Extracting Formulaic Expressions and Grammar and Edit Patterns to Assist Academic Writing

Jhih-Jie Chen¹, Jim Chang¹, Ching-Yu Yang¹, Mei-Hua Chen², Jason S. Chang¹
¹Department of Computer Science
National Tsing Hua University
²Department of Foreign Languages and Literature
Tunghai University
{jjc, jim, chingyu, jason}@nlplab.cc,
chen.meihua@gmail.com

Abstract

We present a method for extracting formulaic expressions, grammar patterns, and editing rules from a given corpus to assist learners in learning to write at the level required in English for Academic Purposes. In our method, sentences in a given corpus are parsed into chunks of base phrases, with the arguments sense disambiguated to derive syntactic and semantic grammar patterns. The method involves executing shallow parsing, transforming phrases into grammar patterns, and filtering and ranking grammar patterns for each headword. We applied the proposed method to a corpus annotated with writing errors and their corrections to derive editing rules. Experiments based on a large-scale academic English corpus and WikEd Error Corpus showed that the proposed method produces reasonable correct grammar patterns as well as edit rules. Thus, the method has the potential to assist learners in writing and self-editing.

1 Introduction

Approximately 1 billion people are learning and using English worldwide [10], mostly as a second language (L2). Specifically [3], there are approximately 375 million native speakers of English, and 750 million people use English as a second language.

This large number of L2 learners has motivated the research and development of computer-assisted language learning (CALL) systems. Cambridge English Write & Improve (writeandimprove.com) helps learners by assessing a language level and providing corrective feedback for a submitted essay. Pigaiwang
(www.pigai.org) assigns a score along with suggested edits based on collocation information. However, the suggestions provided often lack information such as the grammatical structures or collocations related to the errors. The aforementioned systems simply deal with basic edits (replace, insert, and delete) and lack comprehensive information that currently only a human editor can provide. Thus, the corrective feedback is either limited or overwhelming, in that it provides little help for learners to improve their writing.

Specifically, the consideration of taking grammar and collocation into account stems from the analysis of the common sources of learners’ writing errors in the writing section of the Macmillan English Dictionary 2nd Edition [4]. This analysis attributes the common errors to the use of the incorrect complementation patterns after a verb, noun, or adjective. The reason for focusing on verbs, nouns, and adjectives is that they exhibit regularity in local syntax [11]. By contrast, the Longman Dictionary of Common Errors [19] attributes preposition errors and verb form errors to the headword preceding the error (e.g., have difficulty breathing instead of have difficulty to breath).

Quality feedback should provide information such as that presented in the recently published Oxford Learners’ Dictionary of Academic English, which describes how people use feedback. A local grammar is limited in scope, with a group of forms for each verb, noun, and adjective. Each rule consists of a handful of elements including syntactic symbols (n, v, adj, adv, clause), semantic categories (amount, color, ord), and function words (e.g., “wh-” words, prepositions such as in, on, at). Moreover, Patterns Dictionary of English Verbs (PDEV) website takes one step further toward local grammar (e.g., abandon: V n) with coarse-grained semantic labels (e.g., abandon ACTIVITY or PLAN; PERSON; CONCEPT; LOCATION; ARTIFACT).

Although parsing complete sentences accurately remains a difficult problem, identifying local grammar patterns of verbs, nouns, and adjectives is relatively simple, yet pedagogically useful for learners. All these aforementioned elements can be easily and accurately identified by a shallow parser (e.g., GENIA Tagger). By shallow parsing sentences in the given corpus, converting phrases to the linguistic form described in [12], and discarding less frequent patterns, we can induce high-quality grammar patterns.

Subsequently, we can disambiguate the arguments (e.g., car, house, lawyer, attorney) in a grammar pattern (e.g., afford n) and assign each a semantic label (e.g., ARTIFACT and PERSON). Based on the observation noted by PDEV, the classes of argument tend to obey the power law that the most frequent class dominates the less frequent ones. By disambiguating the group of nouns to WordNet [15] senses in order to maximize semantic closeness, we can obtain a reasonably accurate semantic annotation such as the following example:
- e.g., V n : afford a car/house ARTIFACT
- e.g., V n : afford a lawyer/an attorney PERSON

We also apply our method to an error-annotated corpus along with the corrections to derive edit patterns. For example, from the annotated sentence “My father couldn’t afford [-paying-] {+to pay+} for my education.”, we learn the pattern: afford -ing → afford to-inf.

The rest of the paper is organized as follows. First, we review the related work in Section 2. Next, we elaborate our method for extracting grammar patterns, assigning semantic labels to arguments, and deriving edit patterns in Section 3. In Section 4, the performance of the proposed method is evaluated by verifying the accuracy of the derived grammar patterns, semantic labels, and edit patterns over a set of noun, verb, and adjective headwords. Finally, we describe the high potential of the grammar patterns and envision the future of writing.

