly, regardless of context. When conditioned on the head noun of the article, the error type is no longer always dominated by deletions. For example, with the word "*police*", deletions are less prevalent than substitutions, e.g., "*I called [the→a] police*".

The noun most frequently associated with each error type, based on absolute counts, is shown in Table 3.2. This "top offender" list is inevitably influenced by the particular vocabulary distribution of the corpus. The fact that the words "*dinner*" and "*movie*" emerge

[7] The distinction between "*a*" and "*an*" can be easily resolved and is not considered further. Both will henceforth be represented as "*a*".

| Type | Percent | Error | Percent | Example |
|---|---|---|---|---|
| Del | 69.4% | [*a*→*null*]<br>[*the*→*null*] | 41.1%<br>28.4% | Three people go to see [*a*→*null*] **movie** |
| Ins | 17.3% | [*null*→*the*]<br>[*null*→*a*] | 12.9%<br>4.4% | We had [*null*→*a*] **dinner** at a restaurant |
| Sub | 13.3% | [*a*→*the*]<br>[*the*→*a*] | 9.8%<br>3.5% | How about going to see [*a*→*the*] **movie** |

Table 3.2: *Relative frequencies of deletions (Del), insertions (Ins) and substitutions (Sub) of articles, out of a total of 3382. Each error type is broken down into the specific errors. An example is drawn from the noun (bolded) most frequently involved in each type. The notation [⟨crr⟩ → ⟨err⟩] will be used to indicate that the correct word, ⟨crr⟩, is mistakenly replaced with the word ⟨err⟩ by the non-native speaker.*

on top likely has to do with the conversation topics at the examination. This issue may be addressed in two ways, one geared towards grammar checkers (§3.2.1), the other towards CALL systems (§3.2.1).

**Error Likelihood** To decide whether to propose a correction for the article, it would be useful to measure how error-prone the head noun is. One could start by normalizing the article-error count of the noun by the number of times the noun appears in the corpus. However, its error-proneness may vary significantly depending on the article expected. For example, the noun "*place*" has the correct article 62% of the time overall, but for the subset with the article "*a*", the figure drops to 26%.

Thus, the noun by itself would not suffice as the *context* for article errors; it should rather be considered in conjunction with its expected article. The error likelihood of an article-noun pair is, then, the number of times the pair contains an article error, divided by its total number of appearances.

The article-noun pairs with the highest error likelihoods are shown in Table 3.3. Reflecting the overall tendency of underusing articles, the top pairs tend to have the article "*a*" or "*the*", while the ones with "*null*" dominate the bottom of the list (not shown). The most error-prone pair with "*null*" is ⟨*null*,**news**⟩, e.g., "*I read [null→a] news on the Internet*". In grammar checkers, for predicting the likelihood of a particular error, these likelihoods can be easily turned into $P(\langle err\rangle|\langle context\rangle)$ by specifying the identity of the error given each context.

**Frequency in General Domain** Of most interest to users of CALL systems are those article-noun pairs that are not only error-prone, but also common in everyday usage. For the latter criterion, the AQUAINT newswire corpus is utilized to estimate the "unigram" probability of each pair, i.e., the proportion of that pair among all article-noun pairs in the corpus. When multiplied with the error likelihood (§3.2.1), the product may be interpreted as the probability of that pair occurring with an article error, had the AQUAINT corpus been written by a non-native speaker.

The top two pairs with the highest estimated frequencies for each article are listed in Table 3.3. These words tend to be both common and susceptible to article errors.

| Error Likelihood | |
|---|---|
| Context | Example |
| ⟨*a*,**theater**⟩ | Let us go to [*a*→*null*] movie **theater** |
| ⟨*a*,**area**⟩ | It's [*a*→*null*] residential **area** |
| ⟨*a*,**concert**⟩ | They offered [*a*→*null*] free **concert** |
| ⟨*a*,**club**⟩ | I want to belong to [*a*→*null*] basketball **club** |
| ⟨*the*,**guitar**⟩ | I like to play [*the*→*null*] **guitar** |

| Error Likelihood weighted w/ Frequency in General Domain | |
|---|---|
| Context | Example |
| ⟨*the*,**market**⟩ | ... is the number one brand in [*the*→*null*] Japanese shampoo and conditioner **market** |
| ⟨*the*,**team**⟩ | [*The*→*null*] England football **team** has very famous and good players |
| ⟨*a*,**company**⟩ | She works for [*a*→*null*] real estate **company** |
| ⟨*a*,**day**⟩ | it was [*a*→*null*] rainy **day** |
| ⟨*null*,**people**⟩ | I think [*null*→*the*] young Japanese **people** think it's cool |
| ⟨*null*,**one**⟩ | I'm going to visit [*null*→*a*] **one** of my friends |

Table 3.3: *The top table lists article-noun pairs with the highest error likelihoods (§3.2.1). The bottom table lists, for each article, the top two pairs when the likelihood is weighted with frequency in a general domain (§3.2.1).*

#### Confusion between Articles and non-Articles

Past research has focused exclusively on confusions among the articles. Although confusions between articles and other determiners are less numerous, they also exhibit some interesting trends. The single most frequent error is the use of "*some*" in place of an indefinite article, e.g. "*She selected [a→some] tie for him*". Not far behind are errors with possessive pronouns, e.g. "*I believe you understand [the→my] reason why I can't join your party*", and in the reverse direction, "*I like [my→the] children*". Other frequent errors are [*the*→*that*], and [*the*→*this*].

### 3.2.2 Analysis on Prepositions

The most frequent prepositions in the JLE corpus are "*in*" (23%), "*of*" (18%), "*for*" (12%), "*on*" (8%), "*with*" (8%) and "*at*" (8%). A preposition "expresses a relation between two entities, one being that represented by the prepositional complement, the other by another part of the sentence." (Quirk et al., 1985) The *prepositional complement* is typically a noun phrase under the prepositional phrase (PP). The other entity is known as the *lexical head*, which can be a verb, noun or adjectival phrase preceding the PP. In Table 3.4, the bolded words in the examples are prepositional complements, while in Table 3.6, they are lexical heads.

To determine which one of these two entities provides a better context for preposition errors, it is helpful to distinguish between argument and adjunct PPs[8]. Generally speaking,

[8] Preposition errors in JLE are tagged under two categories, "PRP_LXC1" and "PRP_LXC2", which roughly

| Type | Percent | Error | Percent | Example |
|---|---|---|---|---|
| Del | 53.8% | [*in*→*null*]<br>[*at*→*null*]<br>[*on*→*null*] | 14.6%<br>8.3%<br>7.5% | A car is parked [*on*→*null*] the **side** of the road |
| Sub | 46.2% | [*on*→*in*]<br>[*at*→*in*]<br>[*in*→*at*] | 5.1%<br>3.7%<br>2.8% | I study [*at*→*in*] the **university** |

Table 3.4: *Relative frequencies of deletions (Del) and substitutions (Sub) of prepositions in adjunct prepositional phrases, out of a total of 780. An example is drawn from the prepositional complement (bolded) most frequently involved in each error type.*

| Error Likelihood | |
|---|---|
| Context | Example |
| ⟨*along*,**street**⟩ | I walk [*along*→*null*] the **street** |
| ⟨*in*,**team**⟩ | One of the most famous baseball players in Japan [*in*→*at*] the same **team** ... |
| ⟨*on*,**birthday**⟩ | I will go to your place [*on*→*at*] my next **birthday** |

| Error Likelihood weighted w/ Frequency in General Domain | |
|---|---|
| Context | Example |
| ⟨*at*,**university**⟩ | I studied [*at*→*in*] the **university** |
| ⟨*at*,**end**⟩ | [*At*→*In*] the **end** of the month, ... |
| ⟨*for*,**year**⟩ | I have taken lessons [*for*→*null*] about ten **years** |

Table 3.5: *The top table lists preposition-complement pairs (adjuncts) with the highest error likelihoods (§3.2.1). The bottom table lists the top pairs after weighting with frequency in a general domain (§3.2.1).*

a preposition in an argument PP is closely related to the lexical head, serving as its argument marker; a preposition in an adjunct PP is less closely related to it, serving merely as a modifier. Contrast the two sentences "*She looked at a monkey*" and "*She came at night*". In the first sentence, "*at*" is closely related to the lexical head "*look*", marking "*a monkey*" as its argument. In the second, "*at night*" is an adjunct modifier of the lexical head "*came*", and is not an integral part of the phrase.

These distinctions are not always clear-cut but, for arguments, the lexical head generally gives a better context for the preposition. Given "*She looked [at→null] a monkey*", the error seems specific to the lexical head "*look*", and could have occurred with any other animal. In contrast, for adjuncts, the appropriate context word is the prepositional complement, not the lexical head. Consider the error in the sentence "*She came [at→null] night*". The deletion error of "*at*" is clearly tied to "*night*", and could have occurred with lexical heads other than "*came*".

correspond to this distinction. Insertion errors are also included in these categories, but they will be analyzed separately. See comments in §3.2.2.

| Type | Percent | Error | Percent | Example |
|---|---|---|---|---|
| Del | 71.7% | [*to*→*null*]<br>[*at*→*null*]<br>[*about*→*null*] | 42.2%<br>4.7%<br>3.7% | I want to **go** [*to*→*null*] Chicago |
| Sub | 28.3% | [*with*→*to*]<br>[*to*→*for*]<br>[*to*→*with*] | 3.0%<br>2.3%<br>1.9% | I'd like to **exchange** this [*with*→*to*] another one |

Table 3.6: *Frequencies of deletions (Del) and substitutions (Sub) of prepositions in argument prepositional phrases, out of a total of 427. An example is drawn from the lexical head (bolded) most frequently involved in each error type.*

### Adjuncts

In Japanese, case particles can play a role similar to English prepositions, although no simple mapping[9] is possible (Suzuki & Toutanova, 2006). These differences could have contributed to some of the preposition errors. Table 3.4 presents some error statistics for the adjuncts.

Overall, there are more deletions, but some prepositional complements are more prone to substitutions. For example, four-fifths of the preposition errors associated with "*university*" are substitutions. Table 3.4 gives an example from the prepositional complement that suffers most often from each error type.

As in §3.2.1, to illustrate potential applications in grammar checkers and CALL systems, error likelihoods[10] are computed and weighted with frequencies in the general domain. Among the preposition-complement pairs with at least three appearances in the corpus, those with the highest likelihoods are shown in Table 3.5.

### Arguments

Statistics on deletions and substitutions of prepositions in argument PPs are shown in Table 3.6. The proportion of deletions (71.7%) is substantially higher for arguments than for adjuncts. The most prominent error, [*to*→*null*], is due mostly to the one verb "*go*", which alone is responsible for a third of the counts. Using the same procedure as for the adjuncts, the most error-prone verb-preposition pairs are listed in Table 3.7.

### Insertion Errors

Preposition insertion errors belong to neither the adjunct nor argument categories, since no PP is actually needed. They typically occur when an adverb or adjective follows a verb, such as "*She asked him to bring the cat [null→to] home.*" or "*lives [null→in] next door*". The three most frequently inserted prepositions are "*to*", "*in*", and "*with*".

[9] For example, the object particle "*wo*" normally does not correspond to any English word when marking a direct object in Japanese. However, in some contexts, it can necessitate a preposition in English, e.g., "*walk along a street*" from "*michi wo aruku*".

[10] Since argument and adjunct PPs cannot yet be distinguished automatically with high accuracy, the counts are normalized without this distinction, possibly leading to overestimated denominators.

| Error Likelihood | |
|---|---|
| Context | Example |
| ⟨**bump**,*into*⟩ | A motorbike **bumped** [*into*→*to*] my car |
| ⟨**ask**,*about*⟩ | When the officer came to **ask** [*about*→*null*] the situation ... |
| ⟨**graduate**,*from*⟩ | He just **graduated** [*from*→*null*] university |

| Error Likelihood weighted w/ Frequency in General Domain | |
|---|---|
| Context | Example |
| ⟨**look**,*at*⟩ | She was **looking** [*at*→*null*] a monkey |
| ⟨**ask**,*for*⟩ | He **asked** [*for*→*null*] a table near the window |
| ⟨**come**,*to*⟩ | Last October , I **came** [*to*→*in*] Tokyo |

Table 3.7: *The top table lists verb-preposition pairs (arguments) with the highest error likelihoods (§3.2.1). The bottom table lists the top pairs when weighted with frequency in a general domain (§3.2.1).*

## 3.3 Summary

This chapter started with a review of corpora of non-native texts, as well as standard English corpora, and a discussion on how they might be used in empirical approaches of non-native error correction.

This review was followed by a detailed analysis of article and preposition errors in the JLE corpus, the largest annotated, non-native corpus available to the public. Each error type is conditioned on the most salient word in the context. To our knowledge, there has not been any reported effort to build non-native language models through automatic, detailed analysis of grammatical errors in a non-native corpus. This analysis illustrates a step towards this direction of research. All three paradigms of grammatical error correction (§2.2-§2.4) can benefit from the statistics derived from such an analysis. We will elaborate on these applications as possible future work in §10.3.2.

Having assembled our data, we now apply them on a series of non-native text processing tasks, starting with detection of errors (**Chapter 4**), before moving on to correction of errors in prepositions, verb forms and articles.

# Chapter 4

# Non-native Error Detection

For non-native speakers writing in a foreign language, feedback from native speakers is indispensable. While humans are likely to provide higher-quality feedback, a computer system can offer better availability and privacy. A system that can distinguish *non-native* ("ill-formed") English sentences from *native* ("well-formed") ones would provide valuable assistance in improving their writing.

## 4.1 Research Questions

Classifying a sentence into discrete categories can be difficult: a sentence that seems fluent to one judge might not be good enough to another. An alternative is to rank sentences by their relative fluency. This would be useful when a non-native speaker is unsure which one of several possible ways of writing a sentence is the best.

We therefore formulate two tasks on this problem. The **classification** task gives one sentence to the system, and asks whether it is native or non-native. The **ranking** task submits sentences with the same intended meaning, and asks which one is best.

To tackle these tasks, hand-crafting formal rules would be daunting (§2.3). Statistical methods, however, require a large corpus of non-native writing samples, which can be difficult to compile (§2.2). Since machine-translated (MT) sentences are readily available in abundance, we wish to address the question of whether they can substitute as training data. We first provide background on related research (§4.2), then describe our experiments (§4.3-§4.4).

## 4.2 Related Research

Previous research has paid little attention to ranking sentences by fluency. As for classification, one line of research in MT evaluation is to evaluate the fluency of an output sentence without its reference translations, such as in (Corston-Oliver et al., 2001) and (Gamon et al., 2005). Our task here is similar, but is applied on non-native sentences, arguably more challenging than MT output.

Evaluation of non-native writing has followed two trends. Some researchers explicitly focus on individual classes of errors, e.g., mass vs count nouns in (Brockett et al., 2006) and (Nagata et al., 2006). Others implicitly do so with hand-crafted rules, via templates (Heidorn, 2000) or mal-rules in context-free grammars, such as (Michaud et al., 2000) and (Bender et al., 2004). Typically, however, non-native writing exhibits a wide variety

| Type | | Sentence |
|---|---|---|
| Native | Human | New York and London stock markets went up |
| Non-native | Human | The stock markets in New York and London were increasing together |
| | MT | The same step of stock market of London of New York rises |

Table 4.1: *Examples of sentences translated from a Chinese source sentence by a native speaker of English, by a non-native speaker, and by a machine translation system.*

| Data Set | Corpus | # sentences (for classification) | | | # pairs (for ranking) |
|---|---|---|---|---|---|
| | | total | native | non-native | |
| MT train | LDC{2002T01, 2003T18, 2006T04} | 30075 | 17508 | 12567 | 91795 |
| MT dev | LDC2003T17 (*Zaobao* only) | 1995 | 1328 | 667 | 2668 |
| MT test | LDC2003T17 (*Xinhua* only) | 3255 | 2184 | 1071 | 4284 |
| JLE train | Japanese Learners of English | 9848 | 4924 | 4924 | 4924 |
| JLE dev | | 1000 | 500 | 500 | 500 |
| JLE test | | 1000 | 500 | 500 | 500 |

Table 4.2: *Data sets used in this chapter.*

of errors, in grammar, style and word collocations. In this research, we allow unrestricted classes of errors[1], and in this regard our goal is closest to that of (Tomokiyo & Jones, 2001). However, they focus on non-native speech, and assume the availability of non-native training data.

## 4.3 Experimental Set-Up

### 4.3.1 Data

Our data consists of pairs of English sentences, one native and the other non-native, with the same "intended meaning". In our MT data (MT), both sentences are translated, by machine or human, from the same sentence in a foreign language. In our non-native data (JLE), the non-native sentence has been edited by a native speaker[2]. Table 4.1 gives some examples, and Table 4.2 presents some statistics.

MT (Multiple-Translation Chinese and Multiple-Translation Arabic corpora) English MT output, and human reference translations, of Chinese and Arabic newspaper articles.

JLE (Japanese Learners of English Corpus) Transcripts of Japanese examinees in the Standard Speaking Test. False starts and disfluencies were then cleaned up, and grammatical mistakes tagged (Izumi et al., 2003). The speaking style is more formal than spontaneous English, due to the examination setting.

### 4.3.2 Machine Learning Framework

SVM-Light (Joachims, 1999), an implementation of Support Vector Machines (SVM), is used for the classification task.

[1] For spelling mistakes, see a discussion in §1.2.

[2] The nature of the non-native data constrains the ranking to two sentences at a time.

For the ranking task, we utilize the ranking mode of SVM-Light. In this mode, the SVM algorithm is adapted for learning ranking functions, originally used for ranking web pages with respect to a query (Joachims, 2002). In our context, given a set of English sentences with similar semantic content, say $s_1, \ldots, s_n$, and a ranking based on their fluency, the learning algorithm estimates the weights $\vec{w}$ to satisfy the inequalities:

$$\vec{w} \cdot \Phi(s_j) > \vec{w} \cdot \Phi(s_k) \tag{4.1}$$

where $s_j$ is more fluent than $s_k$, and where $\Phi$ maps a sentence to a feature vector. This is in contrast to standard SVMs, which learn a hyperplane boundary between native and non-native sentences from the inequalities:

$$y_i(\vec{w} \cdot \Phi(s_i) + w_0) - 1 \geq 0 \tag{4.2}$$

where $y_i = \pm 1$ are the labels. Linear kernels are used in our experiments, and the regularization parameter is tuned on the development sets.

### 4.3.3 Features

The following features are extracted from each sentence. The first two are real numbers; the rest are indicator functions of the presence of the lexical and/or syntactic properties in question.

`Ent` Entropy[3] from a trigram language model trained on 4.4 million English sentences with the SRILM toolkit (Stolcke, 2002). The trigrams are intended to detect local mistakes.

`Parse` Parse score from Model 2 of the statistical parser (Collins, 1997), normalized by the number of words. We hypothesize that non-native sentences are more likely to receive lower scores.

`Deriv` Parse tree derivations, i.e., from each parent node to its children nodes, such as S → NP VP. Some non-native sentences have plausible $N$-grams, but have derivations infrequently seen in well-formed sentences, due to their unusual syntactic structures.

`DtNoun` Head word of a base noun phrase, and its determiner, e.g., (*the, markets*) from the human non-native sentence in Table 4.1. The usage of articles has been found to be the most frequent error class in the JLE corpus (Izumi et al., 2003).

`Colloc` An in-house dependency parser extracts five types of word dependencies[4]: subject-verb, verb-object, adjective-noun, verb-adverb and preposition-object. For the human non-native sentence in Table 4.1, the unusual subject-verb collocation "*market increase*" is a useful clue in this otherwise well-formed sentence.

## 4.4 Analysis

### 4.4.1 An Upper Bound

To gauge the performance upper bound, we first attempt to classify and rank the MT test data, which should be less challenging than non-native data. After training the SVM on

[3] Entropy $H(x)$ is related to perplexity $PP(x)$ by the equation $PP(x) = 2^{H(x)}$.

[4] Proper nouns and numbers are replaced with special symbols. The words are further stemmed using Porter's Stemmer.

MT train, classification accuracy on MT test improves with the addition of each feature, culminating at 89.24% with all five features. This result compares favorably with the state-of-the-art[5]. Ranking performance reaches 96.73% with all five features.

We now turn our attention to non-native test data, and contrast the performance on JLE test using models trained by MT data (MT train), and by non-native data (JLE train).

| Test Set:<br>JLE test | Train Set<br>MT train | <br>JLE train |
|---|---|---|
| Ent+<br>Parse | 57.2<br>(+) 48.6<br>(-) 65.8 | 57.7<br>(+) 70.6<br>(-) 44.8 |
| +Deriv | 58.4<br>(+) 54.6<br>(-) 62.2 | 64.7<br>(+)72.2<br>(-) 57.2 |
| +DtNoun | **59.0**<br>(+) 57.6<br>(-) 60.4 | **66.4**<br>(+) 72.8<br>(-) 60.0 |
| +Colloc | 58.6<br>(+) 54.2<br>(-) 63.2 | 65.9<br>(+) 72.6<br>(-) 59.2 |

Table 4.3: *Classification accuracy on JLE test. (-) indicates accuracy on non-native sentences, and (+) indicates accuracy on native sentences. The overall accuracy is their average.*

| Test Set: | Train Set | |
|---|---|---|
| JLE test | MT train | JLE train |
| Ent+Parse | 72.8 | 71.4 |
| +Deriv | 73.4 | 73.6 |
| +DtNoun | 75.4 | 73.8 |
| +Colloc | **76.2** | **74.6** |

Table 4.4: *Ranking accuracy on JLE test.*

### 4.4.2 Classification

As shown in Table 4.3, classification accuracy on JLE test is higher with the JLE train set (66.4%) than with the larger MT train set (59.0%). The SVM trained on MT train consistently misclassifies more native sentences than non-native ones. One reason might be that speech transcripts have a less formal style than written news sentences. Transcripts of even good conversational English do not always resemble sentences in the news domain.

