Abstract

This paper describes our submission to the CoNLL 2014 shared task on grammatical error correction using a hybrid approach, which includes both a rule-based and an SMT system augmented by a large web-based language model. Furthermore, we demonstrate that correction type estimation can be used to remove unnecessary corrections, improving precision without harming recall. Our best hybrid system achieves state-of-the-art results, ranking first on the original test set and second on the test set with alternative annotations.

1 Introduction

Grammatical error correction has attracted considerable interest in the last few years, especially through a series of ‘shared tasks’. These efforts have helped to provide a common ground for evaluating and comparing systems while encouraging research in the field. These shared tasks have primarily focused on English as a second or foreign language and addressed different error types. The HOO 2011 task (Dale and Kilgarriff, 2011), for example, included all error types whereas HOO 2012 (Dale et al., 2012) and the CoNLL 2013 shared task (Ng et al., 2013) were restricted to only two and five types respectively.

In this paper, we describe our submission to the CoNLL 2014 shared task (Ng et al., 2014), which involves correcting all the errors in essays written in English by students at the National University of Singapore. An all-type task poses a greater challenge, since correcting open-class types (such as spelling or collocation errors) requires different correction strategies than those in closed classes (such as determiners or prepositions).

In this scenario, hybrid systems or combinations of correction modules seem more appropriate and typically produce good results. In fact, most of the participating teams in previous shared tasks have used a combination of modules or systems for their submissions, even for correcting closed-class types (Dahlmeier et al., 2011; Bhaskar et al., 2011; Rozovskaya et al., 2011; Ivanova et al., 2011; Rozovskaya et al., 2013; Yoshimoto et al., 2013; Xing et al., 2013; Putra and Szabo, 2013; Xiang et al., 2013).

In line with previous research, we present a hybrid approach that employs a rule-based error correction system and an ad-hoc statistical machine translation (SMT) system, as well as a large-scale language model to rank alternative corrections and an error type filtering technique.

The remainder of this paper is organised as follows: Section 2 describes our approach and each component in detail, Section 3 presents our experiments using the CoNLL 2014 shared task development set and Section 4 reports our official results on the test set. Finally, we discuss the performance of our system and present an error analysis in Section 5 and conclude in Section 6.

2 Approach

We tackle the error correction task using a pipeline of processes that combines results from multiple systems. Figure 1 shows the interaction of the components in our final hybrid system, producing the results submitted to the CoNLL 2014 shared task. The following sections describe each of these components in detail.

2.1 Rule-based error correction system (RBS)

The rule-based system is a component of the Self-Assessment and Tutoring (SAT) system, a web service developed at the University of Cambridge aimed at helping intermediate learners of English
in their writing tasks\(^1\) (Andersen et al., 2013). The original SAT system provides three main functionalities: 1) \textit{text assessment}, producing an overall score for a piece of text, 2) \textit{sentence evaluation}, producing a sentence-level quality score, and 3) \textit{word-level feedback}, suggesting specific corrections for frequent errors. Since the focus of the shared task is on strict correction (as opposed to detection), we only used the word-level feedback component of the SAT system.

This module uses rules automatically derived from the Cambridge Learner Corpus\(^2\) (CLC) (Nicholls, 2003) that are aimed at detecting errorful unigrams, bigrams and trigrams. In order to ensure high precision, rules are based on n-grams that have been annotated as incorrect at least five times and at least ninety per cent of the times they occur. In addition to these corpus-derived rules, many cases of incorrect but plausible derivational and inflectional morphology are detected by means of rules derived from a machine-readable dictionary. For further details on specific components, we refer the reader to the aforementioned paper.

Given an input text, the rule-based system produces an XML file containing a list of suggested corrections. These corrections can either be applied to the original text or used to generate multiple correction candidates, as described in Section 2.3.

