NAIST at 2013 CoNLL Grammatical Error Correction Shared Task

Ippei Yoshimoto, Tomoya Kose, Kensuke Mitsuzawa, Keisuke Sakaguchi,
Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Yuji Matsumoto
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
{ippei-y,tomoya-kos,kensuke-mi,keisuke-sa,tomoya-m,yuta-h,komachi,matsu}@is.naist.jp

Abstract

This paper describes the Nara Institute of Science and Technology (NAIST) error correction system in the CoNLL 2013 Shared Task. We constructed three systems: a system based on the Treelet Language Model for verb form and subject-verb agreement errors; a classifier trained on both learner and native corpora for noun number errors; a statistical machine translation (SMT)-based model for preposition and determiner errors. As for subject-verb agreement errors, we show that the Treelet Language Model-based approach can correct errors in which the target verb is distant from its subject. Our system ranked fourth on the official run.

1 Introduction

Grammatical error correction is the task of automatically detecting and correcting grammatical errors in text, especially text written by second language learners. Its purpose is to assist learners in writing and helps them learn languages.

Last year, HOO 2012 (Dale et al., 2012) was held as a shared task on grammatical error correction, focusing on prepositions and determiners. The CoNLL-2013 shared task (Dahlmeier et al., 2013) includes these areas and also noun number, verb form, and subject-verb agreement errors.

We divide the above 5 error types into three groups: (1) subject-verb agreement (*SVA*) and verb form (*Vform*) errors, (2) noun number (*Nn*) errors, and (3) preposition (*Prep*) and determiner (*ArtOrDet*) errors. For the subject-verb agreement and verb form errors, we used a syntactic language model, the Treelet Language Model, because syntactic information is important for verb error correction. For the noun number errors, we used a binary classifier trained on both learner and native corpora. For the preposition and determiner errors, we adopt a statistical machine translation (SMT)-based approach, aiming at correcting errors in conventional expressions. After each subsystem corrects the errors of the corresponding error types, we merge the outputs of all the subsystems.

The result shows our system achieved 21.85 in F-score on the formal run before revision and 28.14 after revision.

The rest of this paper is organized as follows. Section 2 presents an overview of related work. Section 3 describes the system architecture of each of the three subsystems. Section 4 shows experimental settings and results. Section 5 presents discussion. Section 6 concludes this paper.

2 Related Work

Lee and Seneff (2008) tried correcting English verb errors including *SVA* and *Vform*. They proposed correction candidates with template matching on parse trees and filtered candidates by utilizing *n*-gram counts. Our system suggests candidates based on the Part-Of-Speech (POS) tag of a target word and filters them by using a syntactic language model.

For the noun number errors, we improved the system proposed by Izumi et al. (2003). In Izumi et al. (2003), a noun number error detection method is a part of an automatic error detection system for transcribed spoken English by Japanese learners. They used a maximum entropy method whose features are unigrams, bigrams and trigrams of surface words, of POS tags and of the root forms. They trained a classifier on only a learner corpus. The main difference between theirs and ours is a domain of the training corpus and features we used. We trained a classifier on the mixed corpus of the leaner corpus and the native corpus. We employ a treepath feature in our system.

Our SMT system for correcting preposition and

Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 26–33,
Sofia, Bulgaria, August 8-9 2013. ©2013 Association for Computational Linguistics

determiner errors is based on Mizumoto et al. (2012). They constructed a translation model from the data of the language-exchange social network service Lang-8[1] and evaluated its performance for 18 error types, including preposition and determiner errors in the Konan-JIEM Learner Corpus. On preposition error correction, they showed that their SMT system outperformed a system using a maximum entropy model. The main difference with this work is that our new corpus collection here is about three times larger.

3 System Architecture

3.1 Subject-Verb Agreement and Verb Form

For *SVA* and *Vform* errors, we used the Treelet Language Model (Pauls and Klein, 2012) to capture syntactic information and lexical information simultaneously. We will first show examples of *SVA* and *Vform* errors and then describe our model used to correct them. Finally, we explain the procedure for error correction.

3.1.1 Errors

According to Lee and Seneff (2008), both *SVA* and *Vform* errors are classified as syntactic errors. Examples are as follows:

Subject-Verb Agreement (SVA) The verb is not correctly inflected in number and person with respect to its subject.

*They ***has** been to Nara many times.*

In this example, a verb "*has*" is wrongly inflected. It should be "*have*" because its subject is the pronoun "*they*".

