Automatic Collocation Suggestion in Academic Writing

Jian-Cheng Wu[1] **Yu-Chia Chang**[1,*] **Teruko Mitamura**[2] **Jason S. Chang**[1]

[1] National Tsing Hua University
Hsinchu, Taiwan
{wujc86, richtrf, jason.jschang}@gmail.com

[2] Carnegie Mellon University
Pittsburgh, United States
teruko@cs.cmu.edu

Abstract

In recent years, collocation has been widely acknowledged as an essential characteristic to distinguish native speakers from non-native speakers. Research on academic writing has also shown that collocations are not only common but serve a particularly important discourse function within the academic community. In our study, we propose a machine learning approach to implementing an online collocation writing assistant. We use a data-driven classifier to provide collocation suggestions to improve word choices, based on the result of classification. The system generates and ranks suggestions to assist learners' collocation usages in their academic writing with satisfactory results.

1 Introduction

The notion of collocation has been widely discussed in the field of language teaching for decades. It has been shown that collocation, a successive common usage of words in a chain, is important in helping language learners achieve native-like fluency. In the field of English for Academic Purpose, more and more researchers are also recognizing this important feature in academic writing. It is often argued that collocation can influence the effectiveness of a piece of writing and the lack of such knowledge might cause cumulative loss of precision (Howarth, 1998).

Many researchers have discussed the function of collocations in the highly conventionalized and specialized writing used within academia. Research also identified noticeable increases in the quantity and quality of collocational usage by native speakers (Howarth, 1998). Granger (1998) reported that learners underuse native-like collocations and overuse atypical word combinations. This disparity in collocation usage between native and non-native speakers is clear and should receive more attention from the language technology community.

To tackle such word usage problems, traditional language technology often employs a database of the learners' common errors that are manually tagged by teachers or specialists (e.g. Shei and Pain, 2000; Liu, 2002). Such system then identifies errors via string or pattern matching and offer only pre-stored suggestions. Compiling the database is time-consuming and not easily maintainable, and the usefulness is limited by the manual collection of pre-stored suggestions. Therefore, it is beneficial if a system can mainly use untagged data from a corpus containing correct language usages rather than the error-tagged data from a learner corpus. A large corpus of correct language usages is more readily available and useful than a small labeled corpus of incorrect language usages.

For this suggestion task, the large corpus not only provides us with a rich set of common collocations but also provides the context within which these collocations appear. Intuitively, we can take account of such context of collocation to generate more suitable suggestions. Contextual information in this sense often entails more linguistic clues to provide suggestions within sentences or paragraph. However, the contextual information is messy and complex and thus has long been overlooked or ignored. To date, most fashionable suggestion methods still rely upon the linguistic components within collocations as well as the linguistic relationship between misused words and their correct counterparts (Chang et al., 2008; Liu, 2009).

In contrast to other research, we employ contextual information to automate suggestions for verb-noun lexical collocation. Verb-noun collocations are recognized as presenting the most

* Corresponding author: Yu-chia Chang (Email address: richtrf@gmail.com)

Proceedings of the ACL 2010 Conference Short Papers, pages 115–119,
Uppsala, Sweden, 11-16 July 2010.

challenge to students (Howarth, 1996; Liu, 2002). More specifically, in this preliminary study we start by focusing on the word choice of verbs in collocations which are considered as the most difficult ones for learners to master (Liu, 2002; Chang, 2008). The experiment confirms that our collocation writing assistant proves the feasibility of using machine learning methods to automatically prompt learners with collocation suggestions in academic writing.

2 Collocation Checking and Suggestion

This study aims to develop a web service, *Collocation Inspector* (shown in Figure 1) that accepts sentences as input and generates the related candidates for learners.

In this paper, we focus on automatically providing academic collocation suggestions when users are writing up their abstracts. After an abstract is submitted, the system extracts linguistic features from the user's text for machine learning model. By using a corpus of published academic texts, we hope to match contextual linguistic clues from users' text to help elicit the most relevant suggestions. We now formally state the problem that we are addressing:

Problem Statement: Given a sentence S written by a learner and a reference corpus RC, our goal is to output a set of most probable suggestion candidates $c_1, c_2, \ldots, c_m$. For this, we train a classifier MC to map the context (represented as feature set $f_1, f_2, \ldots, f_n$) of each sentence in RC to the collocations. At run-time, we predict these collocations for S as suggestions.