### 2.2 SMT system

We follow a similar approach to the one described by Yuan and Felice (2013) in order to train an SMT system that can ‘translate’ from incorrect into correct English. Our training data comprises a set of different parallel corpora, where the original (incorrect) sentences constitute the source side and corrected versions based on gold standard annotations constitute the target side. These corpora include:

- the NUCLE v3.1 corpus (Dahlmeier et al., 2013), containing around 1,400 essays written in English by students at the National University of Singapore (approx. 1,220,257 tokens in 57,152 sentences),

- phrase alignments involving corrections extracted automatically from the NUCLE corpus (with up to 7 tokens per side), which are used to boost the probability of phrase alignments that involve corrections so as to improve recall,

- the CoNLL 2014 shared task development set, containing 50 essays from the previous year’s test set (approx. 29,207 tokens in 1,382 sentences),

- the First Certificate in English (FCE) corpus (Yannakoudakis et al., 2011), containing 1,244 exam scripts and 2 essays per script (approx. 532,033 tokens in 16,068 sentences),

- a subset of the International English Language Testing System (IELTS) examination dataset extracted from the CLC corpus, containing 2,498 exam scripts and 2 essays per script (approx. 1,361,841 tokens in 64,628 sentences), and

- a set of sentences from the English Vocabulary Profile\(^3\) (EVP), which have been modified to include artificially generated errors (approx. 351,517 tokens in 18,830 sentences). The original correct sentences are a subset of the CLC and come from examinations at different proficiency levels. The artificial error generation method aims at replicating frequent error patterns observed in the NUCLE corpus on error-free sentences, as described by Yuan and Felice (2013).

\(^1\)The latest version of the system, called ‘Write & Improve’, is available at [http://www.cambridgeenglish.org/writeandimprovebeta/](http://www.cambridgeenglish.org/writeandimprovebeta/).

\(^2\)More information at [http://www.cambridge.org/elt/catalogue/subject/custom/item3646603/](http://www.cambridge.org/elt/catalogue/subject/custom/item3646603/).

\(^3\)Sentences were automatically scraped from [http://www.englishprofile.org/index.php?option=com_content&view=article&id=4&Itemid=5](http://www.englishprofile.org/index.php?option=com_content&view=article&id=4&Itemid=5).
Word alignment was carried out using pialign (Neubig et al., 2011), after we found it outperformed GIZA++ (Och and Ney, 2000; Och and Ney, 2003) and Berkeley Aligner (Liang et al., 2006; DeNero and Klein, 2007) in terms of precision and $F_{0.5}$ on the development set. Instead of using heuristics to extract phrases from the word alignments learnt by GIZA++ or Berkeley Aligner, pialign created a phrase table directly from model probabilities.

In addition to the features already defined by pialign, we added character-level Levenshtein distance to each mapping in the phrase table. This was done to allow for the fact that, in error correction, most words translate into themselves and errors are often similar to their correct forms. Equal weights were assigned to these features.

We then built a lexical reordering model using the alignments created by pialign. The maximum phrase length was set to 7, as recommended in the SMT literature (Koehn et al., 2003; Koehn, 2014).

The IRSTLM Toolkit (Federico et al., 2008) was used to build a 4-gram target language model with Kneser–Ney smoothing (Kneser and Ney, 1995) on the correct sentences from the NUCLE, full CLC and EVP corpora.

Decoding was performed with Moses (Koehn et al., 2007), using the default settings and weights. No tuning process was applied. The resulting system was used to produce the 10 best correction candidates for each sentence in the dataset, which were further processed by other modules.

Segmentation, tokenisation and part-of-speech tagging were performed using NLTK (Bird et al., 2009) for consistency with the shared task datasets.

### 2.3 Candidate generation

In order to integrate corrections from multiple systems, we developed a method to generate all the possible corrected versions of a sentence (candidates). Candidates are generated by computing all possible combinations of corrections (irrespective of the system from which they originate), including the original tokens to allow for a ‘no correction’ option. The list of candidates produced for each sentence always includes the original (unmodified) sentence plus any other versions derived from system corrections.

In order for a combination of corrections to generate a valid candidate, all the corrections must be compatible; otherwise, the candidate is discarded. We consider two or more corrections to be compatible if they do not overlap, in an attempt to avoid introducing accidental errors. In addition, if different correction sets produce the same candidate, we only keep one. Figure 2 illustrates the candidate generation process.

![Figure 2: An example showing the candidate generation process.](image)

### Table 1: Performance of language models on the development set after ranking the SMT system’s 10-best candidates per sentence.

| Model                  | CE   | ME   | UE   | P    | R    | $F_{0.5}$ |
|------------------------|------|------|------|------|------|-----------|
| SMT IRSTLM             | 651  | 2766 | 1832 | 0.262 | 0.190 | 0.2438    |
| Microsoft Web N-grams  | 666  | 2751 | 1344 | 0.331 | 0.194 | 0.2907    |

CE: correct edits, ME: missed edits, UE: unnecessary edits, P: precision, R: recall.