Verb Form (Vform) This type of error mainly consists of two subtypes,[2] one of which includes auxiliary agreement errors.

*They have ***be** to Nara many times.*

Since the "*have*" in this sentence is an auxiliary verb, the "*be*" is incorrectly inflected and it should be "*been*".

The other subtype includes complementation errors like the following:

*They want ***go** to Nara this summer.*

Verbs can be a complement of another verb and preposition. The "*go*" in the above sentence is incorrect. It should be in the infinitive form, "*to go*".

3.1.2 Treelet Language Model

We used the Treelet Language Model (Pauls and Klein, 2012) for *SVA* and *Vform* error correction.

Our model assigns probability to a production rule of the form $r = P \rightarrow C_1 \cdots C_d$ in a constituent tree T, conditioned on a context h consisting of previously generated treelets,[3] where P is the parent symbol of a rule r and $C_1^d = C_1 \cdots C_d$ are its children.

$$p(r) = p(C_1^d|h)$$

The probability of a constituent tree T is given by the following equation:

$$p(T) = \prod_{r \in T} p(r)$$

The context h differs depending on whether C_1^d is a terminal symbol or a sequence of non-terminal symbols.

Terminal When C_1^d is a terminal symbol w,

$$p(C_1^d|h) = p(w|P, R, r', w_{-1}, w_{-2})$$

where P is the POS tag of w, R is the right sibling of P, r' is the production rule which yields P and its siblings, and w_{-2} and w_{-1} are the two words preceding w.

Non-Terminal When C_1^d is a sequence of non-terminal symbols,

$$p(C_1^d|h) = p(C_1^d|P, P', r')$$

where P is the parent symbol of C_1^d, P' is the parent symbol of P.

In order to capture a richer context, we apply the annotation and transformation rules below to parse trees in order. We use almost the same annotation and transformation rules as those proposed by

[1] `http://lang-8.com`

[2] In the NUCLE (Dahlmeier et al., 2013) corpus, most of semantic errors related to verbs are included in other error types such as verb tense errors, not *Vform* errors.

[3] The term *treelet* is used to refer to an arbitrary connected subgraph of a tree (Quirk et al., 2005)

Original	Candidates
am/VBP, are/VBP or is/VBZ	{am/VBP, are/VBP, is/VBZ}
was/VBD or were/VBD	{was/VBD, were/VBD}
being/VBG	{be/VB, being/VBG}
been/VBN	{be/VB, been/VBN}
be/VB	{be/VB, being/VBG, been/VBN}

Table 1: Examples of candidates in the case of "*be*"

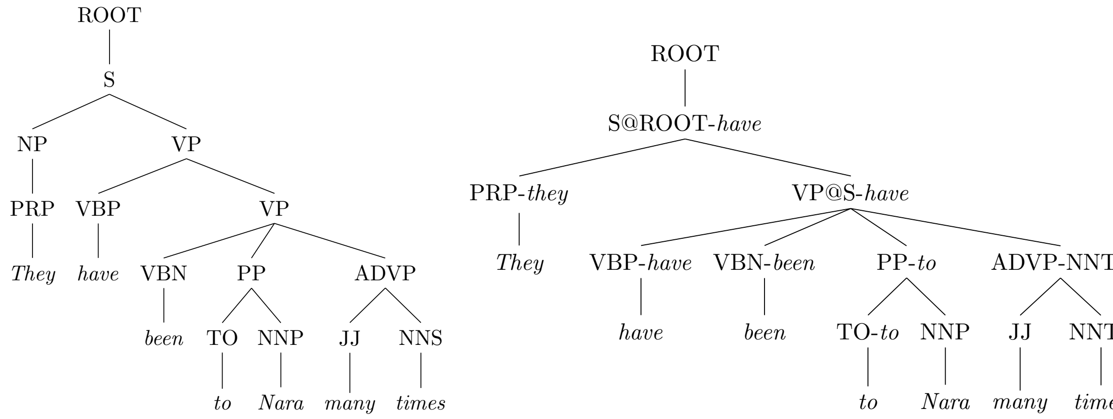

Figure 1: The tree on the left is before annotations and transformations which convert it to the tree on the right.

Pauls and Klein (2012). For instance, the common CFG tree on the left side of Figure 1 is transformed to the one on the right side.

Temporal NPs Pauls and Klein (2012) marked every noun which is the head of an NP-TMP at least once in the Penn Treebank. For example, NN → *time* is replaced with NNT → *time* and NNS → *times* is replaced with NNTS → *times*. This rule seems to be useful for correcting verb tense errors.[4]

Head Annotations We annotated every non-terminal and preterminal with its head word.[5] If the head word is not a closed class word,[6] we annotated non-terminal symbols with the head POS tag instead of the head word.

