NAIST at the HOO 2012 Shared Task

Keisuke Sakaguchi, Yuta Hayashibe, Shuhei Kondo, Lis Kanashiro
Tomoya Mizumoto, Mamoru Komachi, Yuji Matsumoto
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara 630-0192, Japan
{ keisuke-sa, yuta-h, shuhei-k, lis-k, tomoya-m, komachi, matsu }@is.naist.jp

Abstract

This paper describes the Nara Institute of Science and Technology (NAIST) error correction system in the Helping Our Own (HOO) 2012 Shared Task. Our system targets preposition and determiner errors with spelling correction as a pre-processing step. The result shows that spelling correction improves the Detection, Correction, and Recognition F-scores for preposition errors. With regard to preposition error correction, F-scores were not improved when using the training set with correction of all but preposition errors. As for determiner error correction, there was an improvement when the constituent parser was trained with a concatenation of treebank and modified treebank where all the articles appearing as the first word of an NP were removed. Our system ranked third in preposition and fourth in determiner error corrections.

1 Introduction

Researchers in natural language processing have focused recently on automatic grammatical error detection and correction for English as a Second Language (ESL) learners' writing. There have been a lot of papers on these challenging tasks, and remarkably, an independent session for grammatical error correction took place in the ACL-2011.

The Helping Our Own (HOO) shared task (Dale and Kilgarriff, 2010) is proposed for improving the quality of ESL learners' writing, and a pilot run with six teams was held in 2011.

The HOO 2012 shared task focuses on the correction of preposition and determiner errors. There has been a lot of work on correcting preposition and determiner errors, where discriminative models such as Maximum Entropy and Averaged Perceptron (De Felice and Pulman, 2008; Rozovskaya and Roth, 2011) and/or probablistic language models (Gamon, 2010) are generally used.

In addition, it is pointed out that spelling and punctuation errors often disturb grammatical error correction. In fact, some teams reported in the HOO 2011 that they corrected spelling and punctuation errors before correcting grammatical errors (Dahlmeier et al., 2011).

Our strategy for HOO 2012 follows the above procedure. In other words, we correct spelling errors at the beginning, and then train classifiers for correcting preposition and determiner errors. The result shows our system achieved 24.42% (third-ranked) in F-score for preposition error correction, 29.81% (fourth-ranked) for determiners, and 27.12% (fourth-ranked) for their combined.

In this report, we describe our system architecture and the experimental results. Sections 2 to 4 describe the system for correcting spelling, preposition, and determiner errors. Section 5 shows the experimental design and results.

2 System Architecture for Spelling Correction

Spelling errors in second language learners' writing often disturb part-of-speech (POS) tagging and dependency parsing, becoming an obstacle for grammatical error detection and correction tasks. For example, POS tagging for learners' writing fails be-

The 7th Workshop on the Innovative Use of NLP for Building Educational Applications, pages 281–288,
Montréal, Canada, June 3-8, 2012. ©2012 Association for Computational Linguistics

e.g. I think it is *verey/very *convent/convenient for the group.
without spelling error correction: ... ('it', 'PRP'), ('is', 'VBZ'), ('verey', 'PRP'), ('convent', 'NN'), ...
with spelling error correction : ... ('It', 'PRP'), ('is', 'VBZ'), ('very', 'RB'), ('convenient', 'JJ'), ...

Figure 1: POS tagging for learners' writing without and with spelling error correction.

cause of misspelled words (Figure 1).[1]

To reduce errors derived from misspelled words, we conduct spelling error correction as a preprocessing task. The procedure of spelling error correction we use is as follows. First of all, we look for misspelled words and suggest candidates by GNU Aspell[2], an open-source spelling checker. The candidates are ranked by the probability of 5-gram language model built from Google N-gram (Web 1T 5-gram Version 1)[3] (Brants and Franz, 2006) with IRST LM Toolkit (Federico and Cettolo, 2007).[4] Finally, according to the rank, we changed the misspelled word into the 1-best candidate word.

In a preliminary experiment, where we use the original CLC FCE dataset,[5] our spelling error correction obtains 52.4% of precision, 72.2% of recall, and 60.7% of F-score.