2.1 Academic Collocation Checker Training Procedures

Sentence Parsing and Collocation Extraction: We start by collecting a large number of abstracts from the Web to develop a reference corpus for collocation suggestion. And we continue to identify collocations in each sentence for the subsequent processing.

Collocation extraction is an essential step in preprocessing data. We only expect to extract the collocation which comprises components having a syntactic relationship with one another. However, this extraction task can be complicated. Take the following scholarly sentence from the reference corpus as an example (example (1)):

```
(1) We introduce a novel method
for learning to find documents
on the web.
```

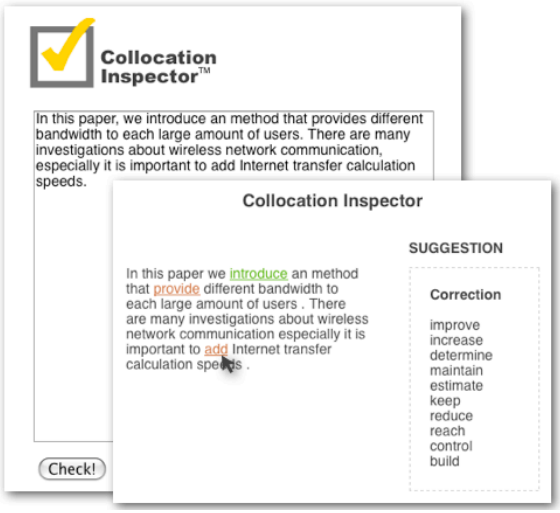

Figure 1. The interface for the collocation suggestion

```
nsubj (introduce-2, We-1)
det (method-5, a-3)
amod (method-5, novel-4)
dobj (introduce-2, method-5)
prepc_for (introduce-2, learning-7)
aux (find-9, to-8)
… …
```
Figure 2. Dependency parsing of Example (1)

Traditionally, through part-of-speech tagging, we can obtain a tagged sentence as follows (example (2)). We can observe that the desired collocation "introduce method", conforming to "VERB+NOUN" relationship, exists within the sentence. However, the distance between these two words is often flexible, not necessarily rigid. Heuristically writing patterns to extract such verb and noun might not be effective. The patterns between them can be tremendously varied. In addition, some verbs and nouns are adjacent, but they might be intervened by clause and thus have no syntactic relation with one another (e.g. "propose model" in example (3)).

```
(2) We/PRP  introduce/VB  a/DT
novel/JJ  method/NN  for/IN
learning/VBG  to/TO  find/VB
documents/NNS  on/IN  the/DT
web/NN  ./.

(3) We proposed that the web-
based model would be more ef-
fective than corpus-based one.
```

A natural language parser can facilitate the extraction of the target type of collocations. Such parser is a program that works out the grammatical structure of sentences, for instance, by identifying which group of words go together or which

word is the subject or object of a verb. In our study, we take advantage of a dependency parser, *Stanford Parser*, which extracts typed dependencies for certain grammatical relations (shown in Figure 2). Within the parsed sentence of example (1), we can notice that the extracted dependency "dobj (introduce-2, method-4)" meets the criterion.

Using a Classifier for the Suggestion task: A classifier is a function generally to take a set of attributes as an input and to provide a tagged class as an output. The basic way to build a classifier is to derive a regression formula from a set of tagged examples. And this trained classifier can thus make predication and assign a tag to any input data.

The suggestion task in this study will be seen as a classification problem. We treat the collocation extracted from each sentence as the class tag (see examples in Table 1). Hopefully, the system can learn the rules between tagged classes (i.e. collocations) and example sentences (i.e. scholarly sentences) and can predict which collocation is the most appropriate one given attributes extracted from the sentences.