NP Flattening Pauls and Klein (2012) deleted NPs dominated by other NPs, unless the child NPs are in coordination or apposition. These NPs typically occur when nouns are modified by PPs. Our model therefore assigns probability to nouns conditioned on the head of modifying PPs with prepositions such as "*in*", "*at*" and so on by applying simultaneously the NP Flattening and the Head Annotations. However, our model cannot assign probability to prepositions conditioned on verbs or nouns on which the prepositions depend. For this reason we did not use our model to correct prepositional errors.

Number Annotations Pauls and Klein (2012) divided numbers into five classes: CD-YR for numbers that consist of four digits, which are usually years; CD-NM for entirely numeric numbers; CD-DC for numbers that have a decimal; CD-MX for numbers that mix letters and digits; and CD-AL for numbers that are entirely alphabetic.

SBAR Flattening They removed any S nodes which are children of an SBAR.

VP Flattening They removed any VPs immediately dominated by a VP, unless it is conjoined with another VP. The chains of verbs are represented as separated VPs for each verb, such as `(VP (MD will) (VP (VB be) (VP (VBG`

[4] Verb tense (*Vt*) errors are not covered in this shared task.

[5] We identified the head with almost the same rules used in Collins (1999).

[6] We took the following to be the closed class words: all inflections of the verbs do, be, and have; and any word tagged with IN, WDT, PDT, WP, WP$, TO, WRB, RP, DT, SYM, EX, POS, PRP, AUX, MD or CC.

`playing) ...)))`. This transformation turns the above VPs into `(VP (MD will) (VB be) (VBG playing) ...)`. This has an effect on the correction of auxiliary agreement errors because our model can assign probability to main verbs strongly conditioned on their auxiliary verbs.

Gapped Sentence Annotation They annotated all S and SBAR nodes that have a VP before any NP.

Parent Annotation They annotated all VPs and children of the *ROOT* node with their parent symbol.

Unary Deletion All unary rules are deleted except the root and the preterminal rules. We kept only the bottom-most symbol of the unary rule chain. This brings many symbols into the context of a production rule.

3.1.3 Procedure

Our system for *SVA* and *Vform* errors tries to correct the words in a sentence from left to right. Correction proceeds in the following steps.

1. If the POS tag of the word is "*VB*", "*VBD*", "*VBG*", "*VBN*", "*VBP*" or "*VBZ*", our system generates sentences which have the word replaced with candidates. For example, if the original word is an inflection of "*be*", the system generates candidates as shown in Table 1.

2. The system parses those sentences and obtains the k-best parses for each sentence.

3. The system keeps only the one sentence to which our language model assigned the highest probability in the parses.

4. The system repeats Steps 1 to 3 with the sentence kept in Step 3 until the rightmost word of that sentence.

Note that the system uses the Berkeley Parser[7] in Step 2.

3.2 Noun Number

3.2.1 Errors

A noun number error is the mistake of using the singular form for a plural noun, and vice versa, as in the following:

I saw many ****student*** *yesterday.*

In this example, the inflection of *"student"* is mistaken. It should be *"students"* because it is modified by *"many"*.

To correct such errors, we use a binary classification approach because the inflection of a noun is either "singular" or "plural". If the binary classifier detects an error with a sufficiently high confidence, the system changes the noun form. We adopt the adaptive regularization of weight vectors algorithm (*AROW*) (Crammer et al., 2009). *AROW* is a variant of a confidence weighted linear classification algorithm which is suitable for the classification of large scale data.

3.2.2 Binary classifier approach

The binary classifier indicates "singular" or "plural" for all nouns except proper and uncountable nouns. First, if a noun is found in the training corpus, we extract an instance with features created by the feature template in Table 2.[8] Second, we train a classifier with extracted instances and labels from the training corpus.

We use unigram, bigram, and trigram features around the target word and the path features between the target word and all the other nodes in the NPs that dominate the target word as the rightmost constituent. The path feature is commonly used in semantic role labeling tasks (Pradhan et al., 2004). For the path features, we do not use the right subtree of the NP as the path features because we assume that right subtrees do not affect the number of the target word. We limit the maximum depth of the subtree containing the NP to be *3* because nodes over this limit may be noisy. To encode the relationship between the target word and another node in the NP, we append a symbol which reflects the direction of tree traversal to the label: *'p'* for going up (parent) and *'c'* for going down (child). For example, we show extracted features in Table 2 for the phrase *"some interesting and recent topics about politics and economics"*.