The ranking performance of these two models was evaluated on the 10-best hypotheses generated by the SMT system for each sentence in the development set. Table 1 shows the results from the M² Scorer (Dahlmeier and Ng, 2012), the official scorer for the shared task that, unlike previous versions, weights precision twice as much as recall.

Results show that using Microsoft’s Web LM yields better performance, which is unsurprising given the vast amounts of data used to build that...
Table 2: Results of individual systems on the development set.

| System | CE  | ME  | UE  | P    | R    | F_{0.5} |
|--------|-----|-----|-----|------|------|---------|
| RBS    | 95  | 3322| 107 | 0.4703 | 0.0278 | 0.1124  |
| SMT    | 452 | 2965| 690 | 0.3958 | 0.1323 | 0.2830  |

We also note that without normalisation, higher probabilities may be assigned to shorter sentences, which can introduce a bias towards preferring deletions or skipping insertions.

### 2.5 Type filtering

Analysing performance by error type is very valuable for system development and tuning. However, this can only be performed for corrections in the gold standard (either matched or missed). To estimate types for unnecessary corrections, we defined a set of heuristics that analyse differences in word forms and part-of-speech tags between the original phrases and their system corrections, based on common patterns observed in the training data. We had previously used a similar strategy to classify errors in our CoNLL 2013 shared task submission (Yuan and Felice, 2013) but have now included a few improvements and rules for new types. Estimation accuracy is 50.92% on the training set and 67.57% on the development set, which we consider to be acceptable for our purposes given that the final test set is more similar to the development set.

Identifying types for system corrections is not only useful during system development but can also be exploited to filter out and reduce the number of proposed corrections. More specifically, if a system proposes a much higher number of unnecessary corrections than correct suggestions for a specific error type, we can assume the system is actually degrading the quality of the original text, in which case it is preferable to filter out those error types. Such decisions will lower the total number of unnecessary edits, thus improving overall precision. However, they will also harm recall, unless the number of matched corrections for the error type is zero (i.e. unless $P_{\text{type}} = 0$). To avoid this, only corrections for types having zero precision should be removed.

## 3 Experiments and results

We carried out a series of experiments on the development set using different pipelines and combinations of systems in order to find an optimal setting. The following sections describe them in detail.

### 3.1 Individual system performance

Our first set of experiments were aimed at investigating individual system performance on the development set, which is reported in Table 2. Results show that the SMT system has much better performance, which is expected given that it has been trained on texts similar to those in the test set.

### 3.2 Pipelines

Since corrections from the RBS and SMT systems are often complementary, we set out to explore combination schemes that would integrate corrections from both systems. Table 3 shows results for different combinations, where RBS and SMT indicate all corrections from the respective systems, subscript ‘c’ indicates candidates generated from a system’s individual corrections, subscript ‘10-best’ indicates the 10-best list of candidates produced by the SMT system, ‘>’ indicates a pipeline where the output of one system is the input to the other and ‘+’ indicates a combination of candidates from different systems. All these pipelines use the RBS system as the first processing step in order to perform an initial correction, which is extremely beneficial for the SMT system.

Results reveal that the differences between these pipelines are small in terms of $F_{0.5}$, although there are noticeable variations in precision and recall. The best results are achieved when the 10 best hypotheses from the SMT system are ranked with Microsoft’s LM, which confirms our results in Table 1 showing that the SMT LM is outperformed by a larger web-based model.

A simple pipeline using the RBS system first and the SMT system second (#3) yields performance that is better than (or comparable to) pipelines #1, #2 and #4, suggesting that there is no real benefit in using more sophisticated pipelines when only the best hypothesis from the SMT system is used. However, performance is improved when the 10 best SMT hypotheses are considered. The only difference between pipelines #5 and #6 lies in the way corrections from the RBS system...
are handled. In the first case, all corrections are applied at once whereas in the second, the suggested corrections are used to generate candidates that are subsequently ranked by our LM, often discarding some of the suggested corrections.

### 3.3 Filtering

As described in Section 2.5, we can evaluate performance by error type in order to identify and remove unnecessary corrections. In particular, we tried to optimise our best hybrid system (§6) by filtering out types with zero precision. Table 5 shows type-specific performance for this system, where three zero-precision types can be identified: Reordering (a subset of Others that we treat separately), Srun (run-ons/comma splices) and Wa (acronyms). Although reordering was explicitly disabled in our SMT system, a translation table can still include this type of mappings if they are observed in the training data (e.g. ‘you also can’ → ‘you can also’).