Another advantage of using a classifier to automate suggestion is to provide alternatives with regard to the similar attributes shared by sentences. In Table 1, we can observe that these collocations exhibit a similar discourse function and can thus become interchangeable in these sentences. Therefore, based on the outputs along with the probability from the classifier, we can provide more than one adequate suggestions.

Feature Selection for Machine Learning: In the final stage of training, we build a statistical machine-learning model. For our task, we can use a supervised method to automatically learn the relationship between collocations and example sentences.

We choose Maximum Entropy (ME) as our training algorithm to build a collocation suggestion classifier. One advantage of an ME classifier is that in addition to assigning a classification it can provide the probability of each assignment. The ME framework estimates probabilities based on the principle of making as few assumptions as possible. Such constraints are derived from the training data, expressing relationships between features and outcomes.

Moreover, an effective feature selection can increase the precision of machine learning. In our study, we employ the contextual features which

Table 1. Example sentences and class tags (collocations)

Example Sentence	Class tag
We introduce a novel method for learning to find documents on the web.	introduce
We presented a method of improving Japanese dependency parsing by using large-scale statistical information.	present
In this paper, we will describe a method of identifying the syntactic role of antecedents, which consists of two phases	describe
In this paper, we suggest a method that automatically constructs an NE tagged corpus from the web to be used for learning of NER systems.	suggest

consist of two elements, the *head* and the *ngram* of context words:

Head: Each collocation comprises two parts, collocate and head. For example, in a given verb-noun collocation, the verb is the collocate as well as the target for which we provide suggestions; the noun serves as the head of collocation and convey the essential meaning of the collocation. We use the head as a feature to condition the classifier to generate candidates relevant to a given head.

Ngram: We use the context words around the target collocation by considering the corresponding unigrams and bigrams words within the sentence. Moreover, to ensure the relevance, those context words, before and after the punctuation marks enclosing the collocation in question, will be excluded. We use the parsed sentence from previous step (example (2)) to show the extracted context features[1] (example (4)):

```
(4) CN=method UniV_L=we
UniV_R=a UniV_R=novel UniN_L=a
UniN_L=novel UniN_R=for
UniN_R=learn BiV_R=a_novel
BiN_L=a_novel BiN_R=for_learn
BiV_I=we_a BiN_I=novel_for
```

[1] CN refers to the head within collocation. Uni and Bi indicate the unigram and bigram context words of window size two respectively. V and N differentiate the contexts related to verb or noun. The ending alphabets L, R, I show the position of the words in context, L = left, R = right, and I = in between.

2.2 Automatic Collocation Suggestion at Run-time

After the ME classifier is automatically trained, the model is used to find out the best collocation suggestion. Figure 3 shows the algorithm of producing suggestions for a given sentence. The input is a learner's sentence in an abstract, along with an ME model trained from the reference corpus.

In Step (1) of the algorithm, we parse the sentence for data preprocessing. Based on the parser output, we extract the collocation from a given sentence as well as generate features sets in Step (2) and (3). After that in Step (4), with the trained machine-learning model, we obtain a set of likely collocates with probability as predicted by the ME model. In Step (5), SuggestionFilter singles out the valid collocation and returns the best collocation suggestion as output in Step (6). For example, if a learner inputs the sentence like Example (5), the features and output candidates are shown in Table 2.

(5) There are many investigations about wireless network communication, especially it is important to add Internet transfer calculation speeds.

Procedure CollocationSuggestion(*sent*, *MEmodel*)
(1) *parsedSen* = Parsing(*sent*)
(2) *extractedColl* = CollocationExtraction(*parsedSent*)
(3) *features* = AssignFeature(*ParsedSent*)
(4) *probCollection* = MEprob(*features*, *MEmodel*)
(5) *candidate* = SuggestionFilter(*probCollection*)
(6) Return candidate

Figure 3. Collocation Suggestion at Run-time

Table 2. An example from learner's sentence

Extracted Collocation	Features	Ranked Candidates
add speed	CN=speed UniV_L=important UniV_L=to UniV_R=internet UniV_R=transfer UniN_L=transfer UniN_L=calculation BiV_L=important_to BiV_R=internet_transfer BiN_L=transfer_calculation BiV_I=to_intenet	improve increase determine maintain … …