In the training corpus, since the proportions of singular and plural nouns are unequal, we set different thresholds for classifying singular and plural forms. These thresholds limit the probabilities which the binary classifier uses for error detection. We have used a development set to determine the

[7] `http://code.google.com/p/berkeleyparser/`

[8] Target word refers to a noun whose POS tag is "NN" or "NNS" in the Penn Treebank tagset.

Feature name	Word, Pos used as features	Example
surface unigram	word±1, word±2	and, recent, about, politics
surface bigram	word±2_word±1	and_recent, about_politics
surface trigram	word±3_word±2_word±1	interesting_and_recent, about_politics_and
POS unigram	POS±1, POS±2	CC, JJ, IN, NN
POS bigram	POS±1_POS±2	CC_JJ, IN_NN
POS trigram	POS±3_POS±2_POS±1	JJ_CC_JJ, IN_NN_CONJ
lemma unigram	lemma±2, lemma±1	and, recent, about, politics
lemma bigram	lemma±2_lemma±1	and_recent, about_politics
lemma trigram	lemma±3_lemma±2_lemma±1	interesting_and_recent, about_politics_and
lemma target	lemma of target word	topic
path feature	path between the target word and the other nodes in NP	p_NP, pc_JJ, pc_recent, pp_NP, ppc_CC, ppc_and, ppc_NP, ppcc_DT, ppcc_some, ppcc_JJ, ppcc_interesting

Table 2: Features used for the detection of noun number errors and example features for the phrase "some interesting and recent topics about politics and economics".

best thresholds for singular and plural forms, respectively.

For proper and uncountable nouns, we do not change number because of the nature of those nouns. In order to determine whether to change number or not, we create a list which consists of words frequently used as singular forms in the native corpus.

3.3 Prepositions and Determiners

For preposition and determiner errors, we construct a system using a phrase-based statistical machine translation (Koehn et al., 2003) framework. The SMT-based approach functions well in corrections of conventional usage of determiners and prepositions such as "*the young*" and "*take care of*". The characteristic of the SMT-based approach is its ability to capture tendencies in learners' errors. This approach translates erroneous phrases that learners often make to correct phrases. Hence, it can handle errors in conventional expressions without over-generalization.

The phrase-based SMT framework which we used is based on the log-linear model (Och and Ney, 2002), where the decision rule is expressed as follow:

$$\underset{e}{\operatorname{argmax}}\, P(e|f) = \underset{e}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m h_m(e, f)$$

Here, f is an input sentence, e are hypotheses, $h_m(e, f)$ feature functions and λ_m their weights. The hypothesis that maximizes the weighted sum of the feature functions is chosen as an output sentence.

The feature functions encode components of the phrase-based SMT, including the translation model and the language model. The translation model suggests translation hypotheses and the language model filters out ill-formed hypotheses.

For an error correction system based on SMT, the translation model is constructed from pairs of original sentences and corrected sentences, and the language model is built on a native corpus (Brockett et al., 2006).

Brockett et al. (2006) trained the translation model on a corpus where the errors are restricted to mass noun errors. In our case, we trained our model on a corpus with no restriction on error types. Consequently, the system corrects all types of errors. To focus on preposition and determiner errors, we retain proposed edits that include 48 prepositions and 25 determiners listed in Table 3.

4 Experiments

4.1 Experimental setting

4.1.1 Subject-Verb Agreement and Verb Form

We describe here the training data and tools used to train our model. Our model was trained with the Berkeley LM[9] version 1.1.3. We constructed the training data by concatenating the WSJ sections of the Penn Treebank and the AFP sections of the English Gigaword Corpus version 5.[10] Our training data consists of about 27 million sentences. Although human-annotated parses for the WSJ are available, there is no gold standard for the AFP, so we parsed the AFP automatically by using the Berkeley Parser released on October 9, 2012.

[9] http://code.google.com/p/berkeleylm/

[10] LDC2011T07

Preposition	about, across, after, against, along, among, around, as, at, before, behind, below, beside, besides, between, beyond, but, by, despite, down, during, for, from, in, inside, into, near, of, off, on, onto, opposite, outside, over, past, round, without, than, through, to, toward, towards, under, until, up, upon, with, within
Determiner	the, a, an, all, these, those, many, much, another, no, some, any, my, our, their, her, his, its, no, each, every, certain, its, this, that

Table 3: Preposition and determiner lists

4.1.2 Noun Number

We trained a binary classifier on a merged corpus of the English Gigaword and the NUCLE data. From the English Gigaword corpus, we used the New York Times (NYT) as a training corpus. In order to create the training corpus, we corrected all but noun number errors in the NUCLE data using gold annotations.