In order to remove such undesired corrections, the following procedure was applied: first, individual corrections were extracted by comparing the original and corrected sentences; second, the type of each extracted correction was predicted, subsequently deleting those that matched unwanted types (i.e. reordering, Srun or Wa); finally, the set of remaining corrections was applied to the original text. This method improves precision while preserving recall (see Table 4), although the resulting improvement is not statistically significant (paired t-test, \( p > 0.05 \)).

### 4 Official evaluation results

Our submission to the CoNLL 2014 shared task is the result of our best hybrid system, described in the previous section and summarised in Figure 1. The official test set comprised 50 new essays (approx. 30,144 tokens in 1,312 sentences) written in response to two prompts, one of which was also included in the training data.

Systems were evaluated using the \( M^2 \) Scorer, which uses \( F_{0.5} \) as its overall measure. As in previous years, there were two evaluation rounds. The first one was based on the original gold-standard annotations provided by the shared-task organisers whereas the second was based on a revised version including alternative annotations submitted by the participating teams. Our submitted system achieved the first and second place respectively. The official results of our submission in both evaluation rounds are reported in Table 6.

### 5 Discussion and error analysis

In order to assess how our system performed per error type on the test set, we ran our type estimation script and obtained the results shown in Table 7. Although these results are estimated and therefore not completely accurate,\(^4\) they can still provide valuable insights, at least at a coarse level. The following sections discuss our main findings.

#### 5.1 Type performance

According to Table 7, our system achieves the best performance for types WOadv (adverb/adjective position) and Wtone (tone), but these results are

---

\(^4\)Estimation accuracy was found to be 57.90% on the test set.
We also found that our system was particularly good at enforcing different types of agreement, as demonstrated by the results for SVA (subject–verb agreement), Pref (pronoun reference), Nn (noun number) and Vform (verb form) types, which add up to 22.80% of the errors. The following example shows a successful correction:

He or she has the right not to tell anyone.

This is because his or her relatives have the right to know about this.

Gold standard:
They have the right not to tell anyone.

In other cases, our system seems to do a good job despite gold-standard annotations:

Original sentence:
This is because his or her relatives have the right to know about this.

System hypothesis:
This is because their relatives have the right to know about this.

Gold standard:
This is because his or her relatives have the right to know about this. (unchanged)

The worst performance is observed for Others (including Reordering and Srun), which only account for 1.69% of the errors. We also note that Reordering and Srun errors, which had explicitly been filtered out, still appear in our final results.
which is due to differences in the edit extraction algorithms used by the M² Scorer and our own implementation. According to our estimations, our system has poor performance on the Wci type (the second most frequent), suggesting it is not very successful at correcting idioms and collocations.

Corrections for more complex error types such as Um (unclear meaning), which are beyond the scope of this shared task, are inevitably missed.

5.2 Deletions

We have also observed that many mismatches between our system’s corrections and the gold standard are caused by unnecessary deletions, as in the following example:

**Original sentence:**
I could understand the feeling of the carrier.

**System hypothesis:**
I understand the feeling of the carrier.

**Gold standard:**
I could understand the feeling of the carrier.
(unchanged)

This effect is the result of using 10-best hypotheses from the SMT system together with LM ranking. Hypotheses from an SMT system can include many malformed sentences which are effectively discarded by the embedded target language model and additional heuristics. However, ranking these raw hypotheses with external systems can favour deletions, as language models will generally assign higher probabilities to shorter sentences. A common remedy for this is normalisation but we found it made no difference in our experiments.

In other cases, deletions can be ascribed to differences in the domain of the training and test sets, as observed in this example:

**Original sentence:**
Nowadays, social media are able to disseminate information faster than any other media.

**System hypothesis:**
Nowadays, the media are able to disseminate information faster than any other media.

**Gold standard:**
Nowadays, social media are able to disseminate information faster than any other media.
(unchanged)

5.3 Uncredited corrections

Our analysis also reveals a number of cases where the system introduces changes that are not included in the gold standard but we consider improve the quality of a sentence. For example:

**Original sentence:**
Demon is not easily to be defeated and it is required much of energy and psychological support.

**System hypothesis:**
Demon is not easily defeated and it requires a lot of energy and psychological support.

**Gold standard:**
The demon is not easily defeated and it requires much energy and psychological support.

Adding alternative corrections to the gold standard alleviates this problem, although the list of alternatives will inevitably be incomplete.