[5] Direct comparison is impossible since the corpora were different. (Corston-Oliver et al., 2001) reports 82.89% accuracy on English software manuals and online help documents, and (Gamon et al., 2005) reports 77.59% on French technical documents.

### 4.4.3 Ranking

In the ranking task, the relative performance between MT and non-native training data is reversed. As shown in Table 4.4, models trained on `MT train` yield higher ranking accuracy (76.2%) than those trained on `JLE train` (74.6%). This indicates that MT training data can generalize well enough to perform better than a non-native training corpus of size up to 10000.

The contrast between the classification and ranking results suggests that train/test data mismatch is less harmful for the latter task. Weights trained on the classification inequalities in (4.2) and the ranking inequalities in (4.1) both try to separate native and MT sentences maximally. The absolute boundary learned in (4.2) is inherently specific to the nature of the training sentences, as we have seen in §4.4.2. In comparison, the relative scores learned from (4.1) have a better chance to carry over to other domains, as long as some gap still exists between the scores of the native and non-native sentences.

## 4.5 Summary

This chapter addresses non-native error detection, a subtask in non-native error correction. We explored two tasks in sentence-level fluency evaluation: ranking and classifying native vs. non-native sentences. In an SVM framework, we examined how well MT data can replace non-native data in training.

For the classification task, training with MT data is less effective than with non-native data. However, for the ranking task, models trained on publicly available MT data generalize well, performing as well as those trained with a non-native corpus of size 10000.

While it is useful to be alerted to possible errors in a sentence, it is often desirable, especially for language students at the elementary stage, to receive suggestions on how to correct them. This is the challenge to which we now turn. We will be using prepositions (**Chapter 5**), verb forms (**Chapter 6**) and articles (**Chapter 7**) to highlight issues in the use of linguistic features, parser robustness and personalization.

# Chapter 5

# Use of Linguistic Analysis: The Case of Prepositions

As discussed in §1.2, preposition usage is among the more frequent types of error made by non-native speakers of English. A system that can automatically detect and correct preposition usage would be of much practical and educational value. The focus of this chapter is on the *preposition generation* task, of which Table 5.1 provides some examples; in particular, we are interested in whether linguistic analysis can improve generation accuracy. We will answer this question by comparing the effectiveness of different kinds of features for this task.

After a motivation of this study (§5.1), previous work is summarized (§5.2) and contrasted with our proposed features (§5.3). The machine learning framework and experimental results are presented in §5.4-§5.6.

## 5.1 Introduction

The features considered in previous research on preposition generation may be divided into three main types. Lexical features, such as word *n*-grams within a window around the preposition; the parts-of-speech (POS) tags of these words; and syntactic features, such as the word modified by the prepositional phrase (PP), or grammatical relations between pairs of words.

Unfortunately, no direct comparison has been made between these different kinds of features. Intuitively, syntactic features should be helpful in choosing the preposition. How much gain do they offer? Does their utility vary for different kinds of PP, or depend on the size of the training set? This chapter seeks to fill this gap in the literature by comparing a lexical baseline feature set with a syntactic feature set that incorporates PP attachment information.

Our key finding is that PP attachment information can improve generation performance. In a memory-based learning approach, this improvement is especially notable when the training data are sparse.

### 5.1.1 Theoretical Motivations

Linguistic analyses suggest that the attachment site of the PP, as well as the argument/adjunct distinction, play significant roles in the choice of preposition. This section

Table 5.1: Example sentences for preposition generation. The lexical head of the PP is in *italics* and the prepositional complement is **bolded**.

| Sent # | Input Text | Output |
|---|---|---|
| 1 | Pierre Vinken *joined* the board ____ a nonexecutive **director**. | as |
| 2 | The $2.5 million Byron plant was *completed* ____ **1985**. | in |
| 3 | The average maturity for funds *open* only ____ **institutions**, ... | to |
| 4 | Newsweek announced new advertising *rates* ____ **1990**. | for |

provides some linguistic background to motivate our research question, and also defines some terminology to be used in the rest of the chapter. The material in this section is based on (Quirk et al., 1985), unless otherwise stated.

### Attachment

A preposition "expresses a relation between two entities, one being that represented by the prepositional complement, the other by another part of the sentence." The *prepositional complement* is, in most cases, a noun phrase[1]. That "another part of the sentence" can be a verb-, noun- or adjectival phrase. The PP is said to be *attached* to this phrase, and the head word of this phrase is called the *lexical head* of the PP.

For example, in sentence #1 in Table 5.1, the preposition "*as*" expresses the relation between the prepositional complement "*director*" and its lexical head, the verb "*joined*". Knowing that the PP is attached to "*joined*", rather than to "*board*", would clearly help predict the preposition "*as*".

### Argument/Adjunct Distinction

The relevance of the lexical head for the choice of preposition may depend on its relation with the prepositional complement. One aspect of this relation is the argument/adjunct distinction. In principle, "arguments depend on their lexical heads because they form an integral part of the phrase. Adjuncts do not."(Merlo & Esteve Ferrer, 2006). The preposition in an argument PP is thus more closely related to the lexical head than one in an adjunct PP. The distinction can be illustrated in two of the syntactic functions of PPs:

- **Complementation**: The preposition marks an argument of the lexical head. The prepositions "*as*" in sentence #1 in Table 5.1 is such an example. In this usage, the PP is said to be an *argument.*

- **Adverbial**: The PP serves as a modifier to its lexical head. The phrase "*in 1985*" in sentence #2 is one example. In this usage, the PP is an *adjunct.*

The argument/adjunct distinction has been shown to be helpful in PP attachment (Merlo & Esteve Ferrer, 2006); it may also be relevant in preposition generation.

[1] Some prepositions function as particles in phrasal verbs, e.g., "give *up*" or "give *in*". We view these particles as part of the verb and do not attempt generation.

### 5.1.2 Practical Motivations

In addition to the linguistic motivations discussed above, the use of PP attachment and the argument/adjunct distinction can also improve the user experience of a grammar checking system.

For a language learner, the system should serve not merely a practical, but also an educational, purpose. Besides having a wrong preposition detected and corrected, the user would also like to learn the reason for the correction, such as, "the verb $X$ requires the preposition $Y$". Without considering PP attachment, this kind of feedback is difficult.

By making known its assumptions on the attachment site, the grammar checker also enhances its transparency. If the user spots an attachment error, for example, s/he may choose to inform the system and can then expect a better prediction of the preposition.

## 5.2 Previous Work

Previous research on preposition generation and error detection has considered lexical, part-of-speech (POS) and syntactic features.

### 5.2.1 Lexical and POS Features

A rule-based approach using lexical features is employed in (Eeg-Olofsson & Knutsson, 2003) for Swedish prepositions. The system can identify insertion, deletion and substitution errors, but does not offer corrections.

A variety of lexical and POS features, including noun and verb phrases in the vicinity of the preposition, as well as their word lemmas and POS tags, are utilized in (Chodorow et al., 2007). The evaluation data consist of newspaper text and a corpus of essays written by 11th and 12th grade students, covering 34 prepositions. A maximum entropy model achieved 69% generation accuracy. Differences in the data set genre, however, prevent a direct comparison with our results.

### 5.2.2 Syntactic Features

To our best knowledge, the only previous work on preposition generation that utilizes syntactic features is (Felice & Pulman, 2007). In addition to a variety of POS features and some WordNet categories, it also considers grammatical relations (e.g., direct or indirect object) extracted from a parser. The grammatical relation feature is identified as a strong feature. A voted perceptron algorithm, trained on five prepositions, yielded 75.6% accuracy on a subset of the British National Corpus.

## 5.3 Features

Despite the variety of features explored in previous work, no analysis on their relative effectiveness has been performed. The main goal of this chapter is to make a direct comparison between lexical and syntactic features. We thus propose two feature sets, LEXICAL and ATTACH. They are restricted to the same types of features except for one difference: the former contains no information on the PP attachment site; the latter does. Some examples of these features are given in Table 5.2.

Table 5.2: Two sets of features are to be contrasted. The Lexical feature set does not specify the PP attachment site; the Attach set does so via the Lexical Head feature. Features extracted from the sentences in Table 5.1 are shown below.

| Sent | Lexical | | | Attach | | |
|---|---|---|---|---|---|---|
| # | VP<br>Head (V) | NP/ADJP<br>Head (N1) | Complement<br>(N2) | Lexical<br>Head (H) | NP/ADJP<br>Head (N1) | Complement<br>(N2) |
| 1 | joined | board | director | joined | board | director |
| 2 | completed | *null* | 1985 | completed | *null* | 1985 |
| 3 | *null* | open | institutions | open | *null* | institutions |
| 4 | announced | rates | 1990 | rates | *null* | 1990 |

### 5.3.1 Lexical Feature Set

Three words in the vicinity of the preposition are extracted[2]:

- **Verb Phrase Head** (V) Head of the verb phrase preceding the preposition.
- **Noun or Adjectival Phrase Head** (N1) Head of the noun phrase or adjectival phrase occurring between V and the preposition.
- **Prepositional Complement** (N2) Head of the noun phrase or nominal *-ing* following the preposition.

For example, for sentence #1 in Table 5.2, V is "*joined*", N1 is "*board*", and N2 is "*director*".

Since the PP may be attached to V or N1, its attachment site cannot be inferred from this feature set. However, either V or N1 can be missing; for example, in sentence #2, N1 is *null* because the verb "*completed*" is immediately followed by the PP "*in 1985*". In such a case, then, there is no PP attachment ambiguity.

### 5.3.2 Attachment Feature Set

In the Lexical feature set, the PP attachment site is left ambiguous. We hypothesize, on linguistic grounds presented in §5.1.1, that it can serve as an informative feature. To test this hypothesis, the Attach feature set re-labels the features in Lexical based on the PP attachment site given by the parse tree:

- **Lexical Head** (H) If the PP is attached to a verb phrase, the lexical head is V; if the PP is attached to a noun- or adjectival phrase, it is N1.
- **Noun or Adjectival Phrase Head** (N1) Similarly, this could be one of two values. If the PP is attached to a noun- or adjectival phrase, this is *null*; if it is attached to a verb phrase, this is the same as the N1 in Lexical. In the latter case, the noun may still play an important role in the choice of preposition. Consider the expressions "*keep the pressure* **on** *someone*" and "*keep pace* **with** *someone*". Under the same

[2] We follow the naming convention in the literature on PP attachment disambiguation, e.g., (Ratnaparkhi et al., 1994). Our Lexical feature set is similar to theirs, with one crucial difference: the preposition *itself* is not included as a feature here, for obvious reasons.

lexical head "*keep*", the N1 nouns "*pressure*" and "*pace*" provide strong clues about the different prepositions.

- **Prepositional Complement** (N2) Same as in the LEXICAL feature set.

## 5.4 Memory-based Learning

The memory-based learning framework has been shown to perform well on a benchmark of language learning tasks (Daelemans et al., 1999). In this framework, feature vectors are extracted from the training set and stored as a database of instances, called the *instance base*. For each test instance, the set of nearest neighbors is retrieved from the instance base. The majority label of this set is returned.

One strength of this approach is that irregular and low-frequency events are preserved in the instance base. This may prove advantageous for our task, as the choice of preposition can be highly context-specific and idiosyncratic.

Of critical importance is the distance metric between two instances, since it determines who the nearest neighbors are. We utilized IB1-IG (Daelemans et al., 1997), an algorithm that uses information gain to define this metric. The following section is a brief summary taken from (Daelemans et al., 1999).

### 5.4.1 IB1-IG

When there are $n$ features, the distance $\Delta$ between two instances $X$ and $Y$ is:

$$\Delta(X,Y) = \sum_{i=1}^{n} w_i \delta(x_i, y_i)$$

where $\delta$, the distance per feature, is defined by:

$$\delta(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases}$$

The weight $w_i$ is intended to reflect the salience of the feature $i$. In IB1-IG, $w_i$ is the information gain (IG) of feature $i$, i.e. the amount of entropy ($H$) reduced by the feature. In order not to favor features with more values, the "split info" ($si(f)$) is used as a normalizing factor. Formally,

$$\begin{aligned} w_i &= \frac{H(C) - \sum_{v \in V_f} P(v) H(C|v)}{si(f)} \\ si(f) &= -\sum_{v \in V_f} P(v) \log_2 P(v) \end{aligned}$$

where $C$ is the set of class labels (i.e., the prepositions), and $V_f$ is the set of values for feature $f$.

Table 5.3: The back-off order of the nearest-neighbor "buckets" in the LEXICAL feature set. The size of each bucket and its corresponding accuracy are listed below for two types of lexical heads: nouns, and verbs with argument PPs.

| Nearest Neighbor Back-off Sequence | Lexical Head | | | |
|---|---|---|---|---|
| | Noun | | Verb (argument PP) | |
| | Size | Accuracy | Size | Accuracy |
| N1+N2+V | 1111 | 78.1% | 395 | 82.3% |
| N1+N2 | 621 | 68.4% | 243 | 24.3% |
| N1+V | 471 | 57.5% | 45 | 51.1% |
| N2+V | 35 | 54.3% | 14 | 78.6% |
| N1 | 14 | 21.4% | 3 | 0% |
| N2 | 0 | n/a | 0 | n/a |
| V | 2 | 100% | 0 | n/a |
| Total | 2254 | 71.8% | 700 | 59.7% |

### 5.4.2 Example

The distance metric could be understood as defining "buckets" of neighbors for each test instance. These buckets, from the nearest ones to the furthest, form the steps of the back-up sequence to be followed by the algorithm, as it searches for the set of nearest neighbors. As an illustration, we now apply the IB1-IG algorithm to the LEXICAL feature set (see §5.3.1).

The information gain of each feature in consideration, V, N1 and N2, is computed on the training set. The information gain for N1 turns out to be the greatest, followed by N2 and then V. By linguistic intuition, N1 and N2 should be most informative for preposition generation when the lexical head is a noun. Since nouns constitute the majority among the lexical heads in our training set (see §5.5), it is natural that N1 and N2 yield the most information gain.

Table 5.3 shows the complete back-off sequence. Given a test instance, its closest neighbors are those training instances that match all three features (N1+N2+V). If such instances exist, the majority label (preposition) of these neighbors is returned. Among our test data whose lexical heads are nouns, 1111 fall into this category, and the predicted preposition is correct 78.1% of the time.

If no training instances match all three features, then the algorithm searches for training instances that match both N1 and N2 (N1+N2), since this combination yields the next largest information gain. The process continues down the back-off sequence in the left column of Table 5.3.

## 5.5 Data

We restrict our attention to the ten most frequently occurring prepositions in the Penn Treebank (Marcus et al., 1994): *of, in, to, for, on, by, at, with, from,* and *as.*

Our test data consists of 3990 occurrences[3] of these ten prepositions in section 23 of the

[3] Some prepositions occur in constructions such as "as ... as", "*because* of" and "*such* as", where their usage is quite predictable. To avoid artificially boosting the generation accuracy, we exclude such cases from our experiments.

Table 5.4: Distribution of lexical heads in our test set, section 23 of the Penn Treebank.

| Lexical Head | Percentage |
|---|---|
| Verb (argument PP) | 17.5% |
| Verb (adjunct PP) | 22.9% |
| Noun | 56.5% |
| Adjective | 3.0% |

Table 5.5: The five most frequently occurring prepositions in the training set, tabulated according to their lexical heads.

| Verbs | | Nouns | | Adjectives | |
|---|---|---|---|---|---|
| Prep. | Frequency | Prep. | Frequency | Prep. | Frequency |
| *in* | 25% | *of* | 55% | *to* | 27% |
| *to* | 16% | *in* | 15% | *of* | 14% |
| *for* | 11% | *for* | 8% | *for* | 14% |
| *on* | 10% | *on* | 5% | *as* | 13% |
| *with* | 10% | *to* | 4% | *with* | 11% |

Penn Treebank. Statistics of the test data are presented in Table 5.4.

Our training data is the AQUAINT Corpus of English News Text, which consists of 10 million sentences drawn from New York Times, Xinhua News Service, and the Associated Press. Parse trees for these sentences are obtained automatically from a state-of-the-art statistical parser (Collins, 2003). The distributions of the prepositions are shown in Table 5.5.

Correctness of the PP attachment in the training data could have been ensured by using a manually parsed corpus, such as the Penn Treebank. However, the parser is reasonably accurate with PP attachments[4], and allows us to take statistical advantage of a much larger training corpus such as AQUAINT. This advantage is especially significant for the memory-based learning framework. Our results may also be more realistic, since treebanks may not be available in other domains.

## 5.6 Evaluation

We conducted experiments to compare the two feature sets described in §5.3: LEXICAL and ATTACH. Results are summarized in Table 6.9.

### 5.6.1 Lexical Feature Set

As discussed in §5.4.2, information gain is greatest for the NP/ADJP Head feature (N1) in the LEXICAL feature set, followed by Prepositional Complement (N2), and lastly Verb Phrase Head (V). This sequence produces the back-off steps of nearest neighbors shown in Table 5.3. Please refer to this table for the rest of this section.

[4] The parser achieves 82.29% recall and 81.51% precision (Collins, 2003) for PP modifications.

Table 5.6: Preposition generation accuracy on the LEXICAL and ATTACH feature sets. The majority baseline is 28.5% (always choosing "*of*"). Results from §5.6.2 are upper bound estimations; results from §5.6.4 is our best without assuming correct attachment information in the test input. For detailed comments, see the individual sections listed in the left column.

| Section | Train Set | Test Set | Verbs (argument PP) | Verbs (adjunct PP) | Nouns | Adjectives | Overall |
|---|---|---|---|---|---|---|---|
| §5.6.1 | LEXICAL | LEXICAL | 59.7% | 58.6% | 71.8% | 75.8% | 66.8% |
| §5.6.4 | ATTACH | LEXICAL | **72.3%** | **60.2%** | **71.7%** | **77.5%** | **69.3%** |
| §5.6.2 | ATTACH | ATTACH | 75.3% | 62.8% | 72.5% | 81.7% | 71.1% |
| §5.6.2 | ATTACH | ARG | 75.3% | 65.9% | n/a | n/a | n/a |

### Nouns and Adjectives

When very similar training instances (N1+N2+V) are available, generation accuracy reaches a relatively high 78.1%. Performance gradually degrades as the nearest neighbors become less similar. The overall accuracy is 71.8% for nouns. The same general trend is observed for adjectives.

### Verbs

Our discussion on verbs will focus on those with argument PPs. Generation accuracy is relatively high (82.3%) when similar neighbors (N1+N2+V) are available. However, at the next back-off level, N1+N2, the accuracy sharply decreases to 24.3%. This drags the overall accuracy down to 59.7%.

The poor performance when backing off to N1+N2 is not accidental. The VP Head (V) feature is most relevant when an argument PP is attached to a verb. Consider the sentence "*They've never shown any inclination to spend money* **on** *production*". Among the N1+N2 neighbors, the preposition "*for*" is the most common, due to expressions such as "*money* **for** *production*". However, the verb "*spend*", coupled with a direct object "*money*", should have signaled a strong preference for the preposition "*on*".

In other words, backing off to V+N2 would have been more appropriate, since the word "*production*" is related more to the verb than to the N1 noun. An obvious remedy is to use a different back-off sequence when the lexical head is a verb. However, there is no way of making this decision, precisely because the PP attachment site is not known. The ATTACH feature set is designed to address this shortcoming.

## 5.6.2 Attachment Feature Set: with Treebank

Without the benefit of attachment information, the LEXICAL feature set is limited to one back-off sequence, ignoring the underlying differences between PPs with verb and noun lexical heads. In contrast, the ATTACH feature set creates an instance base for each kind of lexical head. Each instance base can then optimize its own back-off sequence.

Performance of the ATTACH feature set depends critically on the quality of the PP attachment information. We therefore performed evaluation on the test set under three conditions. In this section, the features were extracted from the manually parsed Penn

Table 5.7: Back-off order of the nearest-neighbor "buckets" for verb lexical heads in the ATTACH feature set. Performance of verbs with argument PPs are listed.

| Nearest Neighbor | Verb (argument PP) | |
|---|---|---|
| Back-off Sequence | Size | Accuracy |
| H+N1+N2 | 389 | 82.3% |
| H+N2 | 143 | 66.4% |
| H | 167 | 67.1% |
| N2 | 1 | 0% |
| Total | 700 | 75.3% |

Treebank; in §5.6.3, they were extracted from automatically produced parse trees; in §5.6.4, no parse tree was assumed to be available.

### Nouns and Adjectives

Information gain is greatest for Lexical Head (H), then Prepositional Complement (N2). Accuracies for both nouns and adjectives (third row in Table 6.9) compare favorably with the LEXICAL set, likely due to the fact that N2 counts are no longer skewed by verb-specific usage.

### Verbs

Information gain is highest for H, followed by N2 and N1, yielding the back-off order shown in Table 5.7. Generation accuracy is 75.3% for verbs with argument PPs, substantially higher than the LEXICAL feature set, at 59.7%.