We apply the spelling error correction to the training and test sets provided, and use both spelling-error and spelling-error-free sets for comparison.

3 System Architecture for Preposition Error Correction

There are so many prepositions in English. Because it is difficult to perform multi-class classification, we focus on twelve prepositions: *of, in, for, to, by, with, at, on, from, as, about, since*, which account for roughly 91% of preposition usage (Chodorow et al., 2010).

The errors are classified into three categories according to their ways of correction. First, **replacement error** indicates that learners use a wrong preposition. For instance, *with* in Example (1) is a replacement error.

I went there ~~with~~$_{by}$ bus. (1)

Second, **insertion error** points out they incorrectly inserted a preposition, such as "about" in Example (2).[6]

We discussed ~~about~~$_{NONE}$ the topic. (2)

Third, **deletion error** means they fail to write obligatory prepositions. For example, "NONE" in Example (3) is an deletion error.

This is the place to relax ~~NONE~~$_{in}$. (3)

Replacement and insertion error correction can be regarded as a multi-class classification task at each preposition occurrence. However, deletion errors differ from the other two types of errors in that they may occur at any place in a sentence. Therefore, we build two models, a combined model for replacement and insertion errors and a model for deletion errors, taking the difference into account.

For the model of replacement and insertion errors, we simultaneously perform error detection and correction with a single model.

For the model of deletion errors, we only check whether direct objects of verbs need prepositions, because it is time consuming to check all the gaps between words. Still, it covers most deletion errors.[7]

We merge the outputs of the two models to get the final output.

We used two types of training sets extracted from the original CLC-FCE dataset. One is the "gold" set, where training sentences are corrected except for preposition errors. In the gold set, spelling errors are also corrected to the gold data in the corpus. The other is the "original" set, which includes the

[1] The example is extracted from the CLC FCE dataset and part-of-speech tagged by Natural Language Toolkit (NLTK). http://www.nltk.org/

[2] GNU Aspell 0.60.6.1 http://aspell.net/

[3] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13

[4] irstlm5.70 http://sourceforge.net/projects/irstlm/

[5] In the CLC FCE dataset, misspelled words are corrected and tagged with a label "S".

[6] "NONE" means there are no words.

[7] 2,407 out of 5,324 preposition errors in CLC-FCE are between verbs and nouns.

Type	Name	Description (NP and PRED refer a noun phrase and a predicate.)
Lexical	Token n-gram	Token n-grams in a 2 word window around the preposition
	POS n-gram	POS n-grams in a 2 word window around the preposition
	HEAD_PREC_VP	The head verb in the preceding verb phrase
	HEAD_PREC_NP	The head noun in the preceding noun phrase
	HEAD_FOLLOW_NP	The head noun in the following noun phrase
Parsing	HEAD	Head of the preposition
	HEAD_POS	POS of the head
	COMP	Complement of the preposition
	COMPLEMENT_POS	POS of the complement
	HEAD_RELATION	Prep-Head relation name
	COMPLEMENT_RELATION	Prep-Comp relation name
Phrase Structure	PARENT_TAG	TAG of the preposition's parent
	GRANDPARENT_TAG	TAG of the preposition's grandparent
	PARENT_LEFT	Left context of the preposition parent
	PARENT_RIGHT	Right context of the preposition's parent
Web N-gram	COUNT	For the frequency $f_{\text{prep,i}}$ of i (3 to 5) window size phrase including the preposition prep, the value of $\log_{100}(f_i + 1)$
	PROPORTION	The proportion $p_{\text{prep,i}}$ (i is 3 to 5). $p_{\text{prep,i}} = \frac{f_{\text{prep,i}}}{\sum_{k \in T} f_{k,i}}$, given the set of target prepositions T.
Semantic	WORDNET_CATEGORY	WordNet lexicographer classes which are about 40 broad semantic categories for all words used as surface features. As De Felice and Pulman (2008) did not perform word sense disambiguation, neither did we.

Table 1: Baseline features for English preposition error correction.

original CLC-FCE plain sentences.