Table 3. MRR for different feature sets

Feature Sets Included In Classifier	MRR
Features of HEAD	0.407
Features of CONTEXT	0.469
Features of HEAD+CONTEXT	0.518

3 Experiment

From an online research database, *CiteSeer*, we have collected a corpus of 20,306 unique abstracts, which contained 95,650 sentences. To train a Maximum Entropy classifier, 46,255 collocations are extracted and 790 verbal collocates are identified as tagged classes for collocation suggestions. We tested the classifier on scholarly sentences in place of authentic student writings which were not available at the time of this pilot study. We extracted 364 collocations among 600 randomly selected sentences as the held out test data not overlapping with the training set. To automate the evaluation, we blank out the verb collocates within these sentences and treat these verbs directly as the only correct suggestions in question, although two or more suggestions may be interchangeable or at least appropriate. In this sense, our evaluation is an underestimate of the performance of the proposed method.

While evaluating the quality of the suggestions provided by our system, we used the mean reciprocal rank (MRR) of the first relevant suggestions returned so as to assess whether the suggestion list contains an answer and how far up the answer is in the list as a quality metric of the system output. Table 3 shows that the best MRR of our prototype system is 0.518. The results indicate that on average users could easily find answers (exactly reproduction of the blanked out collocates) in the first two to three ranking of suggestions. It is very likely that we get a much higher MMR value if we would go through the lists and evaluate each suggestion by hand. Moreover, in Table 3, we can further notice that contextual features are quite informative in comparison with the baseline feature set containing merely the feature of HEAD. Also the integrated feature set of HEAD and CONTEXT together achieves a more satisfactory suggestion result.

4 Conclusion

Many avenues exist for future research that are important for improving the proposed method. For example, we need to carry out the experiment on authentic learners' texts. We will conduct a user study to investigate whether our system would improve a learner's writing in a real setting. Additionally, adding classifier features based on the translation of misused words in learners' text could be beneficial (Chang et al.,

2008). The translation can help to resolve prevalent collocation misuses influenced by a learner's native language. Yet another direction of this research is to investigate if our methodology is applicable to other types of collocations, such as AN and PN in addition to VN dealt with in this paper.

In summary, we have presented an unsupervised method for suggesting collocations based on a corpus of abstracts collected from the Web. The method involves selecting features from the reference corpus of the scholarly texts. Then a classifier is automatically trained to determine the most probable collocation candidates with regard to the given context. The preliminary results show that it is beneficial to use classifiers for identifying and ranking collocation suggestions based on the context features.

Reference

Y. Chang, J. Chang, H. Chen, and H. Liou. 2008. An automatic collocation writing assistant for Taiwanese EFL learners: A case of corpus-based NLP technology. *Computer Assisted Language Learning*, 21(3), pages 283-299.

S. Granger. 1998. Prefabricated patterns in advanced EFL writing: collocations and formulae. In Cowie, A. (ed.) *Phraseology: theory, analysis and applications*. Oxford University Press, Oxford, pages 145-160.

P. Howarth. 1996. *Phraseology in English Academic Writing.* Tübingen: Max Niemeyer Verlag.

P. Howarth. 1998. The phraseology of learner's academic writing. In Cowie, A. (ed.) *Phraseology: theory, analysis and applications*. Oxford University Press, Oxford, pages 161-186.

D. Hawking and N. Craswell. 2002. Overview of the TREC-2001 Web track. In *Proceedings of the 10th Text Retrieval Conference (TREC 2001)*, pages 25-31.

L. E. Liu. 2002. A corpus-based lexical semantic investigation of verb-noun miscollocations in Taiwan learners' English. *Unpublished master's thesis*, Tamkang University, Taipei, January.

A. L. Liu, D. Wible, and N. L. Tsao. 2009. Automated suggestions for miscollocations. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 47-50.

C. C. Shei and H. Pain. 2000. An ESL writer's collocational aid. *Computer Assisted Language Learning,* 13, pages 167-182.