There are also a number of cases where the sentences are considered incorrect as part of a longer text but are acceptable when they are evaluated in isolation. Consider the following examples:

**Original sentence:**
The opposite is also true.

**System hypothesis:**
The opposite is true.

**Gold standard:**
The opposite is also true.
(unchanged)

**Original sentence:**
It has erased the boundaries of distance and time.

**System hypothesis:**
It has erased the boundaries of distance and time.
(unchanged)

**Gold standard:**
They have erased the boundaries of distance and time.

In both cases, system hypotheses are perfectly grammatical but they are considered incorrect when analysed in context. Such mismatch is the result of discrepancies between the annotation and evaluation criteria: while the gold standard is annotated taking discourse into account, system cor-
rections are proposed in isolation, completely de-
void of discursive context.

Finally, the inability of the M² Scorer to com-
bine corrections from different annotators (as op-
posed to selecting only one annotator’s corrections
for the whole sentence) can also result in underes-
timations of performance. However, it is clear that
exploring these combinations during evaluation is
a challenging task itself.

6 Conclusions

We have presented a hybrid approach to error cor-
rection that combines a rule-based and an SMT
error correction system. We have explored dif-
f erent combination strategies, including sequential
pipelines, candidate generation and ranking. In
addition, we have demonstrated that error type
estimations can be used to filter out unnecessary
corrections and improve precision without harm-
ing recall.

Results of our best hybrid system on the offi-
cial CoNLL 2014 test set yield $F_{0.5}=0.3733$ for
the original annotations and $F_{0.5}=0.4355$ for alter-
native corrections, placing our system in the first
and second place respectively.

Error analysis reveals that our system is partic-
ularly good at correcting mechanical errors and
agreement but is often penalised for unnecessary
deletions. However, a thorough inspection shows
that the system tends to produce very fluent sen-
tences, even if they do not match gold standard
annotations.

Acknowledgements

We would like to thank Marek Rei for his valuable
feedback and suggestions as well as Cambridge
English Language Assessment, a division of Cam-
bridge Assessment, for supporting this research.

References

Ołeinstein E. Andersen, Helen Yannakoudakis, Fiona
Barker, and Tim Parish. 2013. Developing and test-
ing a self-assessment and tutoring system. In Pro-
cedings of the Eighth Workshop on Innovative Use
of NLP for Building Educational Applications, BEA
2013, pages 32–41, Atlanta, GA, USA, June. Asso-
ciation for Computational Linguistics.

Pinaki Bhaskar, Aniruddha Ghosh, Santanu Pal, and
Sivaji Bandyopadhyay. 2011. May I check the
English of your paper!!! In Proceedings of the
Generation Challenges Session at the 13th Euro-
pean Workshop on Natural Language Generation,
pages 250–253, Nancy, France, September. Associ-
ation for Computational Linguistics.

Steven Bird, Edward Loper, and Ewan Klein.
2009. Natural Language Processing with Python.
O’Reilly Media Inc.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Bet-
ter evaluation for grammatical error correction. In
Proceedings of the 2012 Conference of the North
American Chapter of the Association for Computa-
tional Linguistics: Human Language Technologies,
NAACL 2012, pages 568–572, Montreal, Canada.

Daniel Dahlmeier, Hwee Tou Ng, and Thanh Phu Tran.
2011. NUS at the HOO 2011 Pilot Shared Task. In
Proceedings of the Generation Challenges Session
at the 13th European Workshop on Natural Lan-
guage Generation, pages 257–259, Nancy, France,
September. Association for Computational Linguistics.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu.
2013. Building a Large Annotated Corpus of
Learner English: The NUS Corpus of Learner En-
lish. In Proceedings of the 8th Workshop on Inno-
utive Use of NLP for Building Educational Appli-
cations, BEA 2013, pages 22–31, Atlanta, Georgia,
USA, June.

Robert Dale and Adam Kilgarriff. 2011. Helping
Our Own: The HOO 2011 Pilot Shared Task. In
Proceedings of the Generation Challenges Session
at the 13th European Workshop on Natural Lan-
guage Generation, pages 242–249, Nancy, France,
September. Association for Computational Linguistics.

Robert Dale, Ilya Anisimoff, and George Narroway.
2012. HOO 2012: A Report on the Preposition and
Determiner Error Correction Shared Task. In
Proceedings of the Seventh Workshop on Building
Educational Applications Using NLP, pages 54–62,
Montréal, Canada, June. Association for Compu-
tational Linguistics.