We performed sentence splitting using the implementation of Kiss and Strunk (2006) in NLTK 2.0.1rc2. We conducted dependency parsing by Stanford parser 1.6.9.[8]

We used the features described in (Tetreault et al., 2010) as shown in Table 1 with Maximum Entropy (ME) modeling (Berger et al., 1996) as a multi-class classifier. We used the implementation of Maximum Entropy Modeling Toolkit[9] with its default parameters. For web n-gram calculation, we used Google N-gram with a search system for giga-scale n-gram corpus, called SSGNC 0.4.6.[10]

4 System Architecture for Determiner Error Correction

We focused on article error correction in the determiner error correction subtask, because the errors related to articles significantly outnumber the errors unrelated to them. Though more than twenty types of determiners are involved in determiner error corrections of the HOO training set, over 90% of errors are related to three articles *a*, *an* and *the*. We defined article error correction as a multi-class classification problem with three classes, *a*, *the* and *null* article, and assumed that target articles are placed at the left boundary of a noun phrase (NP). The indefinite article *an* was normalized to *a* in training and testing, and restored to *an* later in an example-based post-processing step. If the system output was *a* and the word immediately after *a* appeared more frequently with *an* than with *a* in the training corpus, *a* was restored to *an*. If the word appeared equally frequently with *a* and *an* or didn't appear in the training corpus, *a* was restored to *an* if the word's first character was one of a, e, i, o, u.

Each input sentence was parsed using the Berkeley Parser[11] with two models, "normal" and "mixed". The "normal" model was trained on a treebank of normal English sentences. In preliminary experiments, the "normal" model sometimes misjudged the span of NPs in ESL writers' sentences due to missing articles. So we trained the "mixed" model on a concatenation of the normal treebank and a modified treebank in which all the articles appearing as the first word of an NP were removed. By

[8] http://nlp.stanford.edu/software/lex-parser.shtml

[9] https://github.com/lzhang10/maxent

[10] http://code.google.com/p/ssgnc/

[11] version 1.1, http://code.google.com/p/berkeleyparser/

Name	Description
HeadNounWord	The word form of the head noun
HeadNounTag	The POS tag of the head noun
ObjOfPrep	Indicates that the head noun is an object of a preposition
PrepWord	The word form of the preposition
PrepHeadWord	The word form of the preposition's syntactic parent
PrepHeadTag	The POS tag of the preposition's syntactic parent
ContextWindowTag	The POS tag of the words in a 3 word window around the candidate position for the article
ContextWindowWord	The word form of the word immediately following the candidate position for the article
ModByDetWord	The word form of the determiner that modifies the head noun
ModByAdjWord	The word form of the adjective that modifies the head noun
ModByAdjTag	The POS tag of the adjective that modifies the head noun
ModByPrep	Indicates that the head noun is modified by a preposition
ModByPrepWord	The word form of the preposition that modifies the head noun
ModByPossesive	Indicates that the head noun is modified by a possesive
ModByCardinal	Indicates that the head noun is modified by a cardinal number
ModByRelative	Indicates that the head noun is modified by a relative clause

Table 2: Feature templates for English determiner correction.

augmenting the training data for the parser model with sentences lacking articles, the span of NPs that lack an article might have better chance of being correctly recognized. In addition, dependency information was extracted from the parse using the Stanford parser 1.6.9.

For each NP in the parse, we extracted a feature vector representation. We used the feature templates shown in Table 2, which are inspired by (De Felice, 2008) and adapted to the CFG representation.

For the parser models, we trained the "normal" model on the WSJ part of Penn Treebank sections 02-21 with the NP annotation by Vadas and Curran (2007). The "mixed" model was trained on the concatenation of the WSJ part and its modified version. For the classification model, we used the written part of the British National Corpus (BNC) in addition to the CLC FCE Dataset, because the amount of in-domain data was limited. In examples taken from the CLC FCE Dataset, the true labels after the correction were used. In examples taken from the BNC, the article of each NP was used as the label. We trained a linear classifier using opal[12] with the PA-I algorithm. We also used the feature augmentation approach of (Daumé III, 2007) for domain adaptation.