The AROW++ [11] 0.1.2 was used for the binary classification. For changing noun forms, we used the pattern.en toolkit.[12]

The maximum depth of subtrees containing an NP is set to *3* when we extracted the path features.

We built and used a list of nouns that appear in singular forms frequently in a native corpus. We counted the frequency of nouns in entire English Gigaword. If a noun appears in more than 99%[13] of occurrences in singular form, we included it in the list. The resulting list contains 836 nouns.

4.1.3 Prepositions and Determiners

We used Moses 2010-08-13 with default parameters for our decoder[14] and GIZA++ 1.0.5[15] as the alignment tool. The grow-diag-final heuristics was applied for phrase extraction. As a language modeling tool we used IRSTLM version 5.80[16] with Witten-Bell smoothing.

The translation model was trained on the NUCLE corpus and our Lang-8 corpus.[17] From the Lang-8 corpus, we filtered out noisy sentences. Out of 1,230,257 pairs of sentences, 1,217,124 pairs of sentences were used for training. As for the NUCLE corpus we used 55,151 pairs of sentences from the official data provided as training data. We used a 3-gram language model built on the entire English Gigaword corpus.

4.2 Result

Table 4 shows the overall results of our submitted systems and the results of an additional experiment. In the additional experiment, we tried the SMT-based approach described in Section 3.3 for errors in *SVA*, *Vform* and *Nn*. While the system based on the Treelet Language Model outperformed the SMT-based system on the *SVA* errors and the *Vform* errors, the binary classifier approach did not perform as well as the SMT-based system on the *Nn* errors.

5 Discussion

5.1 Subject-Verb Agreement and Verb Form

We provide here examples of our system's output, beginning with a successful example.

source: *This is an age which most people* ****is*** *retired and* ****has*** *no sources of incomes.*

hypothesis: *This is an age which most people* ***are*** *retired and* ***have*** *no sources of incomes.*

The source sentence of this pair includes two *SVA* errors. The first is that "*be*" should agree with its subject "*people*" and must be "*are*". Our system is able to correct errors where the misinflected predicate is adjacent to its subject. The second error is also an agreement error, in this case between "*have*" and its subject "*people*". Our model can assign probability to yields related to predicates conditioned strongly on their subjects even if the distance between the predicate and its subject is long. The same can be said of *Vform* errors.

One mistake made by our system is miscorrection due to the negative effect of other errors.

source/hypothesis: *The rising life* ****expectancies*** ****are*** *like a two side sword to the modern world.*

[11] https://code.google.com/p/arowpp/

[12] http://www.clips.ua.ac.be/pages/pattern-en

[13] We tested many thresholds, and set 99% as threshold.

[14] http://sourceforge.net/projects/mosesdecoder/

[15] http://code.google.com/p/giza-pp/

[16] http://sourceforge.net/projects/irstlm/

[17] consisting of entries through 2012.

		submitted system					additional experiments	
		ALL	Verb	Nn	Prep	ArtOrDet	Verb	Nn
original	Precision	0.2707	0.1378	0.4452	0.2649	0.3118	0.2154	0.3687
	Recall	0.1832	0.2520	0.1641	0.1286	0.2029	0.0569	0.2020
	F-score	0.2185	0.1782	0.2399	0.1732	0.2458	0.0900	0.2610
revised	Precision	0.3392	0.1814	0.5578	0.3245	0.4027	0.3846	0.4747
	Recall	0.2405	0.2867	0.1708	0.1494	0.2497	0.0880	0.2137
	F-score	0.2814	0.2222	0.2616	0.2046	0.3082	0.1433	0.2947

Table 4: Results of the submitted system for each type of error and results of additional experiments with the SMT-based system. The score is evaluated on the m2scorer (Dahlmeier and Ng, 2012). ALL is the official result of formal run, and each of the others shows the result of the corresponding error type. Since our system did not distinguish *SVA* and *Vform*, we report the combined result for them in the column Verb.

gold: *The rising life* ***expectancy is*** *like a two side sword to the modern world.*

Since the subject of "*are*" is "*expectancies*", the sentence looks correct at first. However, this example includes not only an *SVA* error but also an *Nn* error, and therefore the predicate "*are*" should be corrected along with correcting its subject "*expectancies*".