For those test instances with very similar training counterparts (H+N1+N2), the accuracy is 82.3%. This performance is comparable to the analogous category (N1+N2+V) in the LEXICAL feature set. The gain over the LEXICAL feature set is mainly due to the appropriate back-off to H+N2, which yields 66.4% accuracy. This back-off decision, in contrast to the one with the LEXICAL set, recognizes the importance of the identity of the verb.

Overall, when assuming perfect attachment information, the generation accuracy for the ATTACH feature set is 71.1% (third row in Table 6.9).

### Argument/Adjunct distinction

For verbs[5], further gain in accuracy is still possible if the argument/adjunct distinction is known. Preposition generation tends to be more difficult for verbs with adjunct PPs than those with argument PPs. Since adjuncts depend less on the verb than arguments, their performance naturally suffers at the back-off to H. At this back-off level, arguments achieve 67.1% accuracy (see Table 5.7). The analogous figure for adjuncts is only 31.8%.

One case in point is the sentence "*... other snags that infuriated some fund investors* **in** *October 1987*". As an adjunct, the preposition "*in*" should be highly likely in front of the word "*October*". The back-off to H, however, wrongly predicts "*by*" based on statistics associated with the verb "*infuriated*".

[5] We consider verbs only, since "it is difficult to consistently annotate an argument/adjunct distinction" (Bies et al., 1995) for nouns in the Penn Treebank.

Suppose the argument/adjunct distinction is known in the test data, and that the back-off from H+N2 is changed from H to N2 when the PP is an adjunct. The performance for adjuncts would then rise to 65.9% (last row in Table 6.9), an absolute improvement of 3%.

### 5.6.3 Attachment Feature Set: with automatically derived parse trees

In the previous section, where perfect attachment information is available, the overall generation accuracy reaches 71.1%. This section considers the use of automatically parsed sentences (Collins, 2003) rather than the Penn Treebank. The result should still be interpreted as an upper bound, since the parsing was performed on sentences with the correct prepositions in place.

When the ATTACH features are extracted from these parse trees, the overall generation accuracy decreases to 70.5%. It would be interesting to observe how much further the accuracy would degrade if sentences with preposition errors are fed to the parser. Making a meaningful comparison might prove difficult, however, since one needs to simulate how the test sentences would have been written by non-native speakers of English.

Instead, we now discuss some techniques which, without relying on attachment annotation in input sentences, could still help improve the accuracy.

### 5.6.4 Attachment Feature Set: without parse trees

For texts with lots of grammatical errors, parsing could be challenging, making it difficult to obtain attachment information. Lexical features, however, can be extracted more robustly. Could test instances with only LEXICAL features still leverage an instance base with ATTACH features?

A significant portion of prepositional phrases, in fact, have no ambiguity in their attachment site; for example, when a verb is immediately followed by a preposition, or when an N1 noun occurs at the beginning of the sentence. The unambiguous test instances, then, can take advantage of the ATTACH instance base, while the rest are processed as usual with the LEXICAL instance base. This simple mechanism improves the overall accuracy from 66.8% to 68.7%.

For the ambiguous instances[6], their performance on the LEXICAL instance base still has room for improvement. As we have seen in §5.6.1, the back-off decision is crucial when fully matched instances (N1+N2+V) are not available. Instead of always backing off to N1+N2, entropy statistics can help make more informed choices.

Three sets of nearest neighbors — N1+N2, N1+V and N2+V — are the back-off options. If the lexical head is a verb, for example, one may expect the back-off sets involving V to have relatively low entropy, since the distribution of their prepositions should be more constrained. One reasonable approach is to back-off to the set with the lowest entropy. This procedure raises the overall accuracy to 69.3% (second row in Table 6.9), which is within 2% of the upper bound.

[6] Another potential approach is to first disambiguate the PP attachment site, i.e., determine whether V or N1 is to be assigned as the lexical head H. The instance base with ATTACH features can then be used as before. We have not explored this approach, since literature on PP attachment disambiguation suggests that the preposition identity is one of the most important features (Collins & Brooks, 1995).

## 5.7 Summary

In this chapter, we showed that knowledge of prepositional phrase attachment sites can improve accuracy in preposition generation. In a memory-based learning framework, the improvement is especially substantial when similar training instances are not available and a back-off decision must be made.

For noisy texts, such as input to a grammar checker, PP attachment sites may not be readily available. In these cases, attachment information in training data can still boost generation accuracy to within 2% of the upper bound.

The use of linguistic analysis in the processing of non-native texts raises an important question: how well can such analysis be automatically performed on noisy data, such as texts produced by non-native speakers? **Chapter 6** will address this issue via verb forms, which, when misused, can significantly degrade parser performance.

# Chapter 6

# Robustness in Linguistic Analysis: The Case of Verb Forms

In **Chapter 5**, prepositional phrase attachment sites were shown to improve accuracy in predicting prepositions. In general, for non-native texts, the robustness of automatic syntactic analysis is a critical issue. Mistakes in verb forms, for example, can be misleading to automatic parsers trained on well-formed texts. In this chapter, we will propose a method to improve robustness in the correction of verb forms.

In order to describe the nuances of an action, a verb may be associated with various concepts such as tense, aspect, voice, mood, person and number. In some languages, such as Chinese, the verb itself is not inflected, and these concepts are expressed via other words in the sentence. In highly inflected languages, such as Turkish, many of these concepts are encoded in the inflection of the verb. In between these extremes, English uses a combination of inflections (see Table 6.1) and "helping words", or auxiliaries, to form complex verb phrases.

It should come as no surprise, then, that the misuse of verb forms is a common error category for some non-native speakers of English. For example, in the Japanese Learners of English corpus (Izumi et al., 2003), errors related to verbs are among the most frequent categories. Table 6.2 shows some sentences with these errors.

A system that automatically detects and corrects misused verb forms would be both an educational and practical tool for students of English. It may also potentially improve the performance of machine translation and natural language generation systems, especially

| Form | Example |
|---|---|
| base (bare) | *speak* |
| base (infinitive) | to *speak* |
| third person singular | *speaks* |
| past | *spoke* |
| *-ing* participle | *speaking* |
| *-ed* participle | *spoken* |

Table 6.1: Five forms of inflections of English verbs, illustrated with the verb "*speak*". The base form is also used to construct the infinitive with "*to*". An exception is the verb "*to be*", which has more forms.

| Example | Usage |
|---|---|
| *I take a bath and *reading books.* | FINITE |
| *I can't *skiing well , but ...* | $\text{BASE}_{md}$ |
| *Why did this *happened?* | $\text{BASE}_{do}$ |
| *But I haven't *decide where to go.* | $\text{ED}_{perf}$ |
| *I don't want *have a baby.* | $\text{INF}_{verb}$ |
| *I have to save my money for *ski.* | $\text{ING}_{prep}$ |
| *My son was very *satisfy with ...* | $\text{ED}_{pass}$ |
| *I am always *talk to my father.* | $\text{ING}_{prog}$ |

Table 6.2: Sentences with verb form errors. The intended usages, shown on the right column, are defined in Table 6.3.

when the source and target languages employ very different verb systems.

## 6.1 Introduction

Research on automatic grammar correction has been conducted on a number of different parts-of-speech, such as articles (Knight & Chander, 1994) and prepositions (Chodorow et al., 2007). Errors in verb forms have been covered as part of larger systems such as (Heidorn, 2000), but we believe that their specific research challenges warrant more detailed examination.

We build on the basic approach of template-matching on parse trees in two ways. To improve recall, irregularities in parse trees caused by verb form errors are considered; to improve precision, $n$-gram counts are utilized to filter proposed corrections.

We start with a discussion of the scope of our task in the next section. We then analyze the specific research issues in §6.3 and survey previous work in §6.4. A description of our data follows. Finally, we present experimental results and conclude.

## 6.2 Background

An English verb can be inflected in five forms (see Table 6.1). Our goal is to correct confusions among these five forms, as well as the infinitive. These confusions can be viewed as symptoms of one of two main underlying categories of errors; roughly speaking, one category is semantic in nature, and the other, syntactic.

### 6.2.1 Semantic Errors

The first type of error is concerned with inappropriate choices of tense, aspect, voice, or mood. These may be considered errors in semantics. In the sentence below, the verb "*live*" is expressed in the simple present tense, rather than the perfect progressive:

*He *lives there since June.* (6.1)

Either "*has been living*" or "*had been living*" may be the valid correction, depending on the context. If there is no temporal expression, correction of tense and aspect would be even more challenging.

Similarly, correcting voice and mood often requires real-world knowledge. Suppose one wants to say "*I am prepared for the exam*", but writes "*I am preparing for the exam*". Semantic analysis of the context would be required to correct this kind of error, which will not be tackled in this dissertation[1].

### 6.2.2 Syntactic Errors

The second type of error is the misuse of verb forms. Even if the intended tense, aspect, voice and mood are correct, the verb phrase may still be constructed erroneously. This type of error may be further subdivided as follows:

**Subject-Verb Agreement** The verb is not correctly inflected in number and person with respect to the subject. A common error is the confusion between the base form and the third person singular form, e.g.,

*He *have been living there since June.* (6.2)

**Auxiliary Agreement** In addition to the modal auxiliaries, other auxiliaries must be used when specifying the perfective or progressive aspect, or the passive voice. Their use results in a complex verb phrase, i.e., one that consists of two or more verb constituents. Mistakes arise when the main verb does not "agree" with the auxiliary. In the sentence below, the present perfect progressive tense ("*has been living*") is intended, but the main verb "*live*" is mistakenly left in the base form:

*He has been *live there since June.* (6.3)

In general, the auxiliaries can serve as a hint to the intended verb form, even as the auxiliaries "*has been*" in the above case suggest that the progressive aspect was intended.

**Complementation** A nonfinite clause can serve as complementation to a verb or to a preposition. In the former case, the verb form in the clause is typically an infinitive or an *-ing* participle; in the latter, it is usually an *-ing* participle. Here is an example of a wrong choice of verb form in complementation to a verb:

*He wants *live there.* (6.4)

In this sentence, "*live*", in its base form, should be modified to its infinitive form as a complementation to the verb "*wants*".

This chapter focuses on correcting the above three error types: subject-verb agreement, auxiliary agreement, and complementation. Table 6.3 gives a complete list of verb form usages which will be covered.

## 6.3 Research Issues

One strategy for correcting verb form errors is to identify the intended syntactic relationships between the verb in question and its neighbors. For subject-verb agreement, the

[1] See discussion in §1.2. If the input is "*I am *prepare for the exam*", however, we will attempt to choose between the two possibilities.

| Form | Usage | Description | Example |
|---|---|---|---|
| Base Form as<br>Bare Infinitive | $\text{BASE}_{md}$<br>$\text{BASE}_{do}$ | After modals<br>"Do"-support/-periphrasis; emphatic positive | He **may** *call.* **May** he *call?*<br>He **did** not *call.* **Did** he *call?*<br>I **did** *call.* |
| Base or 3rd person | FINITE | Simple present or past tense | He *calls.* |
| Base Form as<br>*to*-Infinitive | $\text{INF}_{verb}$ | Verb complementation | He **wants** her *to call.* |
| *-ing*<br>participle | $\text{ING}_{prog}$<br>$\text{ING}_{verb}$<br>$\text{ING}_{prep}$ | Progressive aspect<br>Verb complementation<br>Prepositional complementation | He **was** *calling.* **Was** he *calling?*<br>He **hated** *calling.*<br>The device is designed **for** *calling* |
| *-ed*<br>participle | $\text{ED}_{perf}$<br>$\text{ED}_{pass}$ | Perfect aspect<br>Passive voice | He **has** *called.* **Has** he *called?*<br>He **was** *called.* **Was** he *called?* |

Table 6.3: Usage of various verb forms. In the examples, the *italized* verbs are the "targets" for correction. In complementations, the main verbs or prepositions are **bolded**; in all other cases, the auxiliaries are **bolded**.

subject of the verb is obviously crucial (e.g., "*he*" in (6.2)); the auxiliary is relevant for resolving auxiliary agreement (e.g., "*has been*" in (6.3)); determining the verb that receives the complementation is necessary for detecting any complementation errors (e.g., "*wants*" in (6.4)). Once these items are identified, most verb form errors may be corrected in a rather straightforward manner.

The success of this strategy, then, hinges on accurate identification of these items, for example, from parse trees. Ambiguities will need to be resolved, leading to two research issues (§6.3.2 and §6.3.3).

### 6.3.1 Ambiguities

The three so-called *primary verbs*, "*have*", "*do*" and "*be*", can serve as either main or auxiliary verbs. The verb "*be*" can be utilized as a main verb, but also as an auxiliary in the progressive aspect ($\text{ING}_{prog}$ in Table 6.3) or the passive voice ($\text{ED}_{pass}$). The three examples below illustrate these possibilities:

*This* **is** *work not play.* (main verb)

*My father* **is** *working in the lab.* ($\text{ING}_{prog}$)

*A solution* **is** *worked out.* ($\text{ED}_{pass}$)

These different roles clearly affect the forms required for the verbs (if any) that follow. Disambiguation among these roles is usually straightforward because of the different verb forms (e.g., "*working*" vs. "*worked*"). If the verb forms are incorrect, disambiguation is made more difficult:

*This* **is** *work not play.*

*My father* **is** **work in the lab.*

*A solution* **is** **work out.*

Similar ambiguities are introduced by the other primary verbs[2]. The verb "*have*" can function as an auxiliary in the perfect aspect ($\text{ED}_{perf}$) as well as a main verb. The versatile "*do*" can serve as "do"-support or add emphasis ($\text{BASE}_{do}$), or simply act as a main verb.

### 6.3.2 Automatic Parsing

The ambiguities discussed above may be expected to cause degradation in automatic parsing performance. In other words, sentences containing verb form errors are more likely to yield an "incorrect" parse tree, sometimes with significant differences. For example, the sentence "*My father is *work in the laboratory*" is parsed (Collins, 1997) as:

```
(S (NP My father)
 (VP is (NP work))
 (PP in the laboratory))
```

The progressive form "*working*" is substituted with its bare form, which happens to be also a noun. The parser, not unreasonably, identifies "*work*" as a noun. Correcting the *verb* form error in this sentence, then, necessitates considering the *noun* that is apparently a copular complementation.

Anecdotal observations like this suggest that one cannot use parser output naively[3]. We will show that some of the irregularities caused by verb form errors are consistent and can be taken into account.

*One goal of this chapter is to recognize irregularities in parse trees caused by verb form errors, in order to increase recall.*

### 6.3.3 Overgeneralization

One potential consequence of allowing for irregularities in parse tree patterns is overgeneralization. For example, to allow for the "parse error" in §6.3.2 and to retrieve the word "*work*", every determinerless noun would potentially be turned into an *-ing* participle. This would clearly result in many invalid corrections. We propose using $n$-gram counts as a filter to counter this kind of overgeneralization.

*A second goal is to show that n-gram counts can effectively serve as a filter, in order to increase precision.*

## 6.4 Previous Research

This section discusses previous research on processing verb form errors, and contrasts verb form errors with those of the other parts-of-speech.

### 6.4.1 Verb Forms

Detection and correction of grammatical errors, including verb forms, have been explored in various applications. Hand-crafted error production rules (or "mal-rules"), augmenting

[2] The abbreviations *'s* (*is* or *has*) and *'d* (*would* or *had*) compound the ambiguities.

[3] According to a study on parsing ungrammatical sentences (Foster, 2007), subject-verb and determiner-noun agreement errors can lower the F-score of a state-of-the-art probabilistic parser by 1.4%, and context-sensitive spelling errors (not verbs specifically), by 6%.

a context-free grammar, are designed for a writing tutor aimed at deaf students (Michaud et al., 2000). Similar strategies with parse trees are pursued in (Bender et al., 2004), and error templates are utilized in (Heidorn, 2000) for a word processor. Carefully hand-crafted rules, when used alone, tend to yield high precision; they may, however, be less equipped to detect verb form errors within a perfectly grammatical sentence, such as the example given in §6.3.2.

An approach combining a hand-crafted context-free grammar and stochastic probabilities is pursued in **Chapter 8**, but it is designed for a restricted domain only. A maximum entropy model, using lexical and POS features, is trained in (Izumi et al., 2003) to recognize a variety of errors. It achieves 55% precision and 23% recall overall, on evaluation data that partially overlap with ours. Unfortunately, results on verb form errors are not reported separately, and comparison with our approach is therefore impossible.

### 6.4.2 Other Parts-of-speech

Automatic error detection has been performed on other parts-of-speech, e.g., articles (Knight & Chander, 1994) and prepositions (Chodorow et al., 2007). The research issues with these parts-of-speech, however, are quite distinct. Relative to verb forms, errors in these categories do not "disturb" the parse tree as much. The process of feature extraction is thus relatively simple.

## 6.5 Data

### 6.5.1 Development Data

To investigate irregularities in parse tree patterns (see §6.3.2), we utilized the AQUAINT Corpus of English News Text. After parsing the corpus (Collins, 1997), we artificially introduced verb form errors into these sentences, and observed the resulting "disturbances" to the parse trees.

For disambiguation with $n$-grams (see §6.3.3), we made use of the WEB 1T 5-GRAM corpus. Prepared by Google Inc., it contains English $n$-grams, up to 5-grams, with their observed frequency counts from a large number of web pages.

### 6.5.2 Evaluation Data

Two corpora were used for evaluation. They were selected to represent two different genres, and two different mother tongues.

**JLE** (Japanese Learners of English corpus) This corpus is based on interviews for the Standard Speaking Test, an English-language proficiency test conducted in Japan[4] By retaining the verb form errors[5], but correcting all other error types, we generated a test set in which 477 sentences (3.1%) contain subject-verb agreement errors, and 238 (1.5%) contain auxiliary agreement and complementation errors.

[4] See details provided in §3.1.2.

[5] Specifically, those tagged with the "v_fml", "v_fin" (covering auxiliary agreement and complementation) and "v_agr" (subject-verb agreement) types; those with semantic errors (see §6.2.1), i.e. "v_tns" (tense), are excluded.

| Input | Hypothesized Correction | | |
|---|---|---|---|
| | None | Valid | Invalid |
| w/ errors | *false_neg* | *true_pos* | *inv_pos* |
| w/o errors | *true_neg* | *false_pos* | |

Table 6.4: Possible outcomes in our experiment. If no correction is hypothesized for a sentence, the outcome is "false negative" or "true negative", depending on whether the sentence has an error. If a valid correction is hypothesized, the outcome is, similarly, either "true positive" or "false positive". A fifth category ("inv_pos") describes cases where a correction is warranted but the hypothesized correction is invalid.

**HKUST** This corpus of short essays, containing a total of 2556 sentences, was collected from students, all native Chinese speakers, at the Hong Kong University of Science and Technology[6].

### 6.5.3 Evaluation Metric

For each verb in the input sentence, a change in verb form may be hypothesized. There are five possible outcomes for this hypothesis, as enumerated in Table 6.4. To penalize "false alarms", a strict definition is used for false positives — even when the hypothesized correction yields a good sentence, it is still considered a false positive so long as the original sentence is acceptable.

It can sometimes be difficult to determine which words should be considered verbs, as they are not clearly demarcated in our evaluation corpora. We will thus apply the outcomes in Table 6.4 at the sentence level; that is, the output sentence is considered a true positive only if the original sentence contains errors, and only if valid corrections are offered for *all* errors.

The following statistics are computed:

**Accuracy** The proportion of sentences which, after being treated by the system, have correct verb forms. That is, (*true_neg* + *true_pos*) divided by the total number of sentences.

**Recall** Out of all sentences with verb form errors, the percentage whose errors have been successfully corrected by the system. That is, *true_pos* divided by (*true_pos* + *false_neg* + *inv_pos*).

**Detection Precision** This is the first of two types of precision to be reported, and is defined as follows: Out of all sentences for which the system has hypothesized corrections, the percentage that actually contain errors, without regard to the validity of the corrections. That is, (*true_pos* + *inv_pos*) divided by (*true_pos* + *inv_pos* + *false_pos*).

**Correction Precision** This is the more stringent type of precision. In addition to successfully determining that a correction is needed, the system must offer a valid correction. Formally, it is *true_pos* divided by (*true_pos* + *false_pos* + *inv_pos*).

[6] See details provided in §3.1.2.

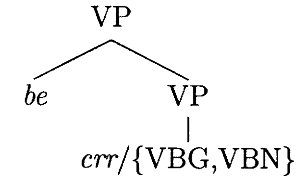
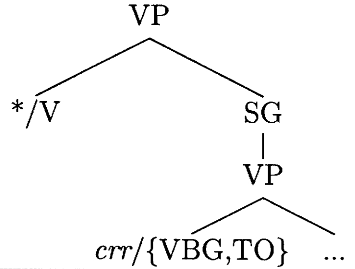
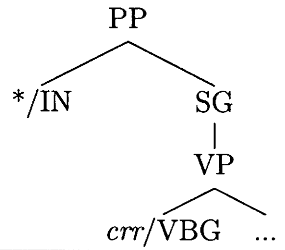
| Expected Tree $\{\langle usage \rangle, ...\}$ |
|---|
| $\{\text{ING}_{prog}, \text{ED}_{pass}\}$ |
|  |
| $\{\text{ING}_{verb}, \text{INF}_{verb}\}$ |
|  |
| $\text{ING}_{prep}$ |
|  |

Table 6.5: Effects of incorrect verb forms on parse trees. This table shows trees normally expected for the indicated usages (see Table 6.3). Table 6.6 shows the resulting trees when the correct verb form ⟨*crr*⟩ is replaced by ⟨*err*⟩. Detailed comments are provided in §6.6.1.