	Subsystem Parameters		
Run	**Spelling**	**Preposition**	**Determiner**
0	no change	gold	mixed
1	no change	gold	normal
2	no change	original	mixed
3	no change	original	normal
4	corrected	gold	mixed
5	corrected	gold	normal
6	corrected	original	mixed
7	corrected	original	normal

Table 3: Distinct configurations of the system.

5 Experiment and Result

Previously undisclosed data extracted from the CLC-FCE dataset was provided as a test set by the HOO organizers. The test set includes 100 essays and each contains 180.1 word tokens on average.

We defined eight distinct configurations based on our subsystem parameters (Table 3). The official task evaluation uses three metrics (Detection, Recognition, and Correction), and three measures Precision, Recall, and F-score were computed[13] for

[12] http://www.tkl.iis.u-tokyo.ac.jp/~ynaga/opal/

[13] For details about the evaluation metrics, see http://

	Detection			Correction			Recognition		
Run	R	P	F	R	P	F	R	P	F
0	29.58	34.09	31.67	19.86	22.90	21.27	26.71	30.78	28.60
1	28.69	36.41	32.09	19.42	24.64	21.72	25.82	32.77	28.88
2*	28.91	37.21	32.54	20.97	26.98	23.60	26.26	33.80	29.56
3	28.03	40.18	33.02	20.52	29.43	24.18	25.38	36.39	29.90
4	30.24	33.66	31.86	20.75	23.09	21.86	27.37	30.46	28.83
5	29.13	35.57	32.03	19.64	23.98	21.60	26.26	32.07	28.88
6	29.35	36.23	32.43	21.41	26.43	23.65	26.26	32.42	29.02
7	28.25	38.67	32.65	20.30	27.29	23.46	25.16	34.44	29.08

Table 4: Result for preposition and determiner errors combined before revisions.

*We re-evaluated the Run2 because we submitted the Run2 with the same condition as Run0.

		Detection			Correction			Recognition		
Spelling	Preposition	R	P	F	R	P	F	R	P	F
no change	gold	25.00	34.70	29.06	14.40	20.00	16.74	20.76	28.82	24.13
no change	original	23.30	42.63	30.13	16.52	30.23	21.36	19.91	36.43	25.75
corrected	gold	26.69	34.80	30.21	15.25	19.88	17.26	22.45	29.28	25.41
corrected	original	24.57	41.13	30.76	16.52	27.65	20.68	20.33	34.04	25.46

Table 5: Result for preposition errors before revisions.

each metric.

Table 4 to Table 9 show the overall results of our systems. In terms of the effect of pre-processing, spelling correction improved the F-score of Detection, Correction, and Recognition for preposition errors after revision, whereas there were fluctuations in other conditions. This may be because there were a few spelling errors corrected in the test set.[14] Another reason why no stable improvement was found in determiner error correction is because spelling correction often produces nouns that affect the determiner error detection and correction more sensitively than prepositions. For example, a misspelled word **freewho / free who* was corrected as *freezer*. This type of error may have increased false positives. The example **National Filharmony / the National Philharmony* was corrected as *National Fleming*, where the proper noun *Fleming* does not need a determiner and this type of error increased false negatives.

As for preposition error correction, the classifier performed better when it was trained with the "original" set rather than the error-corrected (all but preposition errors) "gold" set. The reason for this is that the gold set is trained with the test set that contains several types of errors which the original CLC-FCE dataset also contains. Therefore, the "original" classifier is more optimised and suitable for the test set than the "gold" one.

For determiner error correction, the "mixed" model improved precision and F-score in the additional experiments.

5.1 Error Analysis of Preposition Correction

We briefly analyze some errors in our proposed model according to the three categories of errors.

First, most replacement errors require deep understanding of context. For instance, *for* in Example (4) must be changed to *to*. However, *modifications of* is also often used, so it is hard to decide either *to* or *of* is suitable based on the values of N-gram frequencies.

Its great news to hear you have been given extra money and that you will spend it in modifications ~~for~~$_{to}$ the cinema. (4)

Second, most insertion errors need a grammatical judgement rather than a semantic one. For instance, "in" in Example (5) must be changed to "NONE."

Their love had always been kept ~~in~~$_{NONE}$ secret (5)

In order to correct this error, we need to recog-

correcttext.org/hoo2012/eval.html

[14] There was one spelling correction per document in average.