An example of a *Vform* error is shown below.

source/hypothesis: *Besides, we can try to reduce the bad effect* ****cause*** *by the new technology.*

gold: *Besides, we can try to reduce the bad effect* ***caused*** *by the new technology.*

The word "*cause*" is tagged as "*NN*" in this sentence. This error is ignored because our system makes corrections on the basis of the original POS tag. For a similar example, our system does not make modifications between the *to*-infinitive and the other forms.

5.2 Noun Number

We provide here examples of our system's output, beginning with a successful example.

source: *many of cell* ****phone*** *are equipped with GPS*

hypothesis/gold: *many of cell* ***phones*** *are equipped with GPS*

In the example, the noun *"phone"* should be in the plural form *"phones"*. This is because the phrase *"many of"* modifies the noun. In this case, the unigrams *"many"* and *"are"*, and the bigram *"many of"* are features with strong weights for the plural class as expected.

However, n-gram features sometimes work to the contrary of our expectations.

source/hypothesis: *RFID is not only used to track products for logistical and storage* ****purpose****, it is also used to track people*

gold: *RFID is not only used to track products for logistical and storage* ***purposes****, it is also used to track people*

The *"purpose"* is in the PP which is modified by *"products"*. Thus, *"purpose"* should not be affected by the following words. However, the verb *"is"*, which is immediately after *"purpose"*, has a strong influence to keep the word in singular form. Therefore, it may be better not to use a verb that the word is not immediately dependent on as a feature.

5.3 Prepositions and Determiners

While the SMT-based system can capture the statistics of learners' errors, it cannot correct phrases that are not found in the training corpus.

(1) **source:** ****with*** *economic situation*
gold: ***in*** *economic situation*

(2) **source:** ****with*** *such situation*
gold: ***in*** *such situation*

Our system was not able to correct the source phrase in (1), in spite of the fact that the similar phrase pair (2) was in the training data. To correct such errors, we should construct a system that allows a gap in source and target phrases as in Galley and Manning (2010).

6 Conclusion

This paper described the architecture of our correction system for errors in verb forms, subject verb agreement, noun number, prepositions and determiners. For verb form and subject verb agreement errors, we used the Treelet Language Model. By taking advantage of rich syntactic information, it corrects subject-verb agreement errors which need to be inflected according to a distant subject. For noun number errors, we used a binary classifier using both learner and native corpora. For preposition and determiner errors, we built an SMT-based system trained on a larger corpus than those used in prior works. We show that our subsystems are effective to each error type. On the other hand, our system cannot handle the errors strongly related to other errors well. In future work we will explore joint correction of multiple error types, especially noun number and subject-verb agreement errors, which are closely related to each other.

Acknowledgements

We would like to thank Yangyang Xi for giving us permission to use text from Lang-8 and the anonymous reviewers for helpful comments.

References

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal smt techniques. In *Proceedings of COLING/ACL 2006*, pages 249–256.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Koby Crammer, Alex Kulesza, and Mark Dredze. 2009. Adaptive regularization of weight vectors. In *Proceedings of NIPS 2009*, pages 414–422.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of NAACL 2012*, pages 568–572.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The NUS corpus of learner English. In *Proceedings of BEA 2013*, pages 313–330.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: a report on the preposition and determiner error correction shared task. In *Proceedings of BEA 2012*, pages 54–62.

Michel Galley and Christopher D. Manning. 2010. Accurate non-hierarchical phrase-based translation. In *Processing of HLT/NAACL 2010*, pages 966–974.

Emi Izumi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi, and Hitoshi Isahara. 2003. Automatic error detection in the Japanese learners' English spoken data. In *Proceedings of ACL 2003*, pages 145–148.

Philipp Koehn, Franz Josef Och, and Daniel C Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT/NAACL 2003*, pages 48–54.

John Lee and Stephanie Seneff. 2008. Correcting misuse of verb forms. In *Proceedings of ACL 2008*, pages 174–182.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. The effect of learner corpus size in grammatical error correction of ESL writings. In *Proceedings of COLING 2012*, pages 863–872.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL 2002*, pages 295–302.

Adam Pauls and Dan Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of ACL 2012*, pages 959–968.

Sameer S Pradhan, Wayne H Ward, Kadri Hacioglu, James H Martin, and Dan Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of HLT/NAACL 2004*, pages 233–240.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of ACL 2005*, pages 271–279.