John DeNero and Dan Klein. 2007. Tailoring word
alignments to syntactic machine translation. In Pro-
cedings of the 45th Annual Meeting of the Asso-
ciation of Computational Linguistics, pages 17–24,
Prague, Czech Republic, June. Association for Compu-
tational Linguistics.

Marcello Federico, Nicola Bertoldi, and Mauro Cett-
tolo. 2008. IRSTLM: an open source toolkit for
handling large scale language models. In Proceed-
ings of the 9th Annual Conference of the Interna-
tional Speech Communication Association, INTER-
SPEECH 2008, pages 1618–1621, Brisbane, Aus-
tralia, September. ISCA.

Jianfeng Gao, Patrick Nguyen, Xiaolong Li, Chris
Thrasher, Mu Li, and Kuansan Wang. 2010. A
Comparative Study of Bing Web N-gram Language
Models for Web Search and Natural Language Pro-
cessing. In Web N-gram Workshop, Workshop of the
Elitza Ivanova, Delphine Bernhard, and Cyril Grouin. 2011. Handling Outlandish Occurrences: Using Rules and Lexicons for Correcting NLP Articles. In Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation, pages 254–256, Nancy, France, September. Association for Computational Linguistics.

Reinhard Kneser and Hermann Ney. 1995. Improved backoff for m-gram language modeling. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume I, pages 181–184, Detroit, Michigan, May.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, volume 1 of NAACL ’03, pages 48–54, Edmonton, Canada. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL ’07, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn. 2014. Moses: Statistical Machine Translation System – User Manual and Code Guide. University of Edinburgh, April. Available online at http://www.statmt.org/moses/manual/manual.pdf.

Anoop Kunchukuttan, Ritesh Shah, and Pushpak Bhattacharyya. 2013. IITB System for CoNLL 2013 Shared Task: A Hybrid Approach to Grammatical Error Correction. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 82–87, Sofia, Bulgaria, August. Association for Computational Linguistics.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, pages 104–111, New York City, USA, June. Association for Computational Linguistics.

Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. An unsupervised model for joint phrase alignment and extraction. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 632–641, Portland, Oregon, USA, June. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetruault. 2013. The CoNLL-2013 Shared Task on Grammatical Error Correction. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 1–12, Sofia, Bulgaria, August. Association for Computational Linguistics.

Diane Nicholls. 2003. The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT. In Dawn Archer, Paul Rayson, Andrew Wilson, and Tony McEnery, editors, Proceedings of the Corpus Linguistics 2003 conference, pages 572–581, Lancaster, UK. University Centre for Computer Corpus Research on Language, Lancaster University.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ACL ’00, pages 440–447, Hong Kong, October. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. Comput. Linguist., 29(1):19–51, March.

Desmond Darma Putra and Lili Szabo. 2013. UdS at CoNLL 2013 Shared Task. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 88–95, Sofia, Bulgaria, August. Association for Computational Linguistics.

Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, and Dan Roth. 2011. University of Illinois System in HOO Text Correction Shared Task. In Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation, pages 263–266, Nancy, France, September. Association for Computational Linguistics.

Alla Rozovskaya, Mark Sammons, Joshua Gioja, and Dan Roth. 2013. The University of Illinois System in the CoNLL-2013 Shared Task. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 13–19, Sofia, Bulgaria, August. Association for Computational Linguistics.
Yang Xiang, Bo Yuan, Yaoyun Zhang, Xiaolong Wang, Wen Zheng, and Chongqiang Wei. 2013. A hybrid model for grammatical error correction. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 115–122, Sofia, Bulgaria, August. Association for Computational Linguistics.

Junwen Xing, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Xiaodong Zeng. 2013. UM-Checker: A Hybrid System for English Grammatical Error Correction. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 34–42, Sofia, Bulgaria, August. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 180–189, Portland, Oregon, USA, June. Association for Computational Linguistics.

Ippei Yoshimoto, Tomoya Kose, Kensuke Mitsuzawa, Keisuke Sakaguchi, Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, and Yuji Matsumoto. 2013. NAIST at 2013 CoNLL Grammatical Error Correction Shared Task. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 26–33, Sofia, Bulgaria, August. Association for Computational Linguistics.

Zheng Yuan and Mariano Felice. 2013. Constrained grammatical error correction using statistical machine translation. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 52–61, Sofia, Bulgaria, August. Association for Computational Linguistics.