		Detection			Correction			Recognition		
Spelling	**Determiner**	**R**	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**
no change	mixed	34.10	33.18	33.63	25.80	25.11	25.45	33.17	32.28	32.72
no change	normal	32.25	37.43	34.65	24.88	28.87	26.73	31.33	36.36	33.66
corrected	mixed	33.64	32.30	32.95	26.72	25.66	26.18	32.71	31.41	32.05
corrected	normal	31.33	35.78	33.41	24.42	27.89	26.04	30.41	34.73	32.43

Table 6: Result for determiner errors before revisions.

	Detection			Correction			Recognition		
Run	**R**	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**
0	31.28	37.65	34.18	22.62	27.22	24.71	28.54	34.35	31.17
1	30.44	40.33	34.69	22.19	29.41	25.30	27.69	36.69	31.56
2*	31.07	41.76	35.63	23.04	30.96	26.42	28.11	30.96	32.24
3	30.23	45.25	36.24	22.62	33.86	27.12	27.27	40.82	32.69
4	31.92	37.10	34.31	23.46	27.27	25.22	29.17	33.90	31.36
5	30.86	39.35	34.59	22.41	28.57	25.11	28.11	35.84	31.51
6	31.71	40.87	35.71	23.89	30.79	26.90	28.75	37.05	32.38
7	30.65	43.80	36.06	22.83	32.62	26.86	27.69	39.57	32.58

Table 7: Result for preposition and determiner errors combined after revisions.

*We re-evaluated the Run2 because we submitted the Run2 with the same condition as Run0.

nize “keep” takes an object and a complement; in Example (5) “love” is the object and “secret” is the complement of “keep” while the former is left-extraposed. A rule-based approach may be better suited for these cases than a machine learning approach.

Third, most deletion errors involve discrimination between transitive and intransitive. For instance, “NONE” in Example (6) must be changed to “for”, because “wait” is intransitive.

I'll wait ~~NONE~~$_{for}$ your next letter. (6)

To deal with these errors, we may use rich knowledge about verbs such as VerbNet (Kipper et al., 2000) and FrameNet (Baker et al., 1998) in order to judge whether a verb is transitive or intransitive.

5.2 Error Analysis of Determiner Correction

We conducted additional experiments for determiner errors and report the results here because the submitted system contained a bug. In the submitted system, while the test data were parsed by the “mixed” model, the training data and the test data were parsed by the default grammar provided with Berkeley Parser. Moreover, though there were about 5.5 million sentences in the BNC corpus, only about 2.7 million of them had been extracted. Though these errors seem to have improved the performance, it is difficult to specify which errors had positive effects.

Table 10 shows the result of additional experiments. Unlike the submitted system, the “mixed” model contributed toward a higher precision and F-score. Though the two parser models parsed the sentences differently, the difference in the syntactic analysis of test sentences did not always led to different output by the downstream classifiers. On the contrary, the classifiers often returned different outputs even for an identically parsed sentence. In fact, the major source of the performance gap between the two models was the number of the wrong outputs rather than the number of correct ones. While the “mixed” model without spelling correction returned 146 outputs, of which 83 were spurious, the “normal” model without spelling correction produced 209 outputs, of which 143 were spurious. This may suggest the difference of the two models can be attributed to the difference in the syntactic analysis of the training data.

One of the most frequent types of errors common to the two models were those caused by misspelled words. For example, when *your letter* was misspelled to be **yours letter*, it was regarded as an

		Detection			**Correction**			**Recognition**		
Spelling	**Preposition**	**R**	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**
no change	gold	26.63	38.23	31.40	17.62	25.29	20.77	23.36	33.52	27.53
no change	original	26.22	49.61	34.31	18.44	34.88	24.12	22.54	42.63	29.49
corrected	gold	28.27	38.12	32.47	18.44	24.86	21.17	25.00	33.70	28.70
corrected	original	27.86	48.22	35.32	19.26	33.33	24.41	24.18	41.84	30.64

Table 8: Result for preposition errors after revisions.