2 Related Work

Statistical analysis of corpus data has been an active research area for corpus linguistics, computational lexicography, and CALL.

In the area of corpus linguistics, considerable work has been conducted to identify formulaic expressions (i.e., lexical bundles), or “extended collocations” that shape the structure of text (e.g., [1]). A study [11] presented Pattern Grammar, a comprehensive description of the local regularity of English verbs, nouns, and adjectives. Our work addresses the computational aspect of inducing Pattern Grammar from raw web-scale ngrams without the benefit of full sentences, annotation, and human judgment. For example, we automatically derive the pattern “HARMFUL to n” based on the ngrams such as “harmful to my computer/minors/the environment.”

In the area of computational lexicography, a study [16] described the method and implemented XTract for extracting collocations from ngrams. Another study [13] described the lexicographic tool Sketch Engine, which uses a set of hand-written patterns to extract collocation from a part-of-speech (POS)-tagged corpus. More recently, a study [6] proposed a method based on rank ratio for extracting collocations and multiword expressions. In contrast to Sketch Engine, our method automatically extracts grammar patterns from raw ngrams based on an extended notion of rank ratio.

In the area of automatic grammatical error correction, considerable work has been conducted using rule-based and statistical approaches. Many rule-based systems have been developed for detecting and correcting article and preposition errors in non-native texts (e.g., [7]). Recently, researchers have begun to use sta-
tical methods to correct writing errors related to articles and prepositions (e.g., [5], [17]). More recently, a study [9] described and evaluates a proofing system, English as Second Language Assistant, based on ngrams and language models. An evaluation based on an online user log showed that 36% of suggested edits were accepted. Additionally, researchers have proposed methods for handling open-class verb errors ([14]; [20]).

In the area of automated essay scoring, a study [2] described the implementation of Criterion, which consists of a grammar checker, Critique, and an automated essay scoring system, e-rater. This scoring system assesses a score for a student’s essay and agrees with human judges in the evaluation as often as two human judges agree with each other. More recently, a study [8] described the technique used in Write & Improve, which assesses goodness to each sentence of an essay and marks some obvious grammatical errors.

In contrast to previous research in corpus-based CALL, we present a system that automatically induces grammar patterns from a given web-scale corpus. Moreover, the system interacts with the user intensively and guides the writing and editing process with grammatical suggestions. We exploit Pattern Grammar and the inherent regularity of natural language by filtering and transforming fragmented ngrams into a set of comprehensive grammar patterns.

3 Extracting Grammar and Edit Patterns

Grammar patterns compiled for dictionaries provide much help to language learners with continuous writing patterns. However, the process is labor intensive and time consuming; thus, customizing it for any given corpus is difficult. Additionally, these patterns tend to lack semantic labels, and the general idea of usage is difficult for a learner to grasp.

We attempt to induce the patterns from a large-scale reference corpus and an annotated learner corpus to provide concise, top-down information of word usage and misuses in order to assist learners in academic writing.

3.1 Extracting Semantic Grammar Patterns

In the first stage of the extraction process, we parse the sentences, identify possible grammar patterns, and group the grammar patterns by their content words (or headwords). The extent of the patterns described here (e.g., V-ing from afford to pay) consists of several base phrases produced by a shallow parser (e.g., a VP phrase followed by an infinite VP). The result of this stage is grammar patterns and phrase instances tagged with coarse-grained semantic labels.
The extraction process is as follows. First, we shallow parse all sentences and produce the results in the form of a sequence of base phrase chunks (Step 1). Next, for each sentence and for each sub-sequence of phrase chunk up to a certain length (MaxNumberOfPhrases), we convert the phrase sequence into a sequence of elements called grammar pattern candidates (Step 2).

Subsequently, the sequence of elements is filtered by matching it against a list of legitimate grammar pattern templates and paired along with the headword and phrase instance (Step 3).

After all the sentences are processed, for each headword, we compute the instance counts C of all related patterns and the instance mean μ, as well as the standard deviation δ of C (Step 4). With the statistics, we then retain the high-frequency patterns with C > μ + kδ (Step 5). A sample of extracted patterns with phrase instances is presented as follows.

- afford: V to-inf (afford to pay/miss)
- afford: V n (afford a lawyer/attorney/car/house)
- afford V n n (afford them the opportunities/protection)

After we have extracted patterns with their phrase instances, we continue to disambiguate (to WordNet senses) and label the noun arguments n (or complements) in the phrases with a semantic category (e.g., WordNet’s lexical information or supersenses).

Next, for each pattern and each of the nouns found in related phrases, we refer to the WordNet database and find the most frequent senses with their SuperSense tags—ignoring senses with less than a 5% (MiniFreqPercent)—and sum up the occurrence counts of each SuperSense tag S (Step 6).

We then estimate the sense probability of each noun P(S|n, pat) according to the maximum likelihood estimation of P(S|pat) (Step 7) and recalculate the estimate of P(S|pat) by summing P(S|n, pat) for all n (Step 8). We repeat Steps 7 and 8 until P(S) converges and stabilizes. Finally, we disambiguate each noun n to an S that maximizes P(S|n, pat) and produces (pat, ph, S) as the output, where S represents the semantic labels for the argument n in the phrase instance ph. A sample of the extracted patterns with the phrase instances is presented as follows.

- abandon: V ACTIVITY (abandon search/effort)
- abandon: V CONCEPT (abandon plan/project)
- abandon V LOCATION (abandon car/ship)
- abandon V PERSON (abandon child/wife)
3.2 Extracting Edit Patterns

In the second stage of the extraction process, we apply the method described in Section 3.1 to a given corpus annotated with error and correction tags. The output of this stage is a set of grammar pattern pairs (e.g., <discuss about n, discuss n>), representing the process and linguistic explanation of common edits. Our extraction process is as follows. First, we convert all sentences with at least one edit into <SentWrong, SentRight> pairs of erroneous and corrected sentences. Then, for all sentence pairs, we apply the extraction process (Steps 1 to 3 in Section 3.1) to extract the patterns PatWrong from SentWrong, where each instance of PatWrong spans over at least one error position (Step 1).

Subsequently, we execute the same process for SentRight to extract PatRight (Step 2). We then pair up PatWrong with PatRight to produce the pattern pairs, <pat-wrong, pat-right>, where pat-wrong and pat-right are either headed or ended by the same word and preferably of the same length (Step 3).

Finally, we compute and organize the editing pattern pairs with the corresponding counts. A sample of extracted patterns with phrase instances is presented in Table 1.

4 Experiment and Evaluation Results

WriteAhead was designed to provide general and broad-coverage writing suggestions for L2 learners. As such, WriteAhead was trained using a publicly available sample of 1,244 exam scripts written by learners who sat for the Cambridge ESOL First Certificate in English examination in 2000.

In this section, we first present the details of training WriteAhead (Section 4.1). Then, in Section 4.2, we describe the evaluation and results.

4.1 Experimental Setting

We used the CiteseerX corpus (460 million words in 20 million sentences) and British National Corpus (100 million words in 5 million sentences), with the combined corpus comprising approximately 500 million words in 20 million sentences. We used GENIA Tagger [18] to tag the sentences.

4.2 Evaluation and Results

After training WriteAhead, as described in the preceding section, we conducted a preliminary evaluation to assess its performance by using sentences containing one of the most common types of errors in the publicly available Cambridge Learner
Table 1: Sample test sentences and results

<table>
<thead>
<tr>
<th>Test Sentence</th>
<th>Edit Pattern &amp; Relevant Instances</th>
<th>ErrType</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to make better hairstyle?</td>
<td>make hairstyle → design hairstyle</td>
<td>RV</td>
</tr>
<tr>
<td></td>
<td>hairstyle: v N → v N</td>
<td></td>
</tr>
<tr>
<td>She explained him that she could not help.</td>
<td>explain him → explained to him</td>
<td>MT</td>
</tr>
<tr>
<td></td>
<td>explain: V n → V to n</td>
<td></td>
</tr>
<tr>
<td>I have the chance for giving my opinion.</td>
<td>chance for giving → chance to give</td>
<td>UT+VT</td>
</tr>
<tr>
<td></td>
<td>chance: N for -ing → N to v</td>
<td></td>
</tr>
</tbody>
</table>

Corpus of First Certificate in English (CLC-FCE) [21]. After excluding punctuation, verb tense, and word order errors, we had eight types of error: Replace a Verb (RV), Preposition (RT), Noun (RN), or others (R), Missing Determinant (MD), Preposition (MT), Unnecessary Determinant (UD), Preposition (UT).

The set of test sentences was generated by sampling the CLC-FCE for 10 sentences for each of the 8 error types. The extracted patterns and examples for the words triggering the error were examined to determine whether the information was sufficient to make the correction.

For example, for the top writing error of replacing verbs, we selected the annotated sentence “How to [-make-]{+create+} better hairstyle?”. We looked for instances of v HAIRSTYLE and found the pattern instances that included create a hairstyle; we then determined that WriteAhead was successful in this case. Examples of the edit patterns for some of the test sentences are shown in Table 1.

5 Conclusion and Future Work

In summary, we propose a method for providing writing suggestions and editing feedback while a person is typing or hovering their mouse over some words. The method involves extracting, retrieving, and ranking grammar patterns and examples. We implemented and evaluated the proposed method as applied to a scholarly corpus with promising results.

Many avenues exist for future research on and improvement of WriteAhead. For example, natural language processing and machine learning techniques could be used to improve the run-time ranking of writing suggestions or corrective feedback for editing. Furthermore, an appealing direction for exploration is systematically training a classifier to predict the POS or phrase boundary of an ngram entry by using predecessor or successor characteristics extracted from ngrams. Another direction of research would be to translate patterns and instances into the learners’ first language and derive synchronous pattern grammar to support the learners.
References