### 6.5.4 Evaluation Procedure

For the JLE corpus, all figures above will be reported. The HKUST corpus, however, will not be evaluated on subject-verb agreement, since a sizable number of these errors are induced by other changes in the sentence. For example, whenever the subject of the verb needs to be changed from singular to plural, the verb conjugation must also be altered correspondingly; however, these changes do not constitute a subject-verb agreement error.

Furthermore, the HKUST corpus will require manual evaluation, since the corrections are not annotated. Two native speakers of English were given the edited sentences, as well as the original input. For each pair, they were asked to select one of four statements: one of the two is better, or both are equally correct, or both are equally incorrect. The correction precision is thus the proportion of pairs where the edited sentence is deemed better. Accuracy and recall cannot be computed, since it was impossible to distinguish syntactic errors from semantic ones (see §6.2).

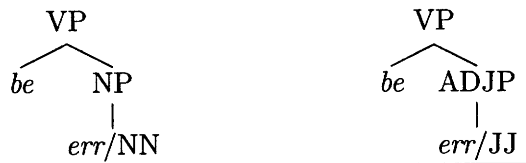
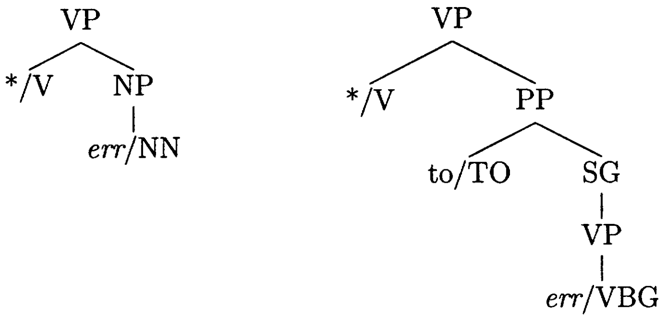
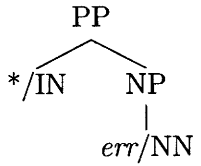
| Tree disturbed by substitution [⟨*crr*⟩ → ⟨*err*⟩] |
|---|
| *A dog is [sleeping→sleep]. I'm [living→live] in XXX city.* |
|  |
| *I like [skiing→ski] very much. She likes to [go→going] around.* |
|  VP<br>*/V<br>NP<br>err/NN<br>VP<br>*/V<br>PP<br>to/TO<br>SG<br>VP<br>err/VBG |
| *I lived in France for [studying→study] French language.* |
|  |

Table 6.6: Effects of incorrect verb forms on parse trees. Table 6.5 shows trees normally expected for the indicated usages (see Table 6.3). This table shows the resulting trees when the correct verb form ⟨*crr*⟩ is replaced by ⟨*err*⟩. Detailed comments are provided in §6.6.1.

### 6.5.5 Baselines

Since the vast majority of verbs are in their correct forms, the *majority baseline* is to propose no correction. Although trivial, it is a surprisingly strong baseline, achieving more than 98% for auxiliary agreement and complementation in JLE, and just shy of 97% for subject-verb agreement.

For auxiliary agreement and complementation, the *verb-only baseline* is also reported. It attempts corrections only when the word in question is actually tagged as a verb. That is, it ignores the spurious noun- and adjectival phrases in the parse tree discussed in §6.3.2, and relies only on the output of the part-of-speech tagger.

## 6.6 Experiments

Corresponding to the issues discussed in §6.3.2 and §6.3.3, our experiment consists of two main steps.

### 6.6.1 Derivation of Tree Patterns

Based on (Quirk et al., 1985), we observed tree patterns for a set of verb form usages, as summarized in Table 6.3. Using these patterns, we introduced verb form errors into AQUAINT, then re-parsed the corpus (Collins, 1997), and compiled the changes in the "disturbed" trees into a catalog.

A portion of this catalog[7] is shown in Table 6.5. Comments on {ING$_{prog}$,ED$_{pass}$} can be found in §6.3.2. Two cases are shown for {ING$_{verb}$,INF$_{verb}$}. In the first case, an *-ing* participle in verb complementation is reduced to its base form, resulting in a noun phrase. In the second, an infinitive is constructed with the *-ing* participle rather than the base form, causing "*to*" to be misconstrued as a preposition. Finally, in ING$_{prep}$, an *-ing* participle in preposition complementation is reduced to its base form, and is subsumed in a noun phrase.

### 6.6.2 Disambiguation with N-grams

The tree patterns derived from the previous step may be considered as the "necessary" conditions for proposing a change in verb forms. They are not "sufficient", however, since they tend to be overly general. Indiscriminate application of these patterns on AQUAINT would result in false positives for 46.4% of the sentences.

For those categories with a high rate of false positives (all except BASE$_{md}$, BASE$_{do}$ and FINITE), we utilized $n$-grams as filters, allowing a correction only when its $n$-gram count in the WEB 1T 5-GRAM corpus is greater than that of the original. The filtering step reduced false positives from 46.4% to less than 1%. Table 6.7 shows the $n$-grams, and Table 6.8 provides a breakdown of false positives in AQUAINT after $n$-gram filtering.

### 6.6.3 Results for Subject-Verb Agreement

In JLE, the accuracy of subject-verb agreement error correction is 98.93%. Compared to the majority baseline of 96.95%, the improvement is statistically significant[8]. Recall is 80.92%; detection precision is 83.93%, and correction precision is 81.61%.

[7] Only those trees with significant changes above the leaf level are shown.

[8] $p < 0.005$ according to McNemar's test.

| $N$-gram | Example |
|---|---|
| be {ING$_{prog}$,<br>ED$_{pass}$} * | The dog is *sleeping.*<br>The door is *open.* |
| verb {ING$_{verb}$,<br>INF$_{verb}$} * | I need *to do* this.<br>I need *beef* for the curry. |
| *verb*$_1$ *ing<br>and {ING$_{verb}$,<br>INF$_{verb}$} | *enjoy* reading and<br>*going* to pachinko<br>*go* shopping and *have* dinner |
| prep<br>{ING$_{prep}$} * | for *studying* French language<br>a class for *sign* language |
| have<br>{ED$_{perf}$} * | I have *rented* a video<br>I have *lunch* in Ginza |

Table 6.7: The $n$-grams used for filtering, with examples of sentences which they are intended to differentiate. The hypothesized usages (shown in the curly brackets) as well as the original verb form, are considered. For example, the first sentence is originally "*The dog is *sleep.*" The three trigrams "*is sleeping .*", "*is slept .*" and "*is sleep .*" are compared; the first trigram has the highest count, and the correction "*sleeping*" is therefore applied.

| Hyp.<br>Usage | False<br>Pos. | Hypothesized<br>Usage | False<br>Pos. |
|---|---|---|---|
| BASE$_{md}$ | 16.2% | {ING$_{verb}$,INF$_{verb}$} | 33.9% |
| BASE$_{do}$ | 0.9% | {ING$_{prog}$,ED$_{pass}$} | 21.0% |
| FINITE | 12.8% | ING$_{prep}$ | 13.7% |
|  |  | ED$_{perf}$ | 1.4% |

Table 6.8: The distribution of false positives in AQUAINT. The total number of false positives is 994, representing less than 1% of the 100,000 sentences drawn from the corpus.

| Corpus | Method | Accuracy | Precision (correction) | Precision (detection) | Recall |
|---|---|---|---|---|---|
| JLE | verb-only | 98.85% | 71.43% | 84.75% | 31.51% |
| | all | 98.94% | 68.00% | 80.67% | 42.86% |
| HKUST | all | not available | 71.71% | not available | |

Table 6.9: Results on the JLE and HKUST corpora for auxiliary agreement and complementation. The majority baseline accuracy is 98.47% for JLE. The verb-only baseline accuracy is 98.85%, as indicated on the second row. "All" denotes the complete proposed method. See §6.6.4 for detailed comments.

| Usage | JLE | HKUST |
|---|---|---|
| | Count (Prec.) | Count (Prec.) |
| BASE$_{md}$ | 13 (92.3%) | 25 (80.0%) |
| BASE$_{do}$ | 5 (100%) | 0 |
| FINITE | 9 (55.6%) | 0 |
| ED$_{perf}$ | 11 (90.9%) | 3 (66.7%) |
| {ING$_{prog}$,ED$_{pass}$} | 54 (58.6%) | 30 (70.0%) |
| {ING$_{verb}$,INF$_{verb}$} | 45 (60.0%) | 16 (59.4%) |
| ING$_{prep}$ | 10 (60.0%) | 2 (100%) |

Table 6.10: Correction precision of individual correction patterns (see Table 6.5) on the JLE and HKUST corpus.

Most mistakes are caused by misidentified subjects. Some *wh*-questions prove to be especially difficult, perhaps due to their relative infrequency in newswire texts, on which the parser is trained. One example is the question "*How much extra time does the local train *takes?*". The word "*does*" is not recognized as a "do"-support, and so the verb "*take*" was mistakenly turned into a third person form to agree with "*train*".

### 6.6.4 Results for Auxiliary Agreement & Complementation

Table 6.9 summarizes the results for auxiliary agreement and complementation, and Table 6.2 shows some examples of real sentences corrected by the system. Our proposed method yields 98.94% accuracy. It is a statistically significant improvement over the majority baseline (98.47%), although not significant over the verb-only baseline[9] (98.85%), perhaps a reflection of the small number of test sentences with verb form errors. The Kappa statistic for the manual evaluation of HKUST is 0.76, corresponding to "substantial agreement" between the two evaluators (Landis & Koch, 1977). The correction precisions for the JLE and HKUST corpora are comparable.

Our analysis will focus on {ING$_{prog}$,ED$_{pass}$} and {ING$_{verb}$,INF$_{verb}$}, two categories with relatively numerous correction attempts and low precisions, as shown in Table 6.10. For {ING$_{prog}$,ED$_{pass}$}, many invalid corrections are due to wrong predictions of voice, which involve semantic choices (see §6.2.1). For example, the sentence "... *the main duty is study well*" is edited to "... *the main duty is studied well*", a grammatical sentence but

[9] With $p = 1 * 10^{-10}$ and $p = 0.038$, respectively, according to McNemar's test

semantically unlikely.

For {$\text{ING}_{verb}$,$\text{INF}_{verb}$}, a substantial portion of the false positives are valid, but unnecessary, corrections. For example, there is no need to turn "*I like cooking*" into "*I like to cook*", as the original is perfectly acceptable. Some kind of confidence measure on the $n$-gram counts might be appropriate for reducing such false alarms.

Characteristics of speech transcripts pose some further problems. First, colloquial expressions, such as the word "*like*", can be tricky to process. In the question "*Can you like give me the money back*", "*like*" is misconstrued to be the main verb, and "*give*" is turned into an infinitive, resulting in "*Can you like *to give me the money back*". Second, there are quite a few incomplete sentences that lack subjects for the verbs. No correction is attempted on them.

Also left uncorrected are misused forms in non-finite clauses that describe a noun. These are typically base forms that should be replaced with *-ing* participles, as in "*The girl *wear a purple skiwear is a student of this ski school*". Efforts to detect this kind of error resulted in a large number of false alarms.

Recall is further affected by cases where a verb is separated from its auxiliary or main verb by many words, often with conjunctions and other verbs in between. One example is the sentence "*I used to climb up the orange trees and *catching insects*". The word "*catching*" should be an infinitive complementing "*used*", but is placed within a noun phrase together with "*trees*" and "*insects*".

## 6.7 Summary

In this chapter, we have presented a method for correcting verb form errors. We investigated the ways in which verb form errors affect parse trees. When allowed for, these unusual tree patterns can expand correction coverage, but also tend to result in overgeneration of hypothesized corrections. $N$-grams have been shown to be a simple yet effective filter for this problem.

So far, our approach for correcting grammatical errors has been "one-size-fits-all"; the same algorithm is used regardless of the characteristics of the input texts. While this approach is reasonable for linguistic classes with clearly defined rules, such as verb forms, it may be inadequate for other classes that exhibit more ambiguity, and where the rules are not as clearly defined. **Chapter 7** will focus on such a class — English articles.

# Chapter 7

# Personalization: The Case of Articles

So far, no consideration has been given to the characteristics of the input text. Whether the text is almost perfect or ridden with errors, the same post-editing algorithm is used. In this chapter, we investigate one dimension of *personalization* — the effects of adjusting the model based on the quality of the input text.

In particular, we will study these effects on the correction of article usage. An English noun phrase (NP) may contain a determiner, such as *this*, *that*, *a*, *an* or *the*, which specifies the reference of its head. The two most common of these determiners, *a/an* and *the*, are also known as articles. Broadly speaking, *the* indicates that the head refers to someone or something that is uniquely defined; *a/an*[1], or the absence of any articles, indicates that it is a general concept. Many languages do not have any articles; native speakers of these languages often have difficulty choosing appropriate English articles, and tend to underuse them.

Although the ultimate goal is to automatically correct article usage in English sentences written by non-native speakers, this chapter will focus on a more specific task: restoring missing articles. Our strategy is to create a series of training sets, in which more articles are progressively removed. These training sets thus yield models that insert articles with varying degrees of aggressiveness, in order to suit input texts of different qualities.

## 7.1 Article Generation

As the first step, we will establish a feature set that yields competitive results in the article generation task. As in the previous chapters, we will tackle article generation[2] as a classification task — for each base NP, predict its article, or lack of it. Then, in §7.2, we will extend this feature set for the article restoration problem, where the model will be adjusted with respect to the initial quality of the input text.

[1] The distinction between "*a*" and "*an*" can be easily resolved and is not considered further. Both will henceforth be represented as "*a*".

[2] See §2.4.2 for a literature review in this area.

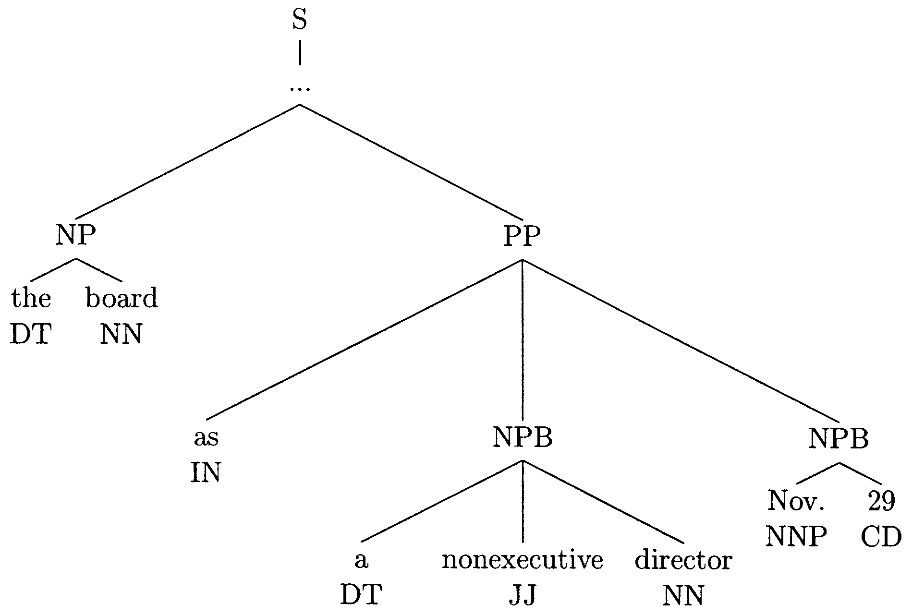


Figure 7-1: A portion of the parse tree of the sentence, *Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.*

### 7.1.1 Features

Features are drawn from two sources: the parse tree of the sentence, and WordNet Version 2.0. Fifteen categories of syntactic and semantic features are extracted from each base NP. Take the sentence "*Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29*" as an example. From its parse tree, part of which is shown in Figure 7-1, the following features are extracted for the base NP "*a nonexecutive director*":

**Head** (*director*) The root form of the head of the NP. A number is rewritten as <*number*>. The head is determined using the rules in (Collins, 1999), except for possessive NPs. The head of a possessive NP is *'s*, which is not indicative of its article preference. Instead, we use the second best candidate for NP head.

**Number** (*singular*) If the POS tag of the NP head is *NN* or *NNP*, the number of the head is *singular*; if the tag is *NNS* or *NNPS*, it is *plural*; for all other tags, it is *n/a*.

**Head POS** (*NN*) The POS tag of the NP head. Any information about the head's number is hidden; *NNS* is re-written as *NN*, and *NNPS* as *NNP*.

**Parent** (*PP*) The category of the parent node of the NP.

**Non-article determiner** (*null*) A determiner other than *a* or *the* in the NP.

**Words before head** (*nonexecutive*) Words inside the NP that precede the head, excluding determiners.

**Words after head** (*null*) Words inside the NP that follow the head, excluding determiners.

**POS of words before head** (*JJ*) The POS tags of words inside the NP that precede the head, excluding determiners.

**POS of words after head** (*null*) The POS tags of words inside the NP that follow the head, excluding determiners.

**Words before NP** (*board, as*) The two words preceding the base NP. This feature may be *null.*

**Words after NP** (*Nov, <number>*) The two words following the base NP. This feature may be *null.*

**Hypernyms** ({*entity*}, {*object, physical object*}, ..., {*head, chief, top dog*}, {*administrator, decision maker*}) Each synset in the hierarchy of hypernyms for the head in WordNet is considered a feature. We do not attempt any sense disambiguation, but always use the hypernyms for the first sense.

**Referent** (*no*) If the same NP head appears in one of the 5 previous sentences, then *yes*; otherwise, *no*.

Finally, the label is the article in the original sentence. For this NP, the label is "*a*"; in other NPs the label could be "*the*" or "*null*".

### 7.1.2 Data

To facilitate comparison with previous work reported in (Knight & Chander, 1994; Minnen et al., 2000), we used the Penn Treebank-3 as evaluation data. We harvested all base NPs and their features from the text of treebank. Rather than reading the NPs off the treebank, however, we extracted them from trees produced by a natural language parser (Collins, 1999)[3]. This procedure reflects better the actual practice in language learning systems, since no gold-standard parse trees would be available in any realistic scenario. Sections 00 to 21 are used as training data[4]. There are about 260,000 base NPs. The distribution of the articles in this set is roughly 70.5% *null*, 20% *the* and 9.5% *a*. Three training sets, with increasingly richer feature sets, were created:

- TRAINGEN$_{base}$: This set uses only three features, `Head`, `Number` and `Head POS`.
- TRAINGEN$_{Minnen}$: This set uses the subset of our features that were also used in (Minnen et al., 2000). In addition to all the features in TRAINGEN$_{base}$, these include `Parent` and `Non-article determiner`.
- TRAINGEN: This set uses our full set of features.

During testing, we would like to measure the degree to which missing articles may corrupt the parser output. Ideally, the parse tree of an input sentence with inappropriate articles should be identical[5] to that of the equivalent correct sentence. However, a natural

[3] Using Ratnaparkhi's MXPOST to produce part-of-speech tags (Ratnaparkhi, 1996).

[4] Since Collins' parser was trained on these sections 02 to 21, we recognize that the parsing accuracy is higher than what would be expected from other texts. However, our results can still be compared with (Knight & Chander, 1994; Minnen et al., 2000), since the context features are read directly from the treebank in both works.

[5] Except, of course, the leaves for the articles.

| Feature Set:↓ Test Set:→ | DROP0 |
|---|---|
| TRAINGEN | 87.7% |
| TRAINGEN$_{Minnen}$ | 82.4% |
| TRAINGEN$_{base}$ | 80.1% |

Table 7.1: Accuracy rates in article generation. The feature sets and test set are described in §7.1.2.

language parser, trained on grammatical sentences, does not perform as well on sentences with inappropriate articles. Not all NPs might be accurately identified, and the context features of the NPs might be distorted.

To this effect, we generated four test sets from the text in section 23 of the Penn Treebank-3 by first dropping 70%, 30% and 0% of the articles, then re-parsing the resulting sentences. We call these sets DROP70, DROP30 and DROP0. There are about 1300 *a*'s and 2800 *the*'s in the section.

Among sentences in DROP30, 97.6% had all their NP heads correctly extracted. Within this subset, 98.7% of the NPs had correct boundaries. The accuracy rate for NP heads decreased to 94.7% for DROP70. Among the sentences in DROP70 with correct heads, 97.5% of the NPs had correct boundaries. Parser robustness will be discussed as a future research direction in §10.3.3.

### 7.1.3 Results

We trained maximum-entropy classifiers on the three feature sets TRAINGEN, TRAINGEN$_{Minnen}$ and TRAINGEN$_{base}$. Contextual predicates that were true in less than 5 base NPs in the training sets were deemed unreliable and rejected. The weight for each surviving predicate was initialized to zero, and then trained by iterative scaling. The three classifiers were applied on DROP0 to predict an article for each NP. Table 7.1 shows the accuracy rates.

Our baseline accuracy rate on DROP0, 80.1%, is close to the corresponding rate reported in (Minnen et al., 2000)[6]. Our full feature set yielded the best result, 87.7%, an improvement over both (Minnen et al., 2000) and (Knight & Chander, 1994). The confusion matrix is shown in Table 7.2.

We added 8 more features[7] to TRAINGEN$_{Minnen}$ to make up TRAINGEN. After adding the features `Words before/after head` and `POS of words before/after head`, the accuracy increased by more than 4%. In fact, features with the heaviest weights were dominated by these feature types; they were not used in (Minnen et al., 2000).

The `Words before/after NP` features gave another 0.8% boost to the accuracy. These features were also used in (Knight & Chander, 1994) but not in (Minnen et al., 2000). The `Hypernyms` feature, which placed NP heads under the WordNet semantic hierarchy, was intended to give a smoothing effect. It further raised the accuracy by 0.3%.

At this point, the biggest source of error was generating *null* instead of the correct *the*. We introduced the `Referent` feature to attack this problem. It had, however, only a modest effect. The 656 misclassifications seemed rather heterogeneous. There was an almost even split between singular and plural NP heads; more than three quarters of these

[6] 80.8% for the "head+its part-of-speech" feature.

[7] see list in §7.1.1.

| Gold:↓ Predicted:→ | *null* | *the* | *a* |
|---|---|---|---|
| *null* | 9647 | 324 | 124 |
| *the* | 656 | 1898 | 228 |
| *a* | 167 | 249 | 878 |

Table 7.2: Confusion matrix for article generation using TRAINGEN on DROP0

heads appeared in the list three times or less. The most frequent ones were <*number*> (22 times), *bond*, *year*, *security*, *court* (8 times), *fifth* and *show* (7 times).

## 7.2 Article Restoration

So far, articles in the original sentences are simply ignored; however, they are clearly not randomly chosen, and may provide useful hints. To this end, we augment the feature vector in TRAINGEN with a new feature, `Article`, which is the article in the original sentence. Its weight is intended to reflect the likelihood of transformations such as [*a*→*null*]. If *a priori* information about the quality of the text, or an estimation[8], is available, performance may be improved by adjusting this weight. For example, if the text is believed to be of high quality, one should be conservative in editing its articles.

In general, "article quality" can be characterized as a 3 × 3 confusion matrix. The articles on the rows are the correct ones; those on the columns are the ones actually used in the sentence. For example, if a sentence has the matrix

$$\begin{array}{cccc} & a & null & the \\ a & 0.1 & 0.9 & 0 \\ null & 0 & 1 & 0 \\ the & 0 & 0.6 & 0.4 \end{array}$$

then the article *the* is correctly used in the sentence with a 40% chance, but is mistakenly dropped (i.e., substituted with *null*) with a 60% chance. If one could accurately estimate the underlying confusion matrix of a sentence, then one could judiciously use the existing articles as a factor when generating articles.

For the article restoration task, we assume that articles may be dropped, but no unnecessary articles are inserted, and the articles *the* and *a* are not confused with each other. In other words, the four zero entries in the matrix above are fixed.

### 7.2.1 Data

To simulate varying qualities of input sentences, we perturbed the `Article` feature with two different confusion matrices, resulting in the following training sets:

- TRAINDROP70: The `Article` feature is perturbed according to the confusion matrix

$$\begin{pmatrix} 0.3 & 0.7 & 0 \\ 0 & 1 & 0 \\ 0 & 0.7 & 0.3 \end{pmatrix}$$

[8] E.g., via observations of writing samples from a specific L1 community, as described in §3.2.

| Label | Predicate |
|---|---|
| *a* | (Word before head = lot) |
| *the* | (Head = Netherlands) |
| *the* | (Head = Beebes) |
| *a* | (Word before head = million) |
| *a* | (Hypernym = {struggle, battle}) |

Table 7.3: The five features associated with the heaviest weights after training on TrainDrop30 for 1500 rounds. Notice that two features, Head and Word before head, dominate the top weights.

That is, 70% of the feature (Article = the), and 70% of the feature (Article = a), are replaced with the feature (Article = null). The rest are unchanged. This set trains the model to aim to insert enough articles such that the initial number of articles in a sentence would constitute about 30% of the final number of articles.

- TrainDrop30: The Article feature is perturbed according to the confusion matrix

$$\begin{pmatrix} 0.7 & 0.3 & 0 \\ 0 & 1 & 0 \\ 0 & 0.3 & 0.7 \end{pmatrix}$$

That is, 30% of (Article = the) and 30% of (Article = a) are replaced with (Article = null). Upon seeing a *null* in an input sentence, all else being equal, TrainDrop30 should be less predisposed than TrainDrop70 to change it to *the* or *a*. The features associated with the heaviest weights[9] are tabulated in Table 7.3.

### 7.2.2 Experiments

We trained maximum-entropy classifiers with TrainDrop30 and TrainDrop70, and used them to predict the articles in the test sentences[10]. While the set-up of the task is the same as in §7.1, the classifiers are significantly different. In contrast to TrainGen, which altogether ignores any articles in the input text, TrainDrop30 and TrainDrop70 take into account the likelihood of inserting new articles, and of deleting or substituting existing articles, via the Article feature. Furthermore, differences in the training data are expected to cause TrainDrop70 to be more aggressive with insertion than TrainDrop30.

Rather than article generation accuracy, deletion/insertion/substitution rates are used as the evaluation metric, since in general it is difficult to determine the number of *null* articles in a text. The results are listed in Table 7.4, and two observations can be made. First, across all test sets, performance significantly improved when articles in the input were taken into account (TrainDrop30 and TrainDrop70). Second, the results reflect the intuition that, for a test set where $n$% of the articles have been dropped, the optimal model

[9] The feature with the heaviest weight, (Head = the) with label "*the*", is omitted because it is purely due to parsing errors in a handful of cases. The article *the* as head of an NP is due to incorrect parses. An example is the sentence *Mr. Nixon, the most prominent American to come to China since ....* The parse had an *S* parent dominating a base NP, which contained *the* alone, and an adjective phrase, which contained *most prominent American* and so forth.

[10] If an NP contained an article, the predicted article replaces it; otherwise, the predicted article is inserted at the beginning of the NP.

| Training Set:↓ Test Set:→ | Drop0 | Drop30 | Drop70 |
|---|---|---|---|
| TrainDrop30 | 4.4% | **20.5%** | 40.7% |
| TrainDrop70 | 8.9% | 22.3% | **38.5%** |
| TrainGen | 43.0% | 46.0% | 49.4% |

Table 7.4: Article error rate, i.e., the total number of errors divided by the number of articles in the gold-standard sentences. Errors include deletions, insertions, and substitutions of articles.

| Training Set:↓ Test Set:→ | Drop0 | Drop30 | Drop70 |
|---|---|---|---|
| TrainDrop30 | 0.4% | 13.0% | 28.4% |
| TrainDrop70 | 0.3% | 9.7% | 20.2% |
| TrainGen | 19.3% | 21.7% | 23.9% |

Table 7.5: Deletion error rate, i.e., the number of deletion errors divided by the number of articles in the gold-standardr sentences. The text in Drop30 has a deletion rate of 30%; a 15% rate in the output would mean that the system successfully restores half of the missing articles.

is the one that has been trained on sentences with $n$% of the articles missing. More generally, one could expect that the optimal training set is the one whose underlying confusion matrix is the most similar to that of the test set.

The breakdown of the article error rate into deletion and insertion errors are shown in Tables 7.5 and 7.6 (substitution errors not shown). The trends in Table 7.5 are quite straightforward: the deletion error rate was lower when the model inserted more articles, and when fewer articles were dropped in the original sentences. For example, starting with a deletion rate of 70% (for test set Drop70), TrainDrop70 reduced it to about 20%, meaning it successfully restored 2 out of every 7 missing articles.

As one becomes more aggressive in inserting articles, the decreasing deletion rate is counter-balanced by the increasing insertion rate, shown in Table 7.6. It is interesting to note that for both TrainDrop30 and TrainDrop70, the insertion error rate rose as more articles were dropped in the test set. It turned out that, in many cases, inaccurate parsing (see §7.1.2) led to incorrect NP boundaries, and hence incorrect insertion points for articles.

Substitution errors (not shown) constitute the smallest category of the three. Most of them were caused by the following: an article (e.g., *a*) was replaced by *null* in the test set; then, the wrong article (e.g., *the*) was generated to replace the *null*. In general, the substitution rate was higher when the model inserted more articles, and when more articles were dropped in the original sentences.

| Training Set:↓ Test Set:→ | Drop0 | Drop30 | Drop70 |
|---|---|---|---|
| TrainDrop30 | 4.0% | 4.9% | 5.9% |
| TrainDrop70 | 8.6% | 9.7% | 11.2% |
| TrainGen | 11.9% | 13.0% | 14.6% |

Table 7.6: Insertion error rate, i.e., the number of insertion errors divided by the number of articles in the gold-standard sentences.

| Training Set:↓ Test Set:→ | DROP0 | DROP30 | DROP70 |
|---|---|---|---|
| TRAINDROP30 | +3.9% | **+24.4%** | +60.1% |
| TRAINDROP70 | +8.1% | +38.1% | **+66.0%** |
| TRAINGEN | -7.5% | +23.8% | +65.9% |

Table 7.7: Change in the number of articles

### 7.2.3 Choosing the Best Model

As Table 7.4 suggests, it is important to choose the optimal training set with respect to the quality of the input sentences. When no *a priori* information about this quality is available, how can one determine the optimal model?

Table 7.7 shows the changes in the number of articles in the system output. When running TRAINGEN on DROP30 and DROP70, there was an increase of 23.8% and 65.9% in the number of articles. These rates of increase were close to those obtained (24.4% and 66.0%) when running their respective optimal sets, TRAINDROP30 and TRAINDROP70. It appeared that TRAINGEN was able to provide a reasonable estimate of the number of articles that should be restored. When given new input sentences, one could use TRAINGEN to estimate the percentage of missing articles, then choose the most appropriate training set accordingly.

## 7.3 Summary

A maximum-entropy classifier was applied on the article generation task, using features drawn from a statistical natural language parser and WordNet. This feature set yielded state-of-the-art results.

The same model was applied on the article restoration task, where input sentences may have missing articles. In a departure from the previous chapters, the article in the input text is taken into account; that is, the difference between the target (or predicted) article and the source article (e.g., [*a*→*null*]) is explicitly modelled. An initial estimation of the quality of the input text helps choose the most appropriate insertion model to use. Our best results are 20.5% article error rate for sentences where 30% of the articles have been dropped, and 38.5% for those where 70% of the articles have been dropped.

From **Chapter 5** till now, the error correction task is reduced to a classification task, by limiting our focus on one part-of-speech at a time. When the input sentence is of lower quality, multiple mistakes, possibly inter-dependent, may need to be considered and corrected simultaneously. For example, if a noun has the wrong number, then it is difficult to independently propose an appropriate article for that noun. In the next chapter, we will view the correction task as a sentence re-generation problem (see §2.4.1), which is more suitable for this type of sentence.

# Chapter 8

# Sentence Re-generation

The previous chapters have treated each class of non-native grammatical error — prepositions, verbs, and articles, respectively — in isolation as a classification problem. This treatment sidesteps two potential issues. First, if an error interacts with another word (e.g., noun number and verb conjugation), the correction of either error cannot be determined independently. Second, if there are errors in other parts of the sentence, then the extracted features may also contain some of these errors, making the classification results less reliable.

If the input sentence is of relatively high quality, these two potential problems may not be severe. If it is ridden with errors, however, then an approach that incorporates global considerations would be more appropriate. In this chapter, we will adopt such an approach, and propose a sentence re-generation method that belongs to the paradigm described in §2.4.1.

In the past few years, the Spoken Language Systems group has been developing a conversational language learning system (Seneff et al., 2004), which engages students in a dialogue in order to help them learn a foreign language. An important component of such a system is to provide corrections of the students' mistakes, both phonetic (Dong et al., 2004) and grammatical, the latter of which is our focus. For example, the student might say, "** What of time it arrive into Dallas?*" The system would be expected to correct this to, "*What time will it arrive in Dallas?*" Given that the system's natural language understanding (NLU) component uses a probabilistic context-free grammar, a parsing-based approach might appear to be a natural solution.

However, as discussed in the literature review on parsing-based approaches (§2.3), the addition of mal-rules makes context-free grammars increasingly complicated, exponentially growing the number of ambiguous parses. We instead propose a two-step, *generation-based* framework. Given a possibly ill-formed input, the first step paraphrases the input into a word lattice, licensing all conceivable corrections; and the second step utilizes language models and parsing to select the best rephrasing.

The rest of the chapter is organized as follows. §8.1 identifies the types of errors we intend to handle and describes our two-step, generation-based framework for grammar correction. §8.3 presents some experiments on a corpus of flight queries. §8.4 describes an application of our approach on postediting MT output.

| Part-of-speech | Words |
|---|---|
| Articles | *a, an, the* |
| Modals,<br>Verb auxiliary | *can, could, will, would, must, might, should*<br>*be, have, do* |
| Prepositions | *about, at, by, for, from, in, of, on, with, to* |
| Nouns | *flight, city, airline, friday, departure, ...* |
| Verbs | *like, want, go, leave, take, book, ...* |

Table 8.1: The parts-of-speech and the words that are involved in the experiments described in §8.1. The five most frequently occurring (in the test set) nouns and verbs are listed in their base forms. Other lists are exhaustive.

## 8.1 Correction Scope and Procedure

Motivated by analyses based on the Japanese Learners' English corpus (see §3.1.2), we consider errors involving these four parts-of-speech:

- All **articles** and ten **prepositions**, listed in Table 8.1.
- **Noun** number.
- **Verb** aspect, mode, and tense.

The algorithm consists of two steps.

**Overgenerate** In the first step, an input sentence, which may contain errors, is reduced to a "canonical form" devoid of articles, prepositions, and auxiliaries. Furthermore, all nouns are reduced to their singular forms, and all verbs are reduced to their root forms. All of their alternative inflections are then inserted into the lattice in parallel. Insertions of articles, prepositions and auxiliaries are allowed at every position. This simple algorithm thus expands the sentence into a lattice of alternatives, as illustrated in Figure 8-1, for the reduced input sentence "*I want flight Monday*".

This step may be viewed as a type of natural language generation (NLG) that is intermediate between conventional NLG from meaning representations, and NLG from keywords[1]. Our approach is most similar to (Uchimoto et al., 2002) in the sense of stripping away and then regenerating the function words and inflectional endings.

**Rerank** In contrast to previous work on building lattices of paraphrases, such as (Barzilay & Lee, 2003), most paths in this lattice would yield ungrammatical sentences. Therefore, in the second step, a language model is used to score the various paths in the lattice. In the NITROGEN natural language generation system, an $n$-gram model is used to select the most fluent path through the word lattice (Langkilde & Knight, 1998); in other work, such as (Ratnaparkhi, 2000) and (Uchimoto et al., 2002), dependency models were used. An $n$-gram model will serve as our baseline; stochastic context-free grammar (CFG) language models will then be utilized to further re-rank the candidates proposed by the $n$-gram model. Three CFG models will be compared.

[1] See §2.4.1 for a literature review of this area.

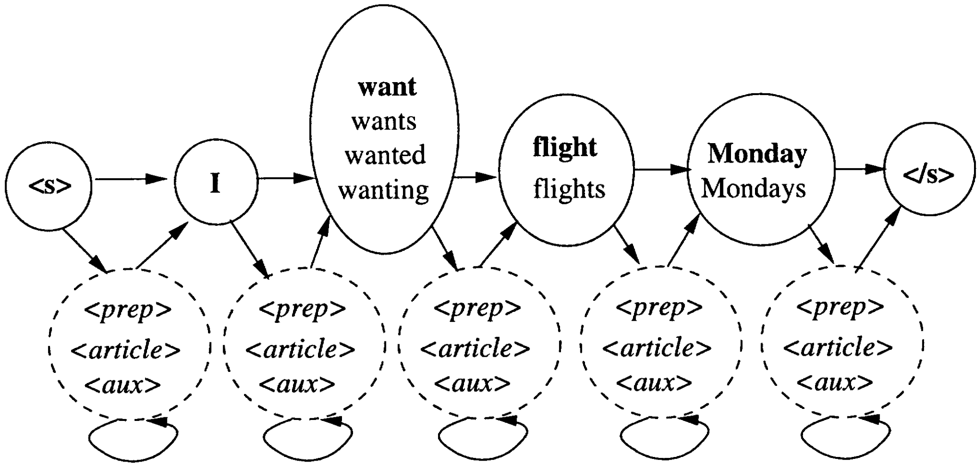


Figure 8-1: Lattice of alternatives formed from the reduced input sentence, "*I want flight Monday*". The lattice encodes many possible corrections, including different noun and verb inflections, and insertions of prepositions, articles, and auxiliaries. One appropriate correction is "*I want a flight on Monday*".

## 8.2 Grammar Resources

The grammars used in this chapter were developed for the natural language understanding system TINA (Seneff, 1992b), as part of a spoken dialog system. To parse a sentence, TINA uses a set of probabilistic context-free rules that describes the sentence structure, and a constraint-unification mechanism that handles feature agreement and movement phenomena. The probability model is applied to nodes in the parse tree, where each node's category is conditioned on its parent and left sibling. The statistics are trained on a large corpus of in-domain utterances. The output of TINA is a semantic frame, which is passed to the dialogue manager to further the conversation.

For this purpose, *domain-specific* grammars, incorporating both syntactic and semantic information into the context-free rules, were designed for each domain. While the higher levels of the parse tree capture general syntactic constraints of the English language, the lower nodes tend to capture specific meaning interpretations in the particular application domain. The grammar developed for the flight domain (Seneff, 2002) will be used in our evaluation and will be denoted PARSE$_{domain}$.

To facilitate portability and re-use of previous grammars in the development of new domains, we have more recently developed a *generic* grammar for English, which consists of syntax-oriented grammar rules that explicitly represent major syntactic constituents, such as subject, predicate, object, etc. In practice, when building a dialogue system, this grammar is manually augmented with proper names relevant to the domain, such as restaurants (Gruenstein & Seneff, 2006), travelers' phrasebook (Xu & Seneff, 2008), music, electronic programming guide and calendar management. For example, for the flight domain, names of cities, states, countries, airports and airlines have been added to the generic grammar as specific proper noun classes. The resulting grammar will be used in our evaluation and will be denoted PARSE$_{generic+geo}$.

Lastly, we will also perform evaluation with a generic grammar that is not augmented with any domain-specific proper names. The names of cities, states, countries, airports, airlines, etc. are treated as unknown nouns. This grammar will be denoted PARSE$_{generic}$.

## 8.3 Training and Evaluation

Because our methods first reduce a sentence to an impoverished, uninflected form, we can both train the system and evaluate its performance by applying it to a corpus collected from a general population. We generated reduced sentences using the scheme described in the first step in §8.1. We then measured the system's ability to recover the original articles, modals, and prepositions, and to produce correct inflectional forms for both nouns and verbs. This set-up provides an idealized situation: the error classes in the data are essentially restricted to the classes we model. Thus we are able to measure the effects of the reranking algorithms in a controlled fashion.

### 8.3.1 Data Source

The training set consists of 33,516 transcripts of utterances, produced by callers in spoken dialogues with the MERCURY flight domain (Seneff, 2002). The development set consists of 317 sentences from the same domain, and is used for tuning word transition weights for the $n$-gram model. The test set consists of 1000 sentences, also from the same domain. These utterances are all at least four words long and have an average length of 7.4 words. None of them are present in the training set. They serve as our "gold-standard" in the automatic evaluation, although we recognize that some of the users may be non-native speakers.

### 8.3.2 Overgeneration Algorithm

Starting with a backbone lattice consisting of the reduced input sentence, the following operations on the lattice are allowed:

**Free Insertions** Articles, prepositions, auxiliaries and modals are allowed to be inserted anywhere. It is possible, based on a shallow analysis of the sentence, to limit the possible points of insertion. For example, insertion of articles may be restricted to the beginning of noun phrases. However, at least in this restricted domain, these constraints were observed to have a negligible effect on the final performance.

**Noun/Verb Inflections** The nouns and verbs, appearing in their uninflected forms in the reduced input, can be substituted by any of their inflected forms. Of the nouns specified in the MERCURY grammar, 92 unique nouns appear in the test set, occurring 561 times; there are 79 unique verbs, occurring 716 times.

### 8.3.3 Reranking Strategies

The following four reranking strategies are contrasted. Details about the three parsing models can be found in §8.2.

TRIGRAM is a word trigram language model, trained on the sentences in the training set.

PARSE$_{domain}$ is the flight-domain context-free grammar, trained on the same sentences. From the 10-best list produced by the trigram language model, the candidate that yields the highest-scoring parse from this grammar is selected. If no parse tree is obtained, we default to the highest-scoring trigram hypothesis.

PARSE$_{generic}$ is a syntax-oriented, generic grammar. The re-ranking procedure is the same as for PARSE$_{domain}$.

| Reranker | Noun/Verb | Auxiliary/Preposition/Article | | |
|---|---|---|---|---|
| (# utterances) | Accuracy | Precision | Recall | F-score |
| TRIGRAM | 89.4 | 0.790 | 0.636 | 0.705 |
| PARSE$_{generic}$ | 90.1 | 0.819 | 0.708 | 0.759 |
| PARSE$_{generic+geo}$ | 89.3 | 0.823 | 0.723 | 0.770 |
| PARSE$_{domain}$ | 90.8 | 0.829 | 0.730 | 0.776 |

Table 8.2: Experimental results, broken down into the various error classes.

PARSE$_{generic+geo}$ is the same as PARSE$_{generic}$, except it is augmented with a set of proper noun classes encoding geographical knowledge, including the names of the same cities, states, countries, airports and airlines that are included in PARSE$_{domain}$. Again, the re-ranking procedure is the same as for PARSE$_{domain}$.

### 8.3.4 Results

The performances of the various re-ranking models are shown in Table 8.2. Compared with TRIGRAM, reranking with all three PARSE models resulted in improved precision and recall for the insertion of articles and prepositions. However, the accuracy in predicting nouns and verbs did not vary significantly. Overall, the quality of the output sentences of all three PARSE models is statistically significantly superior to that of TRIGRAM[2]. The differences in performance among the three PARSE models are not statistically significant[3].

The gains were due mostly to two reasons. The parser was able to reject candidate sentences that do not parse, such as “* _When does the next flight to Moscow?_”. Trigrams are unable to detect long-distance dependencies, in this case the absence of a verb following the noun phrase “_the next flight_”. Furthermore, given candidates that are syntactically valid, the parser was able to assign lower scores to those that are disfluent, such as “*_I would like that 3:55pm flights_” and “*_What cities do you know about Louisiana?_”. PARSE$_{domain}$ in fact rejects this last sentence based on constraints associated with trace movements.

Among the errors in the PARSE$_{domain}$ model, many contain parse tree patterns that are not well modelled in the probabilities in the grammar. Consider the ill-formed sentence “*_I would like a flight 324._”, whose parse tree is partially shown in Figure 8-2. Well-formed sentences such as “_I would like flight 324_” or “_I would like a flight_” have similar parse trees. The only “unusual” combination of nodes is thus the co-occurrence of the `indef` and `flight_number` nodes, which is missed by the spatial-temporal “node”-trigram model used in TINA. The grammar could likely be reconfigured to capture the distinction between a generic flight and a specific flight. Such combinations can perhaps also be expressed as features in a further reranking step similar to the one used in (Collins et al., 2005).

### 8.3.5 Multiple Correct Answers

In the evaluation above, the original transcript was considered as the sole “gold standard”. Just as in machine translation, however, there are often multiple valid corrections for one

[2] The quality is measured by word error rate (WER) — insertion, substitution and deletion — with respect to the original transcript. For each test sentence, we consider one model to be better if the WER of its output is lower than that of the other model. All three PARSE models significantly outperformed TRIGRAM at $p < 10^{-12}$, using McNemar's test.

[3] By McNemar's test, PARSE$_{domain}$ compared with PARSE$_{generic+geo}$ and PARSE$_{generic}$ yield $p = 0.495$ and $p = 0.071$, respectively.

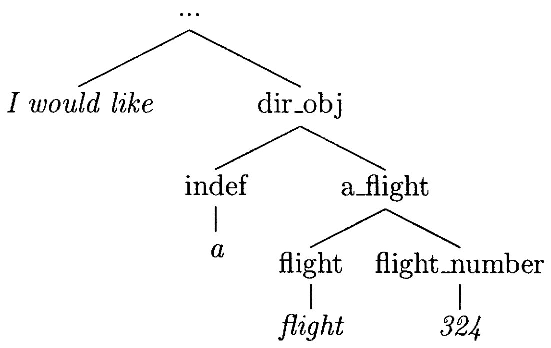


Figure 8-2: Parse tree for the ill-formed sentence "**I would like a flight 324*" in $\textsc{Parse}_{domain}$. The co-occurrence of `indef` and `flight_number` nodes should be recognized to be a clue for ungrammaticality.

| Reduced input: | when delta flight leave atlanta |
|---|---|
| Correction 1: | when does the delta flight leave atlanta |
| Correction 2: | when does the delta flight leave from atlanta |

Table 8.3: A sample entry in the human evaluation. Correction 1 is the transcript, and correction 2 is the $\textsc{Parse}_{domain}$ output. Both evaluators judged these to be equally good. In the automatic evaluation, the $\textsc{Parse}_{domain}$ output was penalized for the insertion of "from".

sentence, and so the performance level may be underestimated. To better gauge the performance, we had previously conducted a human evaluation on the output of the $\textsc{Parse}_{domain}$ model, focusing on the subset that were fully parsed by TINA. Using a procedure similar to §8.3, we harvested all output sentences from $\textsc{Parse}_{domain}$ that differed from the gold standard. Four native English speakers, not involved in this research, were given the ill-formed input, and were asked to compare the corresponding transcript (the "gold-standard") and the $\textsc{Parse}_{domain}$ output, without knowing their identities. They were asked to decide whether the two are of the same quality, or that one of the two is better. An example is shown in Table 8.3.

To measure the extent to which the $\textsc{Parse}_{domain}$ output is distinguishable from the transcript, we interpreted their evaluations in two categories: (1) category OK, when the PARSE output is as good as, or better than the transcript; and (2) category WORSE, when the transcript is better. As shown in Table 8.4, the human judges exhibited "substantial agreement" according to the kappa scale defined in (Landis & Koch, 1977). Of the 317 sentences, 218 were judged to be at least as good as the original transcript. Hence, the

| Test Set 1 | OK | WORSE | Test Set 2 | OK | WORSE |
|---|---|---|---|---|---|
| OK | 101 | 6 | OK | 91 | 3 |
| WORSE | 15 | 45 | WORSE | 25 | 41 |

Table 8.4: Agreement in the human evaluation. The test set was randomly split into two halves, Test Set 1 and Test Set 2. Two human judges evaluated Test Set 1, with kappa = 0.72. Two other evaluated Test Set 2, with kappa = 0.63. Both kappa values correspond to "substantial agreement" as defined in Landis & Koch (1977).

corrections in up to two-thirds of the sentences currently deemed incorrect may in fact be acceptable.

## 8.4 Applications to Machine Translation

While the error correction approach presented in this chapter was initially designed for non-native texts or utterances, it may also be effective on machine translation output.

To assess whether such an application is viable, we applied the approach to the output of an interlingua-based translation system (Wang & Seneff, 2006). This system requires three components for L1-to-L2 translation, as shown in Figure 8-3: a *natural language understanding system* (Seneff, 1992b), which maps a sentence in L1 to a semantic frame encoding syntax and semantics, a *transfer* phase, which modifies the semantic frame to account for linguistic properties unique to L2, and a *natural language generation system* (Baptist & Seneff, 2000), which produces a well-formed surface string in L2.

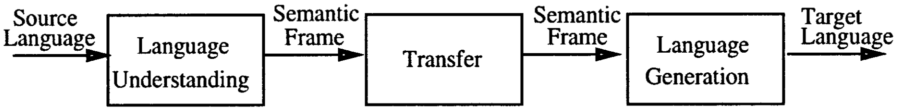


Figure 8-3: Schematic diagram of an interlingua-based translation framework.

An evaluation was performed on a set of 409 parsable Chinese inputs in a Chinese-to-English translation task in the flight domain (Seneff et al., 2006). The types of corrections were restricted to articles, noun number, verb forms, and prepositions before date and time expressions. The performance of our approach was compared to the machine translation paradigm (§2.2), by using PHARAOH (Koehn, 2004), a state-of-the-art phrase-based statistical machine translation (SMT) system to perform "bad English"-to-"good English" translation.

Human judges were asked to rate the translation output from 1 to 5, with 1 being incorrect, 3 being adequate, and 5 being perfect. The average rating of the system output was 4.27. The sentence-regeneration approach raised the average rating to 4.51, with nearly 94% of the output achieving "adequate" or better ratings. The SMT approach also raised the average rating to 4.51, and indeed improved more sentences to the 5 ("perfect") rating, but at the expense of an increase in the number of sentences with low scores.

## 8.5 Summary

In this chapter, we presented a generation-based approach for grammar correction, and performed evaluation in the flight domain using simulated data based on the four most common error classes. This approach consists of a first step of generating a lattice of possible paraphrases from an impoverished form of the input sentence, and a second step of finding the most fluent path through a language model. Language models based on stochastic context-free grammars outperformed $n$-grams, and generic grammars achieved similar performance as a domain-specific grammar. The same approach was performed on machine translation outputs, and was also shown to improve their fluency.

The central problem in correcting non-native grammatical errors, which has been our subject of inquiry, is to determine the most appropriate word, given its context within the sentence. It turns out that the reverse task — given a word in a well-formed sentence, generate *incorrect* alternatives for that word — can be effectively carried out with similar techniques. **Chapter 9** will exploit these techniques for an educational application — the generation of assessment materials for language learners.

# Chapter 9

# Personalization: The Case of Assessment Item Generation

There has been growing interest in harnessing natural language processing (NLP) techniques to enhance the quality of education. Traditional teacher-student interactions in the classroom are now supplemented with intelligent tutoring systems (Litman & Silliman, 2004), online lecture audios (Glass et al., 2007), not to mention grammatical error correction.

Due to their ability to provide automatic and objective feedback, multiple choice questions are commonly used in education applications. One type that is especially popular in language learning and assessment is fill-in-the-blank questions, or *cloze items*, where one or more words is removed from a sentence, and a number of candidate words are offered to the user for filling in the gap. An example is shown in Figure 9-1.

As a language learning tool, cloze tests can be enhanced by using up-to-date, authentic text on topics in which the student takes an interest. According to (Shei, 2001), such *personalization* can "provide motivation, generate enthusiasm in learning, encourage learner autonomy, foster learner strategy and help develop students' reading skills as well as enhance their cultural understanding".

It is clearly not practical to manually design tailor-made cloze tests for every student. This bottleneck has motivated research on automatic generation of cloze items. This chapter is concerned with generating cloze items for prepositions, whose usage often poses problems for non-native speakers of English. This problem is closely related to the grammar correction task; both involve judgments on the appropriateness of a preposition given a context. A context representation, very similar to the one utilized in **Chapter 5**, will be shown to be effective also for this problem.

The quality of a cloze item depends on the choice of distractors. We propose two methods to generate distractors for prepositions. One method is based on word collocations in standard English corpora, similar in spirit to the nearest-neighbor framework used in **Chapter 5** for preposition generation. The other exploits non-native English corpora. Both methods are found to be more successful in attracting users than a baseline that relies only on word frequency, a common criterion in past research.

## 9.1 Problem Definition

Broadly speaking, it takes the following steps to produce a cloze item from a *source corpus*:

1. Determine the *key*, i.e., the word to be removed from a sentence.

The child's misery would move even the most ____ heart.
(a) torpid (b) invidious (c) stolid (d) obdurate

Figure 9-1: An example cloze item taken from Brown:05.

If you don't have anything planned for this evening,
let's go ____ a movie.
(a) to (b) of (c) on (d) *null*

Figure 9-2: An example cloze item on prepositions, generated from the seed sentence "If you don't have anything planned for this evening, let's go to a movie". The key is "to". Distractor (b) is produced by the baseline method in §9.3.2, distractor (c) by the collocation method in §9.3.3, and distractor (d) by the non-native method in §9.3.4.

2. Select a *seed sentence* from the source corpus.
3. Generate *distractors*, i.e., incorrect choices, for the key.

Past research has focused on cloze items whose keys are of an *open-class* part-of-speech (POS), e.g., nouns, verbs, or adjectives. Words that occur relatively infrequently are selected as keys, with the intention of improving the vocabulary level of the user. The cloze item in Figure 9-1 is such an example.

While vocabulary build-up is essential, mastering the usage of function words is also important in language learning. Misuse of prepositions, for example, turns out to be a frequent type of error for Japanese speakers, according to the Japanese Learners of English corpus[1]. Cloze items on prepositions, such as the one shown in Figure 9-2, can provide training that specifically targets this type of error. This chapter is concerned with the automatic generation of such items.

Prepositions, as a closed-class POS, present some new challenges in cloze item generation. First, insertion and deletion of prepositions are common errors, whereas errors in open-class POS are predominantly substitutions. Secondly, the set of prepositions is much smaller than the set of their open-class counterparts. As a result, most prepositions are already familiar to the user, making it more difficult to select good distractors. To address these challenges, we propose two novel techniques for distractor generation.

## 9.2 Related Work

Past research has addressed both key and distractor selection for open-class POS. The key is often chosen according to word frequency (Shei, 2001; Coniam, 1998), so as to match the user's vocabulary level. Machine learning methods are applied in (Hoshino & Nakagawa, 2005) to determine the best key, using cloze items in a standard language test as training material.

### 9.2.1 Distractor Generation

The focus of this chapter is on distractor generation. As is widely observed, a good distractor must satisfy two requirements. First and foremost, it must result in an incorrect sentence.

[1] See §3.1.2.

Secondly, it must be similar enough to the key to be a viable alternative.

To secure the first requirement, the distractor must yield a sentence with zero hits on the web in (Sumita et al., 2005); in (Liu et al., 2005), it must produce a rare collocation with other important words in the sentence.

As for the second, various criteria have been proposed: matching patterns hand-crafted by experts (Chen et al., 2006); similarity in meaning to the key, with respect to a thesaurus (Sumita et al., 2005) or to an ontology in a narrow domain (Karamanis et al., 2006). However, the most widely used criterion, again, is similarity in word frequency to the key (Brown et al., 2005; Shei, 2001).

### 9.2.2 Evaluation

Mirroring the two requirements for distractors, our two main evaluation metrics are *usability* and *difficulty* of the cloze item.

#### Usability

A "usable" item has been defined in different ways, ranging from the simple requirement that only one choice is correct (Liu et al., 2005), to expert judgments (Chen et al., 2006). Others take into account the time needed for manual post-editing (Karamanis et al., 2006), in relation to designing the item from scratch. We adopt the simple requirement as in (Liu et al., 2005).

#### Difficulty

Cloze tests have been used both as a proficiency assessment tool (Brown et al., 2005) (Sumita et al., 2005) and as a language learning tool (Shei, 2001). For assessment purposes, the ability of the cloze test to discriminate between more advanced students and less advanced ones is important. This is expressed in two dimensions (Coniam, 1998), (Mitkov & Ha, 2003): First, *item difficulty* (or *facility index*), i.e., the distractor should be neither too obviously wrong nor too tricky. Second, *effectiveness* (or *discrimination index*), i.e., it should attract only the less proficient students.

For language learning applications, the discriminative power of a cloze test is not as important as its ability to cause users to make mistakes. An easy cloze test, on which the user scores perfectly, would not be very educational; arguably, the user learns most when his/her mistake is corrected. This chapter will emphasize the generation of difficult cloze items.

## 9.3 Approach

Our input is a sentence from the source corpus and its key (a preposition). The output is a distractor, which, for our purposes, is ideally the one that is most likely to attract the user (cf. §9.2.2).

### 9.3.1 Context Representation

An important question is how to represent the *context* of the preposition in the sentence. The granularity of the representation reflects a trade-off similar to precision/recall.

| Prep. | Count | Prep. | Count |
|---|---|---|---|
| to | 5140589 | on | 1351260 |
| of | 5107531 | with | 1325244 |
| in | 3645151 | at | 991039 |
| for | 1865842 | ... | ... |

Table 9.1: Preposition frequencies in a corpus of 10-million sentences from the New York Times.

Suppose one requires matching a rather large window of words centered on the preposition. With this fine-grained representation, new sentences are unlikely to match any sentences in the training set, and few cloze items can be generated. At another extreme, suppose one ignores the context, and determines the distractor solely on the basis of its frequency count. This coarse representation can produce a cloze item out of any sentence with a preposition, but it risks generating a less viable, and hence less difficult, distractor.

We now give a brief overview of the syntactic functions of prepositions (Quirk et al., 1985) in order to motivate our context representation. A preposition can be a particle in a phrasal or prepositional verb; more frequently, however, it forms a prepositional phrase (PP) with a complement, typically a noun. The PP can serve as an adverbial, a post-modifier of a noun phrase, or the complementation of a verb or an adjective.

No attempt is made to distinguish these different functions. The context of a preposition is represented by the triplet $\langle A, p, B\rangle$, where $A$ and $B$, possibly empty, are heads of the noun or verb phrases that are associated with the preposition $p$ in one of its syntactic functions described above. From the sentence "*Let's go to a movie*", for example, the triplet $\langle go, to, movie\rangle$ is extracted.

Our task is to learn a mapping from such a triplet to $\overline{p}$, the distractor which the user is most likely to confuse with $p$:

$$\langle A, p, B\rangle \longmapsto \overline{p}$$

Either $p$ or $\overline{p}$ can be an empty string, in which case it is written as *null*. If $p$ is *null*, then $A$ and $B$ are the head nouns or verbs that are to be erroneously associated with $\overline{p}$. For example, the sentence "*So we decided to take the kitty* ×*to home*" is represented as $\langle take, null, home\rangle$, with "to" as $\overline{p}$.

Thus, this mapping is sufficient to represent substitution, insertion and deletion errors. We now describe three different ways to learn this mapping: first a baseline, then two novel methods that leverage the context of the preposition.

### 9.3.2 Baseline: Using frequencies

The baseline considers only word frequency, a criterion commonly used in cloze item generation for open-class POS. Given $\langle A, p, B\rangle$, it ignores $A$ and $B$, and simply returns the $\overline{p}$ whose frequency count in a large English corpus is closest to that of $p$. According to Table 9.1, the frequency of "to" is closest to that of "of"; when the key is "to", as in the cloze item in Figure 9-2, the baseline distractor is "of". When $p$ is *null*, the baseline method stochastically generates a random preposition according to the probability distribution observed in the English corpus.

| Error | Version | Transcript | Context |
|---|---|---|---|
| Del | Corrected<br>Original | so I'd like to **go to** the **movie** with you.<br>so I'd like to **go movie** with you. | $\langle$*go,to,movie*$\rangle$<br>$\langle$*go,null,movie*$\rangle$ |
| Ins | Corrected<br>Original | So we decided to **take** the kitty **home**.<br>So we decided to **take** the kitty **to home**. | $\langle$*take,null,home*$\rangle$<br>$\langle$*take,to,home*$\rangle$ |
| Sub | Corrected<br>Original | He **studies at** the **university**.<br>He **studies in** the **university**. | $\langle$*study,at,university*$\rangle$<br>$\langle$*study,in,university*$\rangle$ |

Table 9.2: Context representations extracted from a non-native English corpus. All errors in the original sentences not involving prepositions are suppressed before extraction. One example each of insertion, deletion and substitution errors are provided.

### 9.3.3 Using collocations

The context of the preposition may be helpful in choosing attractive distractors. In terms of our evaluation metrics, a preposition that collocates frequently with *either A or B* in a large English corpus might make a more *difficult* distractor for the user; on the other hand, one that has appeared in the corpus with *both A and B* is unlikely to be *usable*.

Following this intuition, this method returns the preposition that appears frequently with either $A$ or $B$, but not both at the same time; formally, $\langle A, p, B\rangle \longmapsto \arg\max_{\overline{p}}\{c(\langle A,\overline{p},*\rangle) + c(\langle *,\overline{p},B\rangle)\}$ with the constraint that $c(\langle A,\overline{p},B\rangle) = 0$, where $c(.)$ is the count. Consider the cloze item in Figure 9-2. On the strength of the popularity of the collocation "go on", and the non-occurrence of $\langle$*go,on,movie*$\rangle$ in the English corpus, the preposition "on" is selected as the distractor.

### 9.3.4 Using a non-native English corpus

From a corpus of non-native sentences and their corrections, mappings from a triplet to a preposition mistake can be directly estimated. Table 9.2 illustrates the context extraction of prepositions in such a corpus. The most frequent mistake for each context would then make a reasonable distractor; formally, $\langle A, p, B\rangle \longmapsto \arg\max_{\overline{p}}\{c(\langle A,\overline{p},B\rangle)\}$. For example, the cloze item in Figure 9-2 has *null* as the distractor because, for the triplet $\langle$*go,to,movie*$\rangle$, the deletion error is more common than substitution errors in the non-native corpus.

Indeed, more than half of the preposition mistakes in the JLE corpus are deletion errors. One advantage of using a non-native corpus is the ability to directly model contexts where deletion errors are common. It is difficult to do so with native English corpora only, as in the two methods above.

The main drawback is data sparseness[2]. Compared to normal English corpora, non-native corpora are much more expensive to collect; they tend to be much smaller, and restricted to speakers of only a few mother tongues, if not just one.

## 9.4 Experiments

This section describes experiments that compare the quality of the distractors generated by the three methods described in §9.3.2, §9.3.3 and §9.3.4. The distractors will be referred to

[2] A possible mitigation of this problem, which we have not yet explored, is to initially generate cloze items from collocations only, then harvest the mistakes made by users to grow a non-native corpus.

as the *baseline distractor*, *collocation distractor* and *non-native distractor*, respectively. We begin by discussing our corpora.

### 9.4.1 Set-up

The 72 prepositions listed in (Quirk et al., 1985) are considered to be the set of prepositions. The context representations are extracted from parse trees derived by a statistical parser (Collins & Koo, 2005).

**English corpus** The English corpus consists of about 10 million sentences from the New York Times.

**Non-native corpus** The non-native corpus is the Japanese Learners of English corpus[3], which contains about 1,300 instances of preposition mistakes. As illustrated in Table 9.2, one $\langle A, p, B\rangle$ and one $\langle A, \overline{p}, B\rangle$ are harvested from each mistake.

**Source corpus** The source corpus is the BTEC corpus (Takezawa et al., 2002), used in the evaluation campaign of the International Workshop on Spoken Language Translation. It consists of about 24,000 transcripts from the travel domain. Only sentences at least five words long are utilized.

To ensure a fair comparison, a sentence qualifies as a seed sentence only when all three methods can generate a distractor. In practice, the non-native corpus is the constraining factor. To select the most reliable distractors, we require the seed sentence's triplet $\langle A, p, B\rangle$ to occur two times or more in the non-native corpus, or its $\langle A, p, *\rangle$ or $\langle *, p, B\rangle$ to occur four times or more. With these restrictions, 328 cloze items were generated.

Interestingly, the three methods rarely generate the same distractor. The non-native distractor agrees with the collocation distractor 9.5% of the time, and intersects with the baseline only 4.4% of the time. The collocation and baseline distractors are identical 12.7% of the time. Most cloze items thus offer four different choices.

### 9.4.2 Analyses

#### Usability

A cloze item is considered usable when all distractors result in an inappropriate sentence (Liu et al., 2005). A native speaker of English, who was not involved in the cloze item generation process, took the cloze test and identified all choices which yielded acceptable English sentences.

In 12 out of 328 cloze items, one of the distractors yielded a correct sentence; in other words, 96.3% of the automatically generated items were usable. To put this performance level in context, usability rates of 93.5% (Sumita et al., 2005) and 91.5% (Liu et al., 2005) have been reported in the literature, although their tasks and corpora are different, and the results are hence not directly comparable.

Among the unusable distractors, more than half are collocation distractors. For example, from the seed sentence "*I have sore pain here*", the collocation method produces the distractor "around", yielding an acceptable sentence "*I have sore pain around here*".

[3] See §3.1.2.

| Test 1 | | Test 2 | |
|---|---|---|---|
| Subject | Score | Subject | Score |
| Student 1 | 75.9% | Student 3 | 91.1% |
| Student 2 | 76.6% | Student 4 | 91.1% |

Table 9.3: *Overall performance on the cloze tests.*

| Subject | Non-native | Collocation | Baseline |
|---|---|---|---|
| Student 1 | 17 | 10 | 10 |
| Student 2 | 20 | 12 | 6 |
| Student 3 | 4 | 9 | 2 |
| Student 4 | 6 | 8 | 2 |
| Total | **47** | **39** | **20** |

Table 9.4: *Number of distractors chosen by subjects.*

### Difficulty

After omitting the unusable cloze items identified in the previous step, we split the remaining 316 cloze items into two tests, with 158 questions each. Our subjects are four students whose mother tongue is Mandarin. They are all students in their second or third year of senior high school in Taiwan.

| | |
|---|---|
| It's really different driving ____ the right side of the street | |
| (a) on [key] | **(b) *null* [non-native]** |
| (c) with [baseline] | **(d) *to* [collocation]** |
| Could I take the leftovers ____ home? | |
| **(a) *in* [collocation]** | (b) about [baseline] |
| **(c) *to* [non-native]** | (d) *null* [key] |

Figure 9-3: *Some cloze items for which both human subjects made an error. The* **bolded** *items are the selected distractors.*

The overall performance of the subjects is listed in Table 9.3. The subjects made a total of 106 mistakes, of which 12% are insertions, 29% are deletions, 58% are substitutions. A breakdown of the distractors responsible for the mistakes is provided in Table 9.4 with respect to the subjects, and in Table 9.5 with respect to error types. Figure 9-3 shows a few cloze items for which both human subjects made an error.

Overall, distractors produced by the collocation and non-native methods were more successful in attracting the subjects. The subjects were two to three times more likely to choose a non-native distractor[4] than a baseline one; with the exception of Student 1, the same difference is observed between the collocation and baseline distractors.

[4] This is true in spite of the fact that the native language of the students being tested (Mandarin) is different from the native language of those who contributed to the non-native training corpus, namely the Japanese Learners of English corpus.

| Error | Non-native | Collocation | Baseline |
|---|---|---|---|
| Type | Success (Total) | Success (Total) | Success (Total) |
| Del | 31 (211) | 0 (0) | 0 (0) |
| Ins | 5 (79) | 5 (79) | 3 (79) |
| Sub | 11 (26) | 34 (237) | 17 (237) |
| Total | **47** (316) | **39** (316) | **20** (316) |

Table 9.5: A breakdown of the the distractors into the error types. "Success" refers to the number of distractors that were selected as answer by the subjects, out of the "Total" number that appeared in the cloze tests.

## 9.5 Summary

After focusing on the grammatical error correction task in previous chapters, this chapter examined the closely related problem of generating assessment items for language learning. Prepositions present some new challenges in cloze item generation, for which we have proposed two novel distractor generation methods, one based on collocations, the other on direct observations in a non-native corpus. We have compared them with a baseline method based on word frequency. The distractors generated by the two novel methods were more successful in attracting the subjects than the baseline method.

This chapter ends our discussion on error correction. In the next and final chapter, we conclude and propose future research directions. Among these directions is the generalization of the work in this chapter, to other types of assessment items, and to subjects beyond language learning (§10.5).

# Chapter 10

# Conclusions & Future Directions

This dissertation has proposed and evaluated methods for correcting grammatical errors made by non-native speakers of English. Significant novelties in these methods include the combination of linguistic and statistical techniques, and several aspects of personalization. The effectiveness of these methods has been demonstrated on prepositions, verb forms, and articles, among other error categories.

This final chapter begins with concluding remarks (§10.1), summarizing our contributions. It will then look forward and sketch four areas of future research (§10.2-§10.5).

## 10.1 Conclusions

This dissertation began with a literature review and data analysis. The review (**Chapter 2**) sketches the prevailing framework adopted by the research community — viewing the grammar correction task as a classification problem, and using words within a fixed window as features. This strategy enables robust feature extraction even in noisy texts, but suffers from the limitation that the neighboring words alone do not always provide adequate context. This motivates the *use of linguistic analysis*, which can help identify salient words located at longer distances from the target word.

The analysis of a non-native English corpus compiled in Japan (**Chapter 3**) reveals patterns of the errors made by native speakers of Japanese. These patterns suggest one dimension of *personalization* — awareness of the author's mother tongue — that can potentially enhance performance[1].

The use of linguistic analysis is illustrated with prepositions. If a preposition serves as an argument marker of a verb, for example, then the verb is informative for predicting the preposition, even if it is located far from it. Indeed, when the prepositional phrase attachment site is taken into account, accuracy in preposition generation improves (**Chapter 5**). This kind of linguistic analysis, however, cannot be naively performed on non-native, noisy texts. The potential pitfall is illustrated in the context of verb conjugations (**Chapter 6**). The correction procedure is made more robust by recognizing ill-formed parse trees.

A second dimension of personalization — the proficiency of the author — is introduced in the context of inserting missing articles. The aggressiveness of the insertion model is adjusted according to an estimation of the initial quality of the input text (**Chapter 7**).

[1] This particular research direction, however, must await data from native speakers of other languages. Future work is discussed in §10.3.2.

For sentences that require major repair, the classification approach does not suffice. Instead, in a restricted domain, these sentences can be processed by a sentence re-generation framework that combines syntactic and semantic analysis to rerank candidates (**Chapter 8**).

The last topic returns to the theme of personalization, addressing a third dimension — the interest of the user. Personalized multiple-choice items for prepositions, tailored to the user's domain of interest, can be automatically generated using corpus statistics (**Chapter 9**).

We feel that we have barely scratched the surface, and that there are many research directions to deepen and broaden our understanding of non-native errors. Our approach can be refined to enrich the output content (§10.2) and to improve performance (§10.3); evaluation methodology can be made more accurate (§10.4); and a greater variety of assessment materials may be automatically generated (§10.5).

## 10.2 Richer Output

Our discussion of future directions starts with a few possible extensions in the output. This dissertation has discussed strategies for correcting the usage of preposition (**Chapter 5**), verb forms (**Chapter 6**) and articles (**Chapter 7**). A logical extension is to expand the coverage to other error classes (§10.2.1). Furthermore, rather than simply returning the recommended or predicted word, the output of the posteditor can be expanded to other informative feedback, such as confidence scores (§10.2.2) and high-level comments (§10.2.3).

### 10.2.1 Coverage

Although less common than those treated in this dissertation, a few other categories of grammatical mistakes in English are worth the attention of the research community. These include usage of noun number (i.e., singular or plural), kinds of verb tense (i.e., present, past or continuous), and word order. Multiple-choice test items for these error classes may also be generated.

### 10.2.2 Confidence Score

Precision is important in computer-assisted language learning. More harm is done when the system proposes an inappropriate correction, than when it leaves a mistake unflagged. An important point to consider is that the system need not feel obliged to alert the student of every error it detects.

One way to improve precision is to compute a confidence measure, proposing a correction only when the measure exceeds a threshold. The measure can be, for example, the difference between the scores of the top- and second-place candidates given by the NLG component (Chodorow et al., 2007). This kind of measure may potentially be sharpened by exploiting known tendencies of the author of the text, which may be estimated by the method described in §10.3.2.

Likewise, in a conversational language learning system (**Chapter 8**), the student can first engage in an interactive dialogue with the system, during which it would conceivably apply an error-correction algorithm in order to increase the probability of obtaining a correct meaning analysis. Any differences between the "corrected" hypothesis and the original input would be recorded in a log file, along with associated parse scores. In a follow-up interaction, the system would provide explicit feedback about the previous dialogue. It could afford to be selective at this point, informing the student only of errors where it has high confidence. This conservative approach would greatly reduce the likelihood that the system misinforms the student.

### 10.2.3 Feedback

Besides the recommended word, it may be desirable to provide higher-level comments and additional real-life examples to the user. One of the motivations behind the use of the nearest-neighbor framework in **Chapter 5** is to facilitate this possibility.

High-level comments may include more detailed explanations (e.g., "the verb '*want*' takes an infinitive"), or general principles ("A verb to-be is needed for the continuous sense"). To help prioritize and tailor the feedback, it may be possible to use an error model (see §10.3.2), or a user profile that is updated on-line (Michaud et al., 2000).

## 10.3 Better Approaches

We now turn to three possible improvements in our approach. The first two propose novel features: features to capture the context (§10.3.1) and to reflect influences from one's native language (§10.3.2). The third aims to make the part-of-speech tagger and natural language parser more robust (§10.3.3).

### 10.3.1 Importance of Context

It goes without saying that the context of the sentence plays a significant role in determining the use of articles, prepositions and verb forms in the sentence. An annotation exercise on noun number and articles showed that agreement increases when human judges are given the context[2]. For example, the presence of an anaphor (e.g., "*Harvard*") in preceding sentences would be an important factor in deciding whether a definite article "*the*" is suitable for a noun (e.g., "*university*"). These observations give evidence that our algorithms should also benefit from features that capture contextual information. Yet, we relied predominantly on intra-sentential features to disambiguate the articles, prepositions and verb forms.

The challenge is how to represent the context. In **Chapter 7**, our only inter-sentential feature, `Referent`, rather naïvely assumed that the referent was explicitly mentioned using the same noun within five preceding sentences. Similarly, in (Han et al., 2005), occurrences of the noun in previous NPs are included as a feature. Techniques in anaphora and coreference resolution, e.g., (Ng & Cardie, 2002), could help refine this feature.

### 10.3.2 L1-aware Error Modeling

The grammatical errors made by non-native speakers may be influenced by their first language, and are hardly random. Understanding the nature of these errors, and the contexts in which they occur, can be helpful to grammar checking systems, just as pronunciation tutoring systems have benefited from adaptation of their acoustic models with non-native speech corpora (Tsubota et al., 2004). However, to our knowledge, there has not been any reported effort to build non-native language models automatically from corpora.

[2] John Lee, Joel Tetreault and Martin Chodorow, "Human Evaluation of Article and Noun Number Usage: Influences of Context and Construction Variability", *in submission.*

An automatic method was presented in **Chapter 3** to estimate an error model from a non-native English corpus. All three paradigms for grammar checking, discussed in **Chapter 2**, can benefit from such a model:

**MT approach (§2.2): Simulating Errors** In the statistical machine translation paradigm, the parallel corpus is created by introducing errors into well-formed text. The quality of the resulting "translation" model depends on both the frequency and authenticity of the errors.

Without *a priori* knowledge of the error frequency, an equal number of grammatical and ungrammatical sentences were represented in the "source language" in (Brockett et al., 2006). Although the MT approach is not followed in the restoration of missing articles in **Chapter 7**, prior information about the error frequency is expressed in a series of training datasets, in which progressively more articles were removed. These datasets yielded models that insert articles with varying degrees of aggressiveness, intended to handle input texts of different qualities. In both of these cases, an error model can help optimize the error frequency in the simulated training data.

An error model may also improve the authenticity of the simulated data[3]. For example, in **Chapter 7**, rather than randomly removing articles, the model can bias towards nouns whose articles are more likely to be omitted by non-native speakers.

**Parsing (§2.3): Anticipating Errors** In these approaches, the grammatical errors to be covered must be anticipated in advance; typically, the set of errors is compiled using anecdotal observations from domain experts. A systematic, corpus-based analysis can help harvest errors that might otherwise be overlooked.

**NLG approach (§2.4): Estimating Confidence** As discussed above (§10.2.2), a confidence measure is a useful feature in a posteditor, and it may potentially be sharpened by exploiting known tendencies of the author of the text. Suppose the input sentence is "*We had *the dinner at a restaurant*", and the NLG system proposes both "*null*" and "*a*". If an error model informs us that, in the context of the noun "*dinner*", the author is more likely to commit the error [*null*→*the*] than [*a*→*the*], then the deletion can be made with more confidence.

### 10.3.3 Tagger and Parser Robustness

Ideally, the parse tree of a non-native sentence should differ from that of its corrected version only in those subtrees where corrections are made. In practice, non-native grammatical mistakes often lead to parsing "errors"[4], which in turn can cause errors in the extraction of context features, and ultimately lower the accuracy in correction. For example, the accuracy of article generation degraded by more than 5% when 30% of the articles in a sentence were dropped, and by more than 11% when 70% were dropped (see §7.1.3).

One solution, adopted in **Chapter 6** for verb forms, is to observe how tree structures may be "perturbed" by non-native errors. Given the (possibly) perturbed tree of an unseen sentence, we try to recover the "original" form in the course of the correction algorithm.

[3] Furthermore, authentically simulated data can serve as a training resource for the generation of cloze items, in the framework described in **Chapter 9**.

[4] For example, see those caused by misused verb forms in Table 6.6.

Another possible solution is to retrain the part-of-speech tagger and the natural language parser on sentences with missing articles (Foster et al., 2008). New training sets for the tagger and parser could be created by injecting errors into well-formed texts. Similar to the MT paradigm, the success of this approach hinges on the quality of the simulation, for which the error model described in §10.3.2 may be helpful.

## 10.4 More Accurate Evaluation

Evaluation should not ignore the widely recognized phenomenon of multiple valid answers (§10.4.1). To facilitate direct comparisons, the research community should also consider compiling common resources for evaluation (§10.4.2).

### 10.4.1 Multiple Valid Answers

The use of standard, well-formed English corpora in evaluation, discussed in §3.1.1, typically makes the assumption that the original text is the only correct answer. However, just as in machine translation, there may be multiple valid corrections for an ill-formed sentence. For example, "**I found answer to equation*" may be interpreted as "*I found an answer to the equation*", "*I found the answer to the equation*", "*I found the answer to the equations*", etc[5].

Three studies have tested the one-answer assumption by asking human raters to compare the system prediction and the original text, when they differ. In a study of prepositions, 28% of these cases were found to be equal to or better than the original preposition (Tetreault & Chodorow, 2008a). In **Chapter 8**, in the flight domain, where multiple parts-of-speech are predicted, 66.7% of the cases were found to be acceptable. A recent study[6] confirmed the same phenomenon in the use of article and noun number; 18% of the time, native speakers judged two or more combinations of article/noun number to be perfect choices for the same noun in a sentence.

### 10.4.2 Common Resources

Common evaluation data sets have enabled other NLP subfields, such as machine translation, to make rapid progress. Unfortunately, a shared task in grammatical error correction is not yet feasible, due to a lack of annotated corpora of non-native writing. The compilation of such corpora deserves serious consideration[7].

An alternative is to use the output of MT systems as a proxy to non-native writing. The results in **Chapter 4** are encouraging in this respect. An additional advantage for MT corpora is the availability of multiple correct answers, or gold standards, in the form of multiple human translations. However, the gold-standard translations can diverge significantly from the machine output, adding challenges in modeling the editing process. It may also be worth exploring techniques for interpolating large MT training corpora and smaller non-native training corpora.

[5] An analysis of the context might rule out some of these choices, but such analysis is beyond the state-of-the-art in natural language understanding.

[6] John Lee, Joel Tetreault and Martin Chodorow, "Human Evaluation of Article and Noun Number Usage: Influences of Context and Construction Variability", *in submission.*

[7] Non-native data would also benefit multiple-choice item generation for language learning. One bottleneck in **Chapter 9** is the small number of examples observable in the non-native corpus.

| |
|---|
| **Question:**<br>Identify three ways in which an animal's body can respond to an invading pathogen. |
| **Concepts:**<br>(1) temperature change or fever;<br>(2) water loss;<br>(3) production of more mucuous; ... |

Table 10.1: Example of a constructed response item. It consists of a question and a scoring rubric, which specifies the concepts expected in the response, and also some paraphrases or model answers (not shown) of these concepts. The response is then scored based on a similarity metric with the model answers.

## 10.5 Greater Variety of Assessment Materials

Besides correction, proficiency assessment is also an important element of the language learning process. Teachers spend a significant amount of time and effort to design tests; students also need practice material for self-assessment and exam preparation. It would be helpful to both teachers and students if assessment items can be automatically generated, along similar lines as **Chapter 9**.

There is, however, no reason to limit oneself to language learning assessment. In principle, these items can be automatically generated from arbitrary textbooks, or online resources such as Wikipedia and other encyclopedia entries. For example, from a set of online Biology textbooks, the system can draft a set of questions for a Biology test. As the web increasingly becomes an arena for learning, these automatically generated exercises can enhance many web-based learning systems.

Tests may include both multiple-choice and constructed-response items. An example of the latter kind, taken from (Sukkarieh & Bolge, 2008), is presented in Table 10.1. Besides the research discussed in **Chapter 9**, there has recently been much research elsewhere on automatic design of multiple-choice items (Chen et al., 2006; Brown et al., 2005). The proposed methods, however, cannot be directly transferred to constructed-response items, because these items require the system to evaluate free-text responses. A possible research direction is to automate both the question generation and model-answer generation processes. With a collection of electronic documents as input, the task is to produce questions about the most important concepts, as well as the corresponding model answers.

### 10.5.1 Previous Work

The crucial first step in our task is to identify concepts that are worthy to be formulated into questions. State-of-the-art systems such as (Mitkov et al., 2006) depend solely on frequency counts to select concepts. These counts do not always reflect the significance of a concept. Indeed, more than half of the generated items in (Mitkov et al., 2006) were deemed unworthy.

A second shortcoming is that concepts are restricted to nouns in declarative sentences, thus ruling out how- and why-questions. This restriction severely limits the diversity of questions expected in a well-balanced test.

Third, model answers need to be manually provided. For example, at Educational Testing Service, a major administrator of standard tests, human developers are responsible for generating both the questions and the model answers, which are then fed to *c-*

*rater* (Sukkarieh & Bolge, 2008) for scoring.

### 10.5.2 Possible Approach

Recent advances in the multi-document summarization and question-answering (QA) research communities may be leveraged to solve this problem.

Automatic summarizers extract the most important sentences from a set of documents. They typically start by identifying similar text segments, using word stems, WordNet, and predicate-argument structures (McKeown et al., 2001). They then cluster similar segments into themes to be summarized. To produce a good summary, summarizers also need to recognize redundant sentences, or paraphrases. Indeed, repeated mention of the same fact in different sentences is a good indicator of its worthiness to be included in the summary.

In many respects, the challenges in multi-document summarization closely mirror those in our task: from a set of textbooks, we need to identify the most significant themes, from which meaningful questions can be formulated. Each theme is likely to contain facts mentioned in multiple sentences in different wording, thus naturally providing model answers for the scoring rubric.

From each theme, a question is to be formulated. This is no trivial task, since the sentences may have considerably different surface forms. This problem can be attacked by adapting machine learning methods for the QA task, which treat the reverse problem: given a question, detect the different surface forms in which an answer is expressed (Clarke et al., 2001).

# Bibliography

Bangalore, S. & Rambow, O. 2000. Exploiting a Probabilistic Hierarchical Model for Generation. In *Proc. COLING*. Saarbruecken, Germany.

Baptist, L. & Seneff, S. 2000. GENESIS-II: A Versatile System for Language Generation in Conversational System Applications. In *Proc. ICSLP* (pp. 271–274). Beijing, China.

Barzilay, R. & Lee, L. 2003. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In *Proc. HLT-NAACL* (pp. 16–23). Edmonton, Canada.

Bender, E. M., Flickinger, D., Oepen, S., Walsh, A. & Baldwin, T. 2004. ARBORETUM: Using a Precision Grammar for Grammar Checking in CALL. In *Proc. InSTIL/ICAL Symposium: NLP and Speech Technologies in Advance Language Learning Systems.* Venice, Italy.

Bies, A., Ferguson, M., Katz, K. & MacIntyre, R. 1995. Bracketing guidelines for treebank ii style. In *Technical Report, University of Pennsylvania.* Philadelphia, PA.

Birnbaum, M. 2005. Composition Corrector – A Browser-Based Tool for Improving Written English Grammar - with Attention to Errors of ASL Users. Talk abstract at the Computer Science and Artificial Intelligence Laboratory at MIT.

Bitchener, J., Young, S. & Cameron, D. 2005. The effect of different types of corrective feedback on esl student writing. *Journal of Second Language Writing, 14(3)*, 191–205.

Bolioli, A., Dini, L. & Malnati, G. 1992. JDII: Parsing Italian with a Robust Constraing Grammar. In *Proc. COLING.* Nantes, France.

Brill, E. & Moore, R. 2000. An Improved Error Model for Noisy Channel Spelling Correction. In *Proc. ACL* (pp. 286–293). Hong Kong, China.

Brockett, C., Dolan, W. & Gamon, M. 2006. Correcting esl errors using phrasal smt techniques. In *Proc. ACL.* Sydney, Australia.

Brown, J. C., Frishkoff, G. A. & Eskenazi, M. 2005. Automatic question generation for vocabulary assessment. In *Proc. HLT-EMNLP.* Vancouver, Canada.

Burstein, J., Chodorow, M. & Leacock, C. 2004. Automated essay evaluation: The criterion online writing service. *AI Magazine, 25(3)*, 27–36.

Bustamante, F. R. & Leon, F. S. 1996. GramCheck: A Grammar and Style Checker. In *Proc. COLING.* Copenhagen, Denmark.

Carlberger, J., Domeij, R., Kann, V. & Knutsson, O. 2005. The Development and Performance of a Grammar Checker for Swedish. *Natural Language Engineering, 1(1).*

Carlson, A., Rosen, J. & Roth, D. 2001. Scaling Up Context-Sensitive Text Correction. In *IAAI* (pp. 45–50). Seattle, WA.

Chen, C.-Y., Liou, H.-C. & Chang, J. S. 2006. Fast — an automatic generation system for grammar tests. In *Proc. COLING/ACL Interactive Presentation Sessions.* Sydney, Australia.

Chodorow, M. & Leacock, C. 2000. An Unsupervised Method for Detecting Grammatical Errors. In *Proc. ANLP-NAACL* (pp. 140–147). Seattle, WA.

Chodorow, M., Tetreault, J. & Han, N.-R. 2007. Detection of grammatical errors involving prepositions. In *Proc. ACL-SIGSEM Workshop on Prepositions.* Prague, Czech Republic.

Clarke, C., Cormack, G. & Lynam, T. 2001. Exploiting Redundancy in Question Answering. In *Proc. SIGIR.*

Collins, M. 1997. Three generative, lexicalised models for statistical parsing. In *Proc. ACL.* Madrid, Spain.

Collins, M. 1999. *Head-Driven Statistical Models for Natural Language Parsing.* PhD thesis, University of Pennsylvania, Philadelphia, PA.

Collins, M. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics, 29(4)*, 589–637.

Collins, M. & Brooks, J. 1995. Prepositional phrase attachment through a backed-off model. In *Proc. 3rd Workshop on Very Large Corpora.* Cambridge, MA.

Collins, M. & Koo, T. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics, 31(1)*, 25–69.

Collins, M., Roark, B. & Saraclar, M. 2005. Discriminative Syntactic Language Modeling for Speech Recognition. In *Proc. ACL* (pp. 507–514). Ann Arbor, MI.

Coniam, D. 1998. From text to test, automatically — an evaluation of a computer cloze-test generator. *Hong Kong Journal of Applied Linguistics, 3(1)*, 41–60.

Copestake, A. 2002. *Implementing Typed Feature Structure Grammars.* Stanford, CA: CSLI Publications.

Corston-Oliver, S., Gamon, M. & Brockett, C. 2001. A machine learning approach to the automatic evaluation of machine translation. In *Proc. ACL.* Toulouse, France.

Daelemans, W., van den Bosch, A. & Weijters, A. 1997. Igtree: Using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review, 11*, 407–423.

Daelemans, W., van den Bosch, A. & Zavrel, J. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning, 34*, 11–41.

Daume, H., Knight, K., Langkilde-Geary, I., Marcu, D. & Yamada, K. 2002. The Importance of Lexicalized Syntax Models for Natural Language Generation Tasks. In *Proc. 2nd International Conference on Natural Language Generation (INLG).*

Dong, B., Zhao, Q., Zhang, J. & Yan, Y. 2004. Automatic assessment of pronunciation quality. In *Proc. ISCSLP.* Hong Kong, China.

Eeg-Olofsson, J. & Knutsson, O. 2003. Automatic Grammar Checking for Second Language Learners – the Use of Prepositions. In *Proc. NoDaLiDa.* Reykjavik, Iceland.

Felice, R. D. & Pulman, S. 2007. Automatically acquiring models of preposition use. In *Proc. ACL-SIGSEM Workshop on Prepositions.* Prague, Czech Republic.

Felice, R. D. & Pulman, S. 2008. A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *Proc. COLING* (pp. 169–176). Manchester, UK.

Fok, A. W. P. & Ip, H. H. S. 2006. The Realization and Evaluation of a Web-Based E-Pedagogy Approach for English Second Language (ESL) Learning in Elementary Schools. In *Proc. ICWL.*

Foster, J. 2003. Parsing Ill-Formed Text using an Error Grammar. In *Proc. Artificial Intelligence/Cognitive Science Conference (AICS)* (pp. 55–60). Dublin, Ireland.

Foster, J. 2005. *Good Reasons for Noting Bad Grammar: Empirical Investigations into the Parsing of Ungrammatical Written English.* PhD thesis, Trinity College, University of Dublin, Dublin, Ireland.

Foster, J. 2007. Treebanks Gone Bad: Generating a Treebank of Ungrammatical English. In *Proc. IJCAI Workshop on Analytics for Noisy Unstructured Data.* Hyderabad, India.

Foster, J., Wagner, J. & van Genabith, J. 2008. Adapting a WSJ-Trained Parser to Grammatically Noisy Text. In *Proc. ACL (Short Papers)* (pp. 221–225). Columbus, OH.

Fouvry, F. 2003. Constraint Relaxation with Weighted Feature Structures. In *Proc. 8th International Workshop on Parsing Technologies.* Nancy, France.

Fujita, A., Inui, K. & Matsumoto, Y. 2004. Detection of Incorrect Case Assignments in Paraphrase Generation. In *Proc. IJCNLP* (pp. 555–565). Hainan Island, China.

Gamon, M., Aue, A. & Smets, M. 2005. Sentence-level MT Evaluation without Reference Translations: Beyond Language Modeling. In *Proc. EAMT.*

Gamon, M., Gao, J., Brockett, C., Klementiev, A., Dolan, W., Belenko, D. & Vanderwende, L. 2008. Using Contextual Speller Techniques and Language Modeling for ESL Error Correction. In *Proc. IJCNLP.* Hyderabad, India.

Gao, J. 2004. English Writing Wizard-Collocation Checker. Demonstrated at the 2004 Microsoft Research Asia Faculty Summit, and its abstract published on http://research.microsoft.com/asia/asiaur/summit04/demofest.aspx.

Glass, J., Hazen, T., Cyphers, S., Malioutov, I., Huynh, D. & Barzilay, R. 2007. Recent Progress in the MIT Spoken Lecture Processing Project. In *Proc. Interspeech.* Antwerp, Belgium.

Granger, S. 1999. Use of Tenses by Advanced EFL Learners: Evidence from an Error-tagged Computer Corpus. In H. Hasselgard & S. Oksefjell (Eds.), *Out of Corpora – Studies in Honour of Stig Johansson* (pp. 191–202). Amsterdam & Atlanta: Rodopi.

Gruenstein, A. & Seneff, S. 2006. Context-Sensitive Language Modeling for Large Sets of Proper Nouns in Multimodal Dialogue Systems. In *Proc. IEEE/ACL Workshop on Spoken Language Technology.* Palm Beach, Aruba.

Gui, S. & Yang, H. 2003. *Zhongguo Xuexizhe Yingyu Yuliaoku.* Shanghai, China: Shanghai Waiyu Jiaoyu Chubanshe.

Habash, Z. 1982. Common Errors In The Use of English Prepositions In The Written Work Of UNRWA Students At The End Of The Preparatory Cycle In The Jerusalem Area. Master's thesis, Birzeit University, Ramallah.

Hakuta, K. 1976. A case study of a japanese child learning english as a second language. *Language Learning, 26*, 321–351.

Han, N.-R., Chodorow, M. & Leacock, C. 2004. Detecting Errors in English Article Usage with a Maximum Entropy Classifier Trained on a Large, Diverse Corpus. In *Proc. LREC.*

Han, N.-R., Chodorow, M. & Leacock, C. 2005. Detecting errors in english article usage by non-native speakers. *Natural Language Engineering, 1*, 1–15.

Heidorn, G. 2000. *Handbook of Natural Language Processing*, chapter Intelligent Writing Assistance. Marcel Dekker, Inc.

Heine, J. E. 1998. Definiteness predictions for japanese noun phrases. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING/ACL-98)* (pp. 519–525). Montréal, Canada.

Hirst, G. & Budanitsky, A. 2005. Correcting Real-word Spelling Errors by Restoring Lexical Cohesion. *Natural Language Engineering, 11(1)*, 87–111.

Hoshino, A. & Nakagawa, H. 2005. A real-time multiple-choice question generator for language testing: A preliminary study. In *Proc. 2nd Workshop on Building Educational Applications using NLP.* Ann Arbor, MI.

Ishioka, T. & Kameda, M. 2006. Automated japanese essay scoring system based on articles written by experts. In *Proc. ACL.* Sydney, Australia.

Izumi, E., Uchimoto, K. & Isahara, H. 2004a. *Nihonjin 1200-nin no eigo supikingu kopasu.* Tokyo, Japan: ALC Press.

Izumi, E., Uchimoto, K. & Isahara, H. 2004b. The NICT JLE Corpus: Exploiting the Language Learners' Speech Database for Research and Education. *International Journal of the Computer, the Internet and Management, 12(2)*, 119–125.

Izumi, E., Uchimoto, K., Saiga, T., Supnithi, T. & Isahara, H. 2003. Automatic Error Detection in the Japanese Learners' English Spoken Data. In *Proc. ACL.* Sapporo, Japan.

Joachims, T. 1999. *Advances in Kernel Methods - Support Vector Learning*, chapter Making Large-Scale SVM Learning Practical. MIT-Press.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *Proc. SIGKDD.* Edmonton, Canada.

Johnson, W. L., Marsella, S. & Vihjálmsson, H. 2004. The darwars tactical language training system. In *Proc. Interservice/Industry Training, Simulation and Educational Conference.*

Karamanis, N., Ha, L. A. & Mitkov, R. 2006. Generating multiple-choice test items from medical text: A pilot study. In *Proc. 4th International Natural Language Generation Conference.* Sydney, Australia.

Kauchak, D. & Elkan, C. 2003. Learning Rules to Improve a Machine Translation System. In *Proc. European Conference on Machine Learning (ECML).*

Knight, K. & Chander, I. 1994. Automated Postediting of Documents. In *Proc. AAAI.* Seattle, WA.

Koehn, P. 2004. PHARAOH: A Beam Search Decoder for Phrase-based Statistical Machine Translation Models. In *Proc. AMTA.* Washington, DC.

Landis, J. R. & Koch, G. G. 1977. The measurement of observer agreement for categorical data. *Biometrics*, *33(1)*, 159–174.

Langkilde, I. 2000. Forest-based Statistical Sentence Generation. In *Proc. NAACL.*

Langkilde, I. & Knight, K. 1998. Generation that Exploits Corpus-based Statistical Knowledge. In *Proc. COLING-ACL.*

Lee, J. 2004. Automatic Article Restoration. In *Proc. HLT-NAACL Student Research Workshop.* Boston, MA.

Lee, J. & Knutsson, O. 2008. The role of pp attachment in preposition generation. In *Proc. CICLing.* Haifa, Israel.

Lee, J. & Seneff, S. 2006. Automatic grammar correction for second-language learners. In *Proc. Interspeech.* Pittsburgh, PA.

Lee, J. & Seneff, S. 2007. Automatic generation of cloze items for prepositions. In *Proc. Interspeech.* Antwerp, Belgium.

Lee, J. & Seneff, S. 2008a. An Analysis of Grammatical Errors in Non-native Speech in English. In *Proc. IEEE Workshop on Spoken Language Technology.* Goa, India.

Lee, J. & Seneff, S. 2008b. Correcting misuse of verb forms. In *Proc. ACL.* Columbus, OH.

Lee, J., Zhou, M. & Liu, X. 2007. Detection of non-native sentences using machine-translated training data. In *Proc. HLT-NAACL (short paper).* Rochester, NY.

Litman, D. & Silliman, S. 2004. ITSPOKE: An Intelligent Tutoring Spoken Dialogue System. In *Proc. HLT/NAACL.* Boston, MA.

Liu, C.-H., Wu, C.-H. & Harris, M. 2008. Word Order Correction for Language Transfer using Relative Position Language Modeling. In *Proc. International Symposium on Chinese Spoken Language Processing*. Kunming, China.

Liu, C.-L., Wang, C.-H., Gao, Z.-M. & Huang, S.-M. 2005. Applications of lexical information for algorithmically composing multiple-choice cloze items. In *Proc. 2nd Workshop on Building Educational Applications using NLP*. Ann Arbor, MI.

Liu, D. & Gleason, J. 2002. Acquisition of the article the by nonnative speakers of english. *Studies in Second Language Acquisition*, *24*, 1–26.

Lu, Y. & Zhou, M. 2004. Collocation Translation Acquisition Using Monolingual Corpora. In *Proc. ACL*. Barcelona, Spain.

Marcus, M., Kim, G., Marcinkiewicz, A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K. & Schasberger, B. 1994. The penn treebank: Annotating predicate argument structure. In *Proc. ARPA Workshop on Human Language Technology*. Plainsboro, NJ.

McCoy, K. F., Pennington, C. & Suri, L. Z. 1996. English Error Correction: A Syntactic User Model Based on Principled 'Mal-Rule' Scoring. In *Proc. 5th International Conference on User Modeling*. Kailua-Kona, HI.

McGraw, I. & Seneff, S. 2008. Speech-enabled Card Games for Language Learners. In *Proc. AAAI*. Chicago, IL.

McKeown, K., Barzilay, R., Evans, D., Hatzivassiloglou, V., Schiffman, B. & Teufel, S. 2001. Columbia Multi-Document Summarization: Approach and Evaluation. In *Proc. SIGIR*.

Merlo, P. & Esteve Ferrer, E. 2006. The notion of argument in prepositional phrase attachment. *Computational Linguistics*, *32(3)*, 341–378.

Michaud, L., McCoy, K. F. & Pennington, C. A. 2000. An Intelligent Tutoring System for Deaf Learners of Written English. In *Proc. 4th International ACM SIGCAPH Conference on Assistive Technologies (ASSETS)*. Arlington, VA.

Michaud, L. N. & McCoy, K. F. 2001. Error Profiling: Toward a Model of English Acquisition for Deaf Learners. In *Proc. ACL* (pp. 386–393). Toulouse, France.

Minnen, G., Bond, F. & Copestake, A. 2000. Memory-based learning for article generation. In *Proceedings of the 4th Conference on Computational Language Learning and the 2nd Learning Language in Logic Workshop (CoNLL/LLL-2000)* (pp. 43–48). Lisbon, Portugal.

Mitkov, R. & Ha, L. A. 2003. Computer-aided generation of multiple-choice tests. In *Proc. HLT-NAACL Workshop on Building Educational Applications using NLP*. Edmonton, Canada.

Mitkov, R., Ha, L. A. & Karamanis, N. 2006. A Computer-aided Environment for Generating Multiple-choice Test Items. *Natural Language Engineering*, *1(1)*, 1–17.

Nagata, R., Kawai, A., Morihiro, K. & Isu, N. 2006. A feedback-augmented method for detecting errors in the writing of learners of english. In *Proc. ACL*. Sydney, Australia.

Ng, V. & Cardie, C. 2002. Improving Machine Learning Approaches to Coreference Resolution. In *Proc. ACL*. Philadelphia, PA.

Park, J. C., Palmer, M. & Washburn, G. 1997. An english grammar checker as a writing aid for students of english as a second language. In *Proc. ANLP*.

Quirk, C., Menezes, A. & Cherry, C. 2005. Dependency Tree Translation: Syntactically Informed Phrasal SMT. In *Proc. ACL*. Ann Arbor, MI.

Quirk, R., Greenbaum, S., Leech, G. & Svartvik, J. 1985. *A Comprehensive Grammar of the English Language*. New York, NY: Longman.

Ratnaparkhi, A. 1996. A Maximum Entropy Part-of-Speech Tagger. In *Proc. EMNLP*. Philadelphia, PA.

Ratnaparkhi, A. 2000. Trainable Methods for Surface Natural Language Generation. In *Proc. NAACL*. Seattle, WA.

Ratnaparkhi, A., Reynar, J. & Roukos, S. 1994. A maximum entropy model for prepositional phrase attachment. In *Proc. ARPA Workshop on Human Language Technology*. Plainsboro, NJ.

Reynar, J. C. & Ratnaparkhi, A. 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries. In *Proc. 5th Conference on Applied Natural Language Processing*. Washington, DC.

Ronowicz, E. & Yallop, C. 2005. *English: One Language, Different Cultures*. Continuum International Publishing Group.

Seneff, S. 1992a. Robust Parsing for Spoken Language Systems. In *Proc. ICASSP*. San Francisco, CA.

Seneff, S. 1992b. TINA: A Natural Language System for Spoken Language Applications. *Computational Linguistics, 18(1)*, 61–86.

Seneff, S. 2002. Response Planning and Generation in the MERCURY Flight Reservation System. *Computer Speech and Language*, *16*, 283–312.

Seneff, S. 2006. Interactive computer aids for acquiring proficiency in mandarin. In *Proc. ISCSLP*.

Seneff, S., Wang, C. & Lee, J. 2006. Combining Linguistic and Statistical Methods for Bi-directional English-Chinese Translation in the Flight Domain. In *Proc. AMTA*. Cambridge, MA.

Seneff, S., Wang, C., Peabody, M. & Zue, V. 2004. Second Language Acquisition through Human Computer Dialogue. In *Proc. ISCSLP*. Hong Kong, China.

Shei, C.-C. 2001. Followyou!: An automatic language lesson generation system. *Computer Assisted Language Learning, 14(2)*, 129–144.

Sjöbergh, J. & Knutsson, O. 2005. Faking Errors to Avoid Making Errors: Very Weakly Supervised Learning for Error Detection in Writing. In *Proc. RANLP*.

Stolcke, A. 2002. Srilm — an extensible language modeling toolkit. In *Proc. ICSLP*. Denver, CO.

Sukkarieh, J. & Bolge, E. 2008. Leveraging C-Raters Automated Scoring Capability for Providing Instructional Feedback for Short Constructed Responses. In *Proc. ITS*.

Sumita, E., Sugaya, F. & Yamamoto, S. 2005. Measuring non-native speakers' proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proc. 2nd Workshop on Building Educational Applications using NLP*. Ann Arbor, MI.

Sun, G., Liu, X., Cong, G., Zhou, M., Xiong, Z., Lee, J. & Lin, C.-Y. 2007. Detecting Erroneous Sentences using Automatically Mined Sequential Patterns. In *Proc. ACL* (pp. 81–88). Prague, Czech Republic.

Suzuki, H. & Toutanova, K. 2006. Learning to predict case markers in japanese. In *Proc. ACL*.

Takezawa, T., Sumita, E., Sugaya, F., Yamamoto, H. & Yamamoto, S. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversation in the real world. In *Proc. LREC*. Las Palmas, Spain.

Tetreault, J. & Chodorow, M. 2008a. Native Judgments of Non-Native Usage. In *Proc. COLING Workshop on Human Judgements in Computational Linguistics*. Manchester, UK.

Tetreault, J. & Chodorow, M. 2008b. The Ups and Downs of Preposition Error Detection in ESL Writing. In *Proc. COLING* (pp. 865–872). Manchester, UK.

Thurmair, G. 1990. Parsing for Grammar and Style Checking. In *Proc. COLING*. Helsinki, Finland.

Tomokiyo, L. M. & Jones, R. 2001. You're not from 'round here, are you? naïve bayes detection of non-native utterance text. In *Proc. NAACL*. Pittsburgh, PA.

Tsubota, Y., Kawahara, T. & Dantsuji, M. 2004. Practical use of english pronunciation system for japanese students in the call classroom. In *Proc. Interspeech*.

Uchimoto, K., Sekine, S. & Isahara, H. 2002. Text Generation from Keywords. In *Proc. COLING* (pp. 1037–1043).

Vogel, C. & Cooper, R. 1995. Robust Chart Parsing with Mildly Inconsistent Feature Structures. *Nonclassical Feature Systems, 10*.

Wang, C. & Seneff, S. 2004. High-Quality Speech Translation for Language Learning. In *Proc. InSTIL*. Venice, Italy.

Wang, C. & Seneff, S. 2006. High-Quality Speech Translation in the Flight Domain. In *Proc. Interspeech*. Pittsburgh, PA.

Wang, Y. & Garigliano, R. 1992. An Intelligent Language Tutoring System for Handling Errors Caused by Transfer. In *Proc. 2nd International Conference on Intelligent Tutoring Systems* (pp. 395–404).

Wible, D., Kuo, C.-H., Chien, F.-Y., Liu, A. & Tsao, N.-L. 2001. A Web-based EFL Writing Environment: Integrating Information for Learners, Teachers and Researchers. *Computers and Education*, *37*, 297–315.

Xu, Y. & Seneff, S. 2008. Two-Stage Translation: A Combined Linguistic and Statistical Machine Translation Framework. In *Proc. AMTA*. Waikiki, Hawaii.

Yi, X., Gao, J. & Dolan, W. B. 2008. A web-based english proofing system for english as a second language users. In *Proc. IJCNLP*.