		Detection			**Correction**			**Recognition**		
Spelling	**Determiner**	**R**	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**
no change	mixed	35.37	36.32	35.84	27.94	28.69	28.31	34.06	34.97	34.51
no change	normal	33.62	41.17	37.01	27.07	33.15	29.80	32.31	39.57	35.57
corrected	mixed	34.93	35.39	35.16	28.82	29.20	29.01	33.62	34.07	33.84
corrected	normal	32.75	39.47	35.79	26.63	32.10	29.11	31.44	37.89	34.36

Table 9: Result for determiner errors after revisions.

		Detection			**Correction**			**Recognition**		
Spelling	**Determiner**	**R**	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**
no change	mixed	27.39	43.15	33.51	23.04	36.30	28.19	27.39	43.15	33.51
no change	normal	28.69	31.57	30.06	22.61	24.88	23.69	28.69	31.57	30.06
corrected	mixed	27.39	41.44	31.98	22.61	34.21	27.22	26.96	40.79	32.46
corrected	normal	30.43	33.33	31.82	24.34	26.67	25.45	30.00	32.86	31.36

Table 10: Result of additional experiments for determiner errors after revisions.

NP without a determiner resulting in a false positive such as **a yours letter*. Among the other types of errors, several seemed to be caused by the information from the context window. For instance, the system output for *It was last month and ...* was *it was *the last month and ...*. It is likely that the word *last* triggered the misinsertion here. Such kind of errors might be avoided by conjunctive features of context information and the head word. Last but not least, compound errors were also frequent and probably the most difficult to solve. For example, it is quite difficult to correct **for a month* to *per month* if we are dealing with determiner errors and preposition errors separately. A more sophisticated approach such as joint modeling seems necessary to correct this kind of errors.

6 Conclusion

This report described the architecture of our preposition and determiner error correction system. The experimental result showed that spelling correction advances the performance of Detection, Correction and Recognition for preposition errors. In terms of preposition error correction, F-scores were not improved when the error-corrected dataset was used. As to determiner error correction, there was an improvement when the constituent parser was trained on a concatenation of treebank and modified treebank where all the articles appearing as the first word of an NP were removed.

Acknowledgements

This work was partly supported by the National Institute of Information and Communications Technology Japan.

References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 86–90, Montreal, Quebec, Canada.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

Thorsten Brants and Alex Franz. 2006. *Web 1T 5-gram Corpus Version 1.1.* Linguistic Data Consortium.

Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. The Utility of Article and Preposition Error Correction Systems for English Language Learners: Feedback and Assessment. *Language Testing*, 27(3):419–436.

Daniel Dahlmeier, Hwee Tou Ng, and Thanh Phu Tran. 2011. NUS at the HOO 2011 Pilot Shared Task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 257–259, Nancy, France.

Robert Dale and Adam Kilgarriff. 2010. Helping Our Own: Text Massaging for Computational Linguistics as a New Shared Task. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 261–266, Trim, Co. Meath, Ireland.

Hal Daumé III. 2007. Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic.

Rachele De Felice and Stephen G. Pulman. 2008. A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 169–176, Manchester, UK.

Rachele De Felice. 2008. *Automatic Error Detection in Non-native English*. Ph.D. thesis, University of Oxford.

Marcello Federico and Mauro Cettolo. 2007. Efficient Handling of N-gram Language Models for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 88–95, Prague, Czech Republic.

Michael Gamon. 2010. Using Mostly Native Data to Correct Errors in Learners' Writing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 163–171, Los Angeles, California.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-based Construction of a Verb Lexicon. In *Proceedings of the 7th National Conference on Artificial Intelligence*, pages 691–696, Austin, Texas, USA.

Tibor Kiss and Jan Strunk. 2006. Unsupervised Multilingual Sentence Boundary Detection. *Computational Linguistics*, 32(4):485–525.

Alla Rozovskaya and Dan Roth. 2011. Algorithm Selection and Model Adaptation for ESL Correction Tasks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 924–933, Portland, Oregon, USA.

Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using Parse Features for Preposition Selection and Error Detection. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics Short Papers*, pages 353–358, Uppsala, Sweden.

David Vadas and James Curran. 2007. Adding Noun Phrase Structure to the Penn Treebank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 240–247, Prague, Czech